DEPARTMENT OF HEALTH & HUMAN SERVICES
Centers for Medicare & Medicaid Services
7500 Security Boulevard, Mail Stop N2-14-26
Baltimore, Maryland 21244-1850

CENTERS FOR MEDICARE & MEDICAID SERVICES (CMS)

Office of Information Services (OIS)
Systems Security Group (SSG)
7500 Security Blvd
Baltimore, MD 21244-1850

CMS INFORMATION SECURITY TESTING APPROACH

Version 1.0
May 13, 2005

Summary of changes from DRAFT Version 1

1. Section 1.2: Added: publication date of February 2005 for NIST SP 800-53; titles covering Electronic Authentication guidance to NIST SP 800-63 and OMB Memorandum M-04-04; and the General Accounting Office GAO/AIMD-12.19.6, January publication date to FISCAM publication.
2. Section 2.1.1: Added a paragraph explaining E-Authentication Implications.
3. Section 2.1.2: Changed wording to show that the information sensitivity level classification for the data processed by the application(s) supporting the business function is contained in the *IS Business RA Report* for the business function under review.
4. Section 2.1.3: Added to explain how assurance level classification is determined for E-Authentication.
5. Section 2.1.5: Added bullet 6 to show the use of e-authentication and bullet 7 to identify authentication controls.
6. Section 3.1: Added, into the middle of the first paragraph, the last sentence to show that identifying and prioritizing security controls for testing was done by CMS, SSG.
7. Section 4: Added the last sentence, of the first paragraph, to show that the identification and prioritization of application security testing will be made by CMS, SSG; a section to briefly describe the testing techniques used; and the Application Source Code Review.
8. Section 4.3.1: Added a bullet to show the addition of Appendix G.
9. Appendix A: Added an introductory disclaimer paragraph to the tool tables.
10. Appendix B; Added an introductory disclaimer paragraph to the tool tables.
11. Appendix C; Added an introductory disclaimer paragraph to the tool tables.
12. Appendix G: Added as new appendix.

Table of Contents

1 INTRODUCTION

The CMS Application Security Test Approach establishes a uniform approach for the conduct of information security testing of CMS' Information Systems for Major Applications (MAs) and their underlying component application systems. This Approach is designed to provide guidance for completing business risk-driven information security assessments of CMS information systems. To employ a risk-based model for security testing successfully, the business functions supported by the underlying technology must be understood fully by the testing entity. A thorough understanding of the CMS business function will drive the review and the identification of business risks, which determine the need for, and reasonableness of, internal controls. This Approach divides the information security testing process into three phases; Business Process Review and Documentation Gap Analysis, Infrastructure Security Assessment, and Application Security Test. Each of the three phases has distinct goals and objectives, however, all three phases are interrelated and interdependent.

The Business Process Review (see Section II) forms the foundation for all subsequent testing activities. The level and type of infrastructure and application security testing to be conducted shall be dependent upon the information sensitivity, the documented security control requirements, and the known business risks. This information is gathered during the Business Process Review, and is analyzed as part of the Documentation Gap Analysis. The interrelation between the three phases is represented in the diagram to the right. Each phase is distinct and individual, yet a part of each phase overlaps with, and is dependent upon, the other two phases.

1.1 PURPOSE

The purpose of this Approach is to establish a formal and consistent process for the conduct of information security testing of CMS information systems. The CMS Application Security Test Approach provides a standardized methodology for scoping, planning, performing, documenting, and managing the information security assessment of CMS infrastructure components and applications. The goal of this Approach is to establish a uniform process that complies with CMS information security policies and federal legislative and regulatory requirements and which all CMS personnel and contractors will follow when conducting security testing.

1.2 SCOPE

This document applies to information security testing of all CMS information systems conducted by CMS personnel and / or contractors.

This Approach is based upon the requirements and guidance documented within the National Institute of Standards and Technology (NIST) Special Publication 800-53, *Recommended Security Controls for Federal Information Systems*, February 2005; NIST SP 800-63, *Electronic*

Authentication Guideline, Version 1.0.1, September 2004; Office of Management and Budget (OMB) Memorandum M-04-04, *E-Authentication Guidance for Federal Agencies*, December 2003; and the General Accountability Office, *Federal Information Systems Audit Manual* (FISCAM), GAO/AIMD-12.19.6, January 1999.

2 BUSINESS PROCESS REVIEW AND DOCUMENTATION GAP ANALYSIS

2.1 BUSINESS PROCESS REVIEW

In order to conduct a comprehensive evaluation of the management, operational, and technical security controls implemented to safeguard a CMS application or information system, it is important to gain a fundamental understanding of the business function(s) supported by the application(s). This is accomplished by conducting interviews with key personnel, and by reviewing all associated application documentation. The initial documentation resource will be the *IS Business RA Report* for the business function supported by the application under review. IS Business RAs are conducted in accordance with the *CMS Information Security (IS) Business Risk Assessment (RA) Methodology.*

The data gathered during this phase will form the basis for all subsequent activities and will be used to develop the test plan and test scripts, which will guide the application test.

2.1.1 REVIEW BUSINESS ENVIRONMENT

To understand the business environment, obtain a mission statement and a general description of the business function, including any interdependencies between the application under review and any other CMS business functions. Identify and document all assets, including information types and personnel supporting the business function. Most subsequent activities, including the review and identification of business and technical risks, are determined by the sensitivity and criticality requirements of the business function.

A thorough understanding of the CMS business function is required in order to be able to assess risks and to recommend reasonable and effective management, operational and technical security controls where the current controls do not adequately protect the application. The reviewer shall request copies of any audits, reviews or studies that have been conducted on the application. These include, but are not limited to: any GAO, OIG, or internal audit reports; internal reviews; self-assessment studies; reports of Congressional hearings; copies of Congressional testimony; and summaries or status of any on-going or planned reviews or audits. The reviewer shall also request copies of brochures, booklets, pamphlets, etc., that document or are related to the business function, automated applications or operations.

Request any overview diagram(s) that document the business function data flow. This should cover the major inputs and data entry points, data flows, communication networks, process sites, and major outputs and output points.

Business Portability Implications

Besides evaluating controls and procedures, it is necessary to identify any business portability implications. The portability implications are determined by the business function's requirements, which could be driven by the need to distribute software, developed and tested at CMS, to CMS business partners off-site. Alternatively, the application could be hosted at a non-

CMS site, or a change of host site might be required. Any business portability implications identified during the business environment analysis will guide the technical portability requirements that must be addressed during the system environment review.

E-Authentication Implications
During the business environment analysis it is essential to determine any e-authentication implications for the application. The Government Paper Elimination Act, October 1998, requires CMS to allow individuals or entities the option to submit information electronically and to maintain records electronically, wherever possible. OMB M-04-04, *e-Authentication Guidance for Federal* Agencies, mandates and drives the consistent implementation and documentation of controls for e-authentication and their periodic testing and verification during a required risk assessment process. This includes determining and documenting any e-authentication implications for the application. A business risk analysis and a cost-benefit analysis will determine if an electronic transaction is possible. The assessment should evaluate the suitability of e-authentication alternatives.

2.1.2 DETERMINE INFORMATION SENSITIVITY LEVEL

The *IS Business RA Report* for the business function under review contains the information sensitivity level classification for the data processed by the application(s) supporting the business function. The sensitivity level is determined by the potential business impact to CMS that would be realized if the security of the data were to be compromised due to the loss of confidentiality (including privacy), integrity or availability. Confidentiality (including privacy) requires data to be protected from unauthorized disclosure. Integrity relates to the quality of the data protected from unauthorized modification, which is to say: the authenticity of the data (ability to verify that the data have not been modified in transit); non-repudiation (the source and destination of a message must be verifiable by a third party); and accountability (the ability to trace the actions of an entity only to that entity). Availability requires that the data be accessible on a timely basis to meet CMS' business requirements.

The information sensitivity level for the business function will, in part, drive the development of the application test plan and test scripts. Applications supporting high-sensitivity business functions will require greater emphasis on confidentiality, integrity and availability (CIA) controls. Test procedures shall be defined in the test plan and test scripts to validate the effectiveness of such controls.

2.1.3 DETERMINE ASSURANCE LEVEL FOR E-AUTHENTICATION

The *IS RA Report* for the system under review contains the assurance level classification. The assurance level is determined, based on the impacts of an authentication risk (threat) on potential impact categories as shown below:

- Inconvenience, distress or damage to standing or reputation;
- Financial loss or agency liability;
- Harm to agency programs or public interests;
- Unauthorized release of sensitive information;
- Personal Safety; and

- Civil or criminal violations.

The assurance level for e-authentication will, in part, drive the development of the application test plan and test scripts. Applications with highest assurance level will require greater emphasis on CIA controls. Test procedures shall be defined in the test plan and test scripts to validate the effectiveness of such controls.

2.1.4 DETERMINE CRITICALITY OF BUSINESS FUNCTION

The criticality of the business function is largely a measure of the reliance CMS or the public places upon the continued CIA of the function. The criticality of the business function defines the security requirements for integrity and availability controls for the technical application supporting the underlying function. The criticality assessment shall consider: (1) the value of the business function to the CMS mission; (2) the value of the business function to the CMS revenue stream; and (3) the political and / or legal significance of the business function.

The value of the business function to the CMS mission is, in actuality, a measure of the potential business impact that would be realized if the function were not available. A significant negative business impact resulting from interruption of the business function corresponds to a high criticality level. Likewise, if interruption of the business function were likely to cause a significant impact to CMS financial processing and payment functions, the criticality of the business function would be high. The third factor to consider is the political / legal significance of the business function. A business function may not have a major financial or business mission impact on CMS, but if law mandates the availability of a business function or the business function is politically high profile, the business function criticality can be high.

The criticality assessment for the business function will, in part, drive the categorization of security controls and the development of the application test plan and test scripts. Applications supporting high-criticality business functions will require greater emphasis on CIA controls. The appropriate control categories shall then be assigned greater priority, and test procedures shall be defined in the test plan and test scripts to validate the effectiveness of such controls.

2.1.5 REVIEW SYSTEM ENVIRONMENT

Once the purpose of the system is understood, it is necessary to understand the system environment. Obtain a general description of the technical specifications of the business function by interviewing the appropriate personnel and by requesting copies of relevant documentation. This review will include any environmental or technical issues that may raise special security concerns such as dial-up access, system interconnections, e-authentication, and portability requirements.

Obtain any contingency planning or disaster recovery planning documentation, off-site storage policy if relevant, the date and results of the last test of the contingency plan and disaster recovery plan, and the last audit date and its results of the off-site storage facility. Request policies, procedures, and other documentation related to continuity of operations, and disaster recovery planning.

The system environment is defined by the technical systems supporting the business function. In order to understand the system environment, first obtain a general overview of the technical systems supporting the business function, and then request any documentation describing the technical environment of the application being reviewed. This review is concerned specifically with the infrastructure components directly supporting or contributing to the operating environment in which the application runs. Once the system environment is understood at a high level, then that knowledge guides the collection of the detailed technical specifications of the application(s) supporting the business function.

In order to understand the processing environment, obtain any diagrams that describe the relationships between: information systems; major peripherals; network(s) (LAN / WAN); network topology; speed and type of communication links; and the telecommunication system's use of modems and terminals. Note any technical portability requirements, as these details will need to be documented when listing the system's operational requirements once the system environment is understood.

An inventory of the application's hardware, software, network connections and any other relevant technical information should be contained within the GSS' System Security Plan (SSP). Review the inventory list, and validate that the following components are accounted for:

- The system specifications and the operating system. Common operating systems include Windows, AIX, Solaris, Novell, MVS and z/OS;
- All peripherals and their technical specifications, location and the quantity of master consoles, direct access storage devices, other storage devices, optical scanners, modems, tape units, disk units, printers, communication controllers (by type), intelligent terminals (and purpose), dumb terminals;
- Network infrastructure components, related directly to application technical environment (firewall, router, switch, hub, etc.);
- The telecommunications environment. If cooperative processing is used, describe the arrangement. Review any agreements negotiated between the parties, such as the Interconnection Security Agreement (ISA), Data Use Agreement (DUA), etc.;
- Electronic Data Interchange (EDI) use which should be noted and described where relevant;
- Use of e-authentication which should be noted and described where relevant;
- Authentication controls (PIN, passwords, biometrics, etc.);
- The Data Encryption Policy, if any, which should be noted;
- Any system interconnections or information sharing, which should be listed, along with system identifiers. Determine if the external system is covered by a security plan. If not, identify any security concerns to be reviewed. Obtain all MOUs / MOAs (Memoranda of Understanding / Memoranda of Agreement) detailing the rules of behavior for the interconnecting systems. These rules must be included in the security plan and included in the application security test plan;
- On-line monitors;
- Tape management systems;
- Program library software for source code;
- Program library software for object code;

- Job accounting software;
- On-line program development system software;
- Access control software;
- Database management systems e.g., DB2, IMS, IDMS, ADABASE, ORACLE, DATACOM or SQL Server;
- Audit software packages;
- Report writer / generator software;
- Network master control system software;
- Job entry subsystems;
- Job scheduling systems;
- Performance monitor software;
- Dial-up security software packages; and
- Technical portability requirements.

Using the list above as a guide to types of possible infrastructure components, compile a detailed list of these and any other systems software supporting the application system's environment. While creating the list, any technical portability requirements must be included as this information will be integrated later in the application test plan. When executing the application test plan, portability shortfalls identified will be included as a "finding" in the application security test report.

As part of the system environment review, processing statistics and abnormal termination (abends) data will be collected when available. These data can be used to identify operational problems such as excessive downtime or system utilization / capacity issues. System processing statistics include: a breakdown of the most recent system usage and availability by quantifying CPU (or processing unit) production processing; test processing; re-run processing; maintenance efforts; idle time; unplanned downtime; and any other available processing statistics. Abnormal terminations should be broken down by type, which can be: systems software; application software; hardware; operator error; or any other category for which data are available.

During the review of the system's software any future major application software enhancements or planned changes should be noted. If planned changes will replace a current system's software component they should be noted along with the current system's software component documentation.

2.1.6 REVIEW APPLICABLE LAWS OR REGULATIONS AFFECTING THE SYSTEM

Determine all applicable federal laws, regulations, policies, guidelines, and standards governing the system. This includes all MOUs, MOAs and interconnection Agreements that establish application specific requirements for confidentiality, integrity and / or availability of data / information in the system.

2.1.7 DETERMINE OPERATIONAL STATUS

Determine the operational status of the application under review. This status can be operational, under development, or undergoing a major modification. An operational system is defined as a system that has been moved to production and is currently supporting a business function. A system under development can be in the initiation phase, the development / acquisition phase, or the implementation phase. A system defined as undergoing a major modification is undergoing a major transition or conversion.

Operational
For an application system identified as operational, the SSP and / or IS RA and the supporting documentation will drive the application testing requirements.

Under Development
If an application system is identified as under development, determine the current phase of development as defined by the *CMS SDLC Methodology* (System Development Life Cycle). For the purpose of application security testing, the relevant CMS SDLC development phases are "Design and Engineering", "Development", "Testing", and "Implementation".

Undergoing a Major Modification
For a system undergoing a major modification, review the process that was followed to ensure that security controls were included in the design phase and carried through the life-cycle of the modification. Also ensure the previous system security plans and other system documentation were evaluated and updated appropriately.

2.1.8 MEET WITH KEY PERSONNEL

After collecting and reviewing all relevant application documentation, an introductory meeting shall be held to define the scope of the testing engagement and delineate the logical and organizational boundaries of the application. The attendees of this meeting will be CMS and, if applicable, contractor personnel responsible for 1) managing the business function, and 2) the information security of the technical application supporting the business function.

The management, business owners, security team, technical support and operations personnel of the business function and the application shall be present to gain an understanding of the purpose, methodology and scope of the review and the subsequent application security test to be developed. They will also be available to give perspective on any issues encountered during the previous stages of this review and identify the most qualified and responsible individual(s) to answer any questions.

During this meeting, the evaluator and CMS shall determine whether there will be a need to interview CMS and / or contractor personnel during the review, and, if so, CMS shall identify and schedule the appropriate personnel. Personnel to be interviewed shall be identified by their functional responsibilities.

2.1.9 REVIEW PERSONNEL ROLES AND RESPONSIBILITIES

The responsibility for securing CMS' computer systems ultimately rests with senior management; however, anyone who can affect computer security is also obligated to contribute to maintenance of the confidentiality, integrity, and availability of the system's information.

Review Relevant Roles and Responsibilities Documentation
Review the available documentation for the following security-related personnel security controls. While they apply to all roles, the level of the control will vary according to the level of responsibility and the exact nature of the position. The following shall be considered when reviewing the roles and responsibilities documentation:

- References are to be verified and background checks are to be conducted when evaluating prospective employees;
- Periodic re-investigations are conducted on employees;
- When granted access to sensitive information, employees and contractors are required to sign confidentiality and security agreements;
- Employees are required regularly to schedule vacations that exceed several days while their work is temporarily re-assigned;
- Termination and transfer procedures include: exit interviews; return of CMS property, keys, identification cards, passes, etc.; notification to security management and prompt revocation of User IDs; confirming the length of non-disclosure requirements and escorting terminated employees from CMS' premises;
- User access is restricted using the least privileged concept;
- Access authorizations are approved by senior management, documented on standard forms and kept on file;
- Procedures exist for revoking User IDs;
- Periodic reviews of the access authorization list by the system owner / manager ensure they are appropriate;
- Security managers review access authorizations and resolve any issues with the system owners;
- Audit trails for changes to security levels are to be created by security managers. Non-security management periodically reviews the changes;
- Audit trails are in place to track and hold users responsible and accountable for their activities; and
- Incompatible functions have been identified and different individuals are assigned to perform them.

Conduct Interviews
After reviewing the relevant documentation it may be necessary to interview key personnel for further clarification or additional information. Review the SSP and / or IS RA (Risk Assessment) to validate that all of the required functions are defined and addressed adequately. The application test plan and test scripts shall include test procedures to validate that the roles are staffed appropriately. The list of typical roles and responsibilities that follow is meant as a guide to assessing security personnel roles and their functions within CMS:

- Senior Management has the ultimate responsibility for the security of CMS' information systems. In order to support CMS' mission, it is management who sets the goals, priorities and objectives for an information security plan. It is also a management responsibility to be committed to the security plan and lead by example.
- The Computer Security Management team controls day-to-day computer security activities. These individuals are tasked with coordinating all computer security-related issues between the various elements within and without CMS.
- Typically, Application Owners / Managers are responsible for a business function and the supporting system. These managers usually have a technical support staff to assist them and they are responsible for management, technical and operational security controls. In larger computer systems a security officer may assist the application manager.
- System Administrators / Managers design, operate, and manage computer systems. They concern themselves with the implementation of the technical aspects of computer security. They are also responsible for ensuring the availability of their systems and guarding against, and assessing, threats to the system. They can be part of a larger Information Resource Management (IRM) team.
- Telecommunications staff is responsible for providing communications services including data, voice, fax and video. They have responsibility for the communications systems in much the same way the system managers have for their systems.
- The Systems Security Manager / Officer is responsible for day-to-day administration and implementation of security matters and frequently works with the system managers.

Analyze Roles and Responsibilities to Identify Relationships and Gaps
After reviewing relevant documentation and interviewing key personnel, the information gathered must be analyzed to identify inappropriate or incompatible duty assignments. This can occur when an individual employee has complete control over incompatible support functions or incompatible transaction functions. The test plan and test scripts will include procedures to validate that the documented controls are in place and enforced adequately.

In order to ensure that all the required security functions are being performed, review all the relevant documentation gathered, together with the results of interviews, and determine if any gaps exist in the coverage. This may occur when roles and responsibilities are not clearly defined or when a particular security function has been overlooked completely.

The same person should not perform more than one of certain systems support functions. The lack of independent oversight and verification can allow security controls and audit procedures to be compromised or bypassed, placing the system at risk. They include:

- Network Administration;
- Data control;
- Quality Assurance / testing;
- Data security;
- IS management;
- Data administration;
- System design;
- Production control and scheduling;

- Computer operations;
- Systems programming;
- Library management / change management; and
- Application management.

Certain combinations of transaction processing functions, if performed by the same individual, increase the risk of compromise, as security controls and checks can be by-passed due to the lack of independent verification or oversight. Specifically, the following combinations of functions should be segregated:

- Data entry and data verification;
- Data entry and the reconciliation of input data to output;
- Supervisory authorization functions and data entry (e.g., having the authority to permit a rejected entry to continue that would normally require a supervisor to review because the entry exceeded some limit); and
- The same individual completing the input for vendor invoices / purchasing and receiving purchase data is an example of incompatible input processing functions.

Since each application and project staffing is unique, the potential combinations of incompatible business functions vary. It is therefore important to understand fully the business mission in order to be able to identify incompatible duties and responsibilities.

2.1.10 REVIEW INFORMATION SECURITY BUSINESS RISK ASSESSMENT

The *CMS Information Security (IS) Business Risk Assessment (RA) Methodology* is the practical guide CMS business owners follow to conduct RAs of CMS business functions. Business risk management is a balance between the operational and economic costs of internal controls and safeguards, and the adequate protection of information confidentiality (including privacy), integrity, and availability that is necessary to complete the CMS business mission. Risk management has three components: risk assessment; risk mitigation; and evaluation and assessment. As CMS does not function in isolation, any external interdependencies are also included in the IS Business RA.

Review the *CMS Information Security (IS) Business Risk (RA) Report*, if one was previously conducted, for the business function supported by the application. This will aid in gaining an understanding of the business function supported by the application under review. Using the *CMS IS Business RA Methodology* as a guide, review the *IS Business RA Report* for the business function under review.

2.1.10.1 IDENTIFY BUSINESS FUNCTION INTERDEPENDENCIES

The *IS Business RA Report* for the business function supported by the application under review contains a list of business function interdependencies. Interdependences include other business functions to which the function under review *feeds* information, as well as other business functions that *feed* information to the function under review.

Obtain a list of all interconnected systems, including system identifiers where relevant. For each of the systems identified, request all security documentation and any documents governing rules of behavior. This documentation should include: the interconnected system's SSP; the interconnection Agreement; any authorization Agreements (MOUs, MOAs, ISAs, etc.); any reports from previous risk assessments or previous audit reports; and any other relevant information.

When reviewing this documentation the following factors should be considered:

- If the interconnected system does not have a security plan, identify and discuss any security concerns or controls that should be taken into account for securing the application under review.
- Verify that written authorization was obtained prior to establishing the interconnection and / or the sharing of sensitive information in the form of interconnection agreements, MOUs and MOAs.
- Ensure these documents detail all the rules of behavior and any legal requirements to be considered when sharing information.
- Any security control issues from the interconnection documentation should be included in the system security plan of the system under review.

2.1.10.2 REVIEW BUSINESS RISK FACTORS

The *IS Business RA Report* shall be reviewed to: identify the security concerns that exist for the business function; to verify that the list of threats and risks is complete; and to validate that the recommended safeguards are appropriate to reduce the risk to an acceptable level. Any threats or risks not sufficiently documented within the *IS Business RA Report*, or any inadequacies in the recommended safeguards, will be documented within the *Business and IS Risks Gap Analysis Report* (see Section 2.3). The test plan and test scripts shall include appropriate procedures to validate the effectiveness of the safeguards documented within the *IS Business RA Report*.

Business Function Documentation
This section of the *IS Business RA Report* includes background information describing the business function and the resources and information supported by the application. Review this section of the *IS Business RA Report* to determine:

- The business function name;
- The organization responsible for the application;
- The names of senior people responsible for the application;
- The assignment of application security responsibility;
- The business resources;
- The business function interdependencies;
- The operational environment and any special considerations;
- The system security level assessment; and
- The business function criticality assessment.

Risk Determination Table
This section of the *IS Business RA Report* describes the level of risk for each potential threat to the business function based on the likelihood of threat occurrence and the impact that the threat occurrence will have on the loss of CIA of the business function's information. Determine the completeness of the Risk Determination Table by examining the following elements:

- The list of threats identified for completeness;
- The risk description;
- The business impact;
- The strength of the internal controls;
- The likelihood of the occurrence;
- The severity of the impact; and
- The level of risk assigned.

Safeguard Determination Table
Review the Safeguard Determination Table to ensure the safeguards, additional controls and corrective actions adequately address the level of risk, and that all necessary safeguards have been identified. Examine the following for completeness and accuracy:

- The recommended safeguards, also called internal controls;
- The residual likelihood of occurrence of the threat;
- The severity of the residual impact; and
- The assigned residual risk level.

Implementation of Recommended Safeguards
Review the Implementation Analysis Phase Table for effectiveness, completeness and accuracy by examining the following as they apply to the business function supported by the application:

- The description of the risk;
- The business impact of the risk being exploited;
- The strength and effectiveness of the recommended control;
- The assigned risk level of the threat;
- The recommended safeguards;
- The implementation priority of the recommended safeguards; and
- The implementation rationale.

2.1.10.3 REVIEW THE RISK MITIGATION DOCUMENTATION

Risk mitigation is concerned with prioritizing, evaluating, and implementing the security controls identified by the risk assessment step. Since it is nearly impossible or impractical to eliminate all risk, it is the responsibility of senior management and system owners to implement the most effective, least-cost controls that have the smallest amount of disruptive effect on the business function's mission.

Risk Mitigation Options
The various options for risk mitigation are briefly discussed here. Review the documentation to establish which of the following options were specifically chosen by CMS:

- **Risk assumption** is the acceptance of the potential risk and implementation of the recommended controls or the continuation of operations without additional controls.
- **Risk avoidance** involves eliminating the risk by removing the cause (e.g., shut down the system at risk).
- **Risk limitation** seeks to limit the risk by implementing controls that contain and minimize the damage caused by the exploitation of a weakness.
- **Risk planning** manages risk by initiating a risk mitigation plan that ranks, implements and maintains controls.
- **Research and acknowledgement** is the process of acknowledging the vulnerability and researching controls to remedy the weakness.
- **Risk transference** uses other options, such as flood insurance, to transfer the risk.

Risk Mitigation Strategy
Review CMS' risk mitigation strategy, which is based on evaluating recommended control options and judging them for feasibility, effectiveness, and the potential cost or gain to the attacker and CMS.

Review the Control Implementation Approach
When implementing security controls the guiding principle is to mitigate the greatest threats with the least mitigation cost, while having the smallest impact on CMS' mission fulfillment ability.

Review the following activities:

- Prioritizing actions that produced an actions ranking of Low, Medium or High.
- The evaluation of recommended control options which produced a list of possible controls, based on the effectiveness and feasibility of implementing them.
- The analysis process that assessed the impact of implementing the controls as well as any other associated cost to produce the Cost-Benefit Analysis.
- The selection of controls by management using the Cost-Benefit Analysis to produce a list of selected controls.
- The assignment of responsibility for the implementation of the selected controls.
- The development of the safeguard implementation plan that prioritizes the implementation actions and projects the start and target completion dates, along with the controls selected, persons assigned and maintenance requirements.
- The implementation of the selected controls that lower the risk, but result in residual risk.

Review Residual Risk
Residual risk is the risk remaining after the implementation of enhanced or new security controls. It is not possible to eliminate completely all risk even after going through risk mitigation activities. Examine the controls implemented to determine if the risk has been reduced to an acceptable level or if more security controls are needed.

Review the Cost-Benefit Analysis
A Cost-Benefit Analysis should have been conducted on the selected controls after they were evaluated for feasibility and effectiveness. The Cost-Benefit Analysis allows CMS to allocate resources and implement controls in a cost-effective way and helps to identify the most appropriate controls. The Cost-Benefit Analysis should have considered the following for each of the controls selected:

- The impact of implementing the new or enhanced control;
- The impact of not implementing the control; and
- The costs of implementation, which include training costs, maintenance costs, new hardware or software costs, effect on system performance, and cost of new personnel.

Review Control Categories
Recommended controls can be management, operational, technical, or a combination of these. Choosing the appropriate security control can help limit or eliminate harm to CMS' mission, but the cost of implementation should always be weighed against the benefits of implementation. The control categories are:

1. Management controls are mainly concerned with the formulation of policies, procedures and standards to protect CMS' information. Management security controls can be preventive, which are concerned with the assignment of responsibility for security, the development, implementation and awareness training of the security plan. Detection management controls focus on conducting on-going Risk Assessments, reviewing security controls and auditing the system periodically. Recovery management controls are concerned with ensuring the continuity of business operations after an emergency or disasters, and incident response capabilities to recognize, report and react to incidents affecting the information system.
2. Operational controls are designed to protect the information system's operational capabilities and protect the system from the exploitation of vulnerabilities. Operational security controls are divided into preventive and detection controls. Preventive controls address issues such as controlling media access and disposal, controlling software viruses, providing emergency power, and off-site storage procedures and policies. Detection security controls include ensuring the environmental safety of CMS' operations and the physical security of CMS.
3. Technical controls are grouped according to their primary purpose into supporting controls, preventive controls and detection and recovery controls.

2.1.10.4 ANALYZE THE MANAGEMENT, OPERATIONAL, AND TECHNICAL SECURITY CONTROLS

Review the recommended and implemented security controls, focusing on the identified threats and related vulnerabilities, and the risk to CMS' information systems. As business systems are constantly changing, new threats might have emerged due to enhancements or new technology being introduced into the business environment. Ensure any new threats have been identified and addressed.

After reviewing the threats and vulnerabilities for completeness, analyze the recommended security controls with the goal of ensuring that the risk mitigation strategy is appropriate to the business risk. The acceptance of residual risk must be made in accordance with the level of acceptable risk commensurate with the system requirements of CIA.

2.1.10.5 VALIDATE ROLES AND RESPONSIBILITIES

Authorize Processing
The authorization granted by senior management officials for a system to process information is known as authorize processing. The manager authorizing processing also accepts the risk associated with the system.

Obtain the date of authorization, name, and title of the management official responsible for authorizing processing of the system. If the system is not yet authorized, obtain the name and title of the management official requesting the authorization processing.

Planning for Security in the Life-Cycle
Validate that appropriate personnel are assigned to plan for information security throughout the SDLC. Personnel roles shall be clearly defined, and individual duties clearly established. Appropriate personnel shall be assigned to perform specific security-related functions during each phase of the SDLC. Validate that personnel roles are properly assigned and documented based upon the *CMS SDLC Methodology*, and that appropriate work product has been generated during each of the SDLC phases for the system or application under review.

2.2 REVIEW SECURITY DOCUMENTATION

Information security documentation shall be reviewed to gain an understanding of the technical environment and risks, and to identify information security control requirements. Documented security control requirements will drive development of the test plan and test scripts, and will be evaluated and / or validated during the infrastructure or application test. Budgetary, personnel, and technical resource limitations render it impractical to evaluate all security controls at one time, so a phased approach must be utilized, whereby large or complex systems and applications can, if needed due to resource constraints, be tested in phases over a one-to-three year period. In order to implement properly a phased approach to security testing, it is necessary to prioritize security controls. Those security controls with the highest priority will be evaluated during the initial phase. Security control documentation will provide the base from which to prioritize controls and develop a multi-phased security testing strategy.

2.2.1 INFORMATION SECURITY DOCUMENTATION RESOURCES

The IS RA for the system or application to be tested shall be reviewed. Using the *CMS Information Security (IS) Risk Assessment (RA) Methodology* as a guide, review the IS RA Report for the application under review. The *IS RA Report* shall be reviewed to identify the security concerns that exist for the application; to verify that the list of vulnerabilities; threats and risks is complete; and to validate that the recommended safeguards are appropriate to reduce risk to an acceptable level. Any vulnerabilities, threats or risks not sufficiently documented within the *IS RA report*, or any inadequacies in the recommended safeguards, will be

documented within the *Business and IS Risks Gap Analysis Report* (see Section 2.3). The test plan and test scripts shall include appropriate procedures to validate the effectiveness of the safeguards documented within the IS RA Report.

Identify the System Security Level assigned and any system or application interdependencies. Review the system or application risks and safeguards in place to reduce or eliminate each risk. The test plan and test scripts shall include procedures to validate that safeguards documented within the IS Risk Assessment are implemented and effective in reducing risk to an acceptable level.

The primary resource for security control documentation will generally be the SSP covering the system or application under review from which controls will be prioritized. The IS RA will be the primary resource for security control documentation for applications that are not required to have a SSP. All other documentation containing security controls descriptions and requirements, including but not limited to, technical design documentation, network, system, and application diagrams, Interconnection Security Agreements, and system architecture documentation, shall be reviewed as well.

Organizational information security documentation that directly relates to, or affects, the security controls required for the system or application shall be reviewed as well. Organizational security documentation includes, but is not limited to, the *CMS Information Security Acceptable Risk Safeguards* (ARS), *Core Security Requirements* (CSR), NIST requirements, Federal Information System Controls Audit Manual (FISCAM) requirements, information security standards and procedures, disaster recovery plans, contingency / continuity of operations plans, incident response plans, and other documentation covering the organization as a whole.

2.2.2 INFORMATION SECURITY CONTROLS EVALUATION BASELINE

The SSP, IS RA, and any other documentation resources that establish security control requirements shall be reviewed against the criteria in this section. The following tables contain the seventeen (17) security control categories described in *NIST Special Publication 800-53* and the minimum-security controls that are required for each category. The security control categories are grouped into Management, Operational, and Technical controls, and do not follow the exact ordering of the NIST document, however, the categories and control types mirror the NIST guidance. The evaluator shall review the SSP, IS RA and other information security documentation to:

- Determine what types of controls are required for the system or application under review;
- Identify what control parameters are specified;
- Determine the expected adequacy of the controls; and
- Evaluate the documented security control requirements against the NIST guidance.

Development of the test plan and test scripts will depend, in large part, upon the security control requirements documented within the SSP and / or IS RA, because test procedures will be required to evaluate the presence and effectiveness of actual controls, as compared to the

documented requirements. Any gaps between the NIST guidance, as presented in the following tables, and the documented control requirements will feed the Gap Analysis Report.

Table 1: Management Security Controls

Security Control Category	Types of Controls (Minimum Required Controls)
Risk Assessment	**Risk Assessment Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented Risk Assessment Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Risk Assessment Policy and associated risk assessment controls. **Security Categorization**: The organization categorizes the information system and the information processed, stored, or transmitted by the system in accordance with FIPS 199 and documents the results (including supporting rationale) in the system security plan. Designated senior-level officials within the organization review and approve the security categorizations. **Risk Assessment**: The organization conducts assessments of the risk and magnitude of harm that could result from the unauthorized access, use, disclosure, disruption, modification, or destruction of information and information systems that support the operations and assets of CMS. **Risk Assessment Update**: The organization updates the risk assessment on a regular basis, or whenever there are significant changes to the information system, the facilities where the system resides, or other conditions that may impact the security or accreditation status of the system. **Vulnerability Scanning**: Using appropriate vulnerability scanning tools and techniques, the organization scans for vulnerabilities in the information system on a regular basis.
Planning	**Security Planning Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Security Planning Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Security Planning Policy and associated security planning controls.

Security Control Category	Types of Controls (Minimum Required Controls)
	System Security Plan: The organization develops and implements a System Security Plan (SSP) for the information system that provides an overview of the security requirements for the system and a description of the security controls in place or planned for meeting those requirements. Designated officials within the organization review and approve the Plan. **System Security Plan Update**: The organization reviews the SSP for the information system on a regular basis and revises the Plan to address system / organizational changes or problems identified during plan implementation or security control assessments. **Rules of Behavior**: The organization establishes and makes readily available to all information system users a set of rules that describes their responsibilities and expected behavior with regard to information system usage. The organization receives written acknowledgement from users indicating that they have read, understand, and agree to abide by the rules of behavior, prior to authorizing access to the information system. **Privacy Impact Assessment**: The organization conducts a Privacy Impact Assessment (PIA) on the information system.
System and Services Acquisition	**System and Services Acquisition Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, System and Services Acquisition Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the System and Services Acquisition Policy and associated system and services acquisition controls. **Allocation of Resources**: The organization determines, documents, and allocates as part of its capital planning and investment control process the resources required to protect the information system adequately. **Life-Cycle Support**: The organization manages the information system using a System Development Life-Cycle (SDLC) methodology. **Acquisitions**: The organization includes security requirements and / or security specifications, either explicitly or by reference, in information system acquisition contracts based on

Security Control Category	Types of Controls (Minimum Required Controls)
	an assessment of risk. **Information System Documentation**: The organization ensures that adequate documentation for the information system and its constituent components is available, protects the documentation when required, and distributes the documentation only to authorized personnel. **Software Usage Restrictions**: The organization complies with software usage restrictions. **User Installed Software**: The organization enforces explicit rules governing the downloading and installation of software by users. **System Design Principles**: The organization designs and implements the information system using security-engineering principles. **Outsourced Information System Services**: The organization ensures that third-party providers of information system services employ adequate security controls in accordance with applicable federal laws, directives, policies, regulations, standards, and guidance. The organization monitors security control compliance. **Developer Configuration Management**: The information system developer creates and implements a configuration management plan that controls changes to the system, including security control changes during development, tracks security flaws, requires authorization of changes, and provides documentation of the Plan and its implementation. **Developer Security Testing**: The information system developer creates a security test and evaluation plan, implements the plan, and documents the results. Developmental security test results may be used in support of the security Certification & Accreditation process for the delivered information system.
Certification, Accreditation, and Security Assessments	**Certification & Accreditation (C&A) and Security Assessment Policy (SAP) and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) formal, documented, security assessment and C&A policies that address purpose, scope, roles, responsibilities, and

Security Control Category	Types of Controls (Minimum Required Controls)
	compliance; and (ii) formal, documented procedures to facilitate the implementation of the Security Assessment and C&A policies and associated assessment, certification, and accreditation controls. **System Assessments**: In support of the continuous monitoring process, the organization conducts assessments of the security controls in the information system at least once a year to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system. **Information System Connections**: The organization authorizes all connections from the information system to other information systems outside of the accreditation boundary and monitors / controls the system interconnections on an on-going basis. **Security Certification**: In support of the security accreditation process, the organization conducts an assessment of the security controls in the information system to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system. **Plan of Action and Milestones**: The organization develops and updates on a regular basis, a Plan of Action & Milestones (POA&M) for the information system that documents the organization's planned, implemented, and evaluated remedial actions to correct any deficiencies noted during the assessment of the security controls and to reduce or eliminate known vulnerabilities in the system. **Security Accreditation**: The organization authorizes (i.e., accredits) the information system for processing prior to operations and updates the authorization on a regular basis. A senior organizational official signs and approves the Security Accreditation. **Continuous Monitoring**: The organization monitors the security controls in the information system on an on-going basis.

Table 2: Operational Security Controls

Security Control Category	Types of Controls
Personnel Security	**Personnel Security Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, personnel security policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Personnel Security Policy and associated personnel security controls. **Position Categorization**: The organization assigns a risk designation to all positions and establishes screening criteria for individuals filling those positions. The organization reviews and revises position risk designations on a regular basis. **Personnel Screening**: The organization screens individuals requiring access to organizational information and information systems prior to authorizing access. **Personnel Termination**: When employment is terminated, the organization terminates information system access, conducts exit interviews, ensures the return of all organizational information system-related property (e.g., keys, identification cards, building passes), and ensures organization has access to official records created by the employee that are stored on organizational information systems. **Personnel Transfer**: The organization reviews information systems / facilities access authorizations when individuals are re-assigned or transferred to other positions within the organization and initiates appropriate actions (e.g., re-issuing keys, identification cards, building passes; closing old accounts and establishing new accounts; and changing system access authorizations). **Access Agreements**: The organization completes appropriate access agreements (e.g., non-disclosure agreements, acceptable use agreements, rules of behavior, conflict-of-interest agreements) for individuals requiring access to organizational information and information systems prior to authorizing access. **Third Party Personnel Security**: The organization establishes personnel security requirements for third-party providers (e.g., service bureaus, contractors, and other organizations providing information system development,

Security Control Category	Types of Controls
	information technology services, outsourced applications, network and security management) and monitors provider compliance to ensure adequate security. **Personnel Sanctions**: The organization employs a formal sanctions process for personnel failing to comply with established information security policies and procedures.
Physical and Environmental Protection	**Physical and Environmental Protection Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Physical and Environmental Protection Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Physical and Environmental Protection Policy and associated physical and environmental protection controls. **Physical Access Authorizations**: The organization develops, and keeps current, lists of personnel with authorized access to facilities containing information systems and issues appropriate authorization credentials (e.g., badges, identification cards, smart cards). Designated officials within the organization review and approve the access list and authorization credentials on a regular basis. **Physical Access Control**: The organization controls all physical access points (including designated entry / exit points) to facilities containing information systems and verifies individual access authorizations before granting access to the facilities. The organization also controls access to areas officially designated as publicly accessible, as appropriate, in accordance with the organization's assessment of risk. **Access Control for Transmission Medium**: The organization controls physical access to information system transmission lines carrying unencrypted information to prevent eavesdropping, in-transit modification, disruption, or physical tampering. **Access Control for Display Medium**: The organization controls physical access to information system devices that display information in order to prevent unauthorized individuals from observing the display output. **Monitoring Physical Access**: The organization monitors

Security Control Category	Types of Controls
	physical access to information systems in order to detect and respond to incidents. **Visitor Control**: The organization controls physical access to information systems by authenticating visitors (including government contractors) prior to authorizing access to facilities or areas other than those designated as publicly accessible. **Access Logs**: The organization maintains a visitor access log that includes: (i) name and organization of the person visiting; (ii) signature of the visitor; (iii) form of identification; (iv) date of access; (v) time of entry and departure; (vi) purpose of visit; and (vii) name and organization of person visited. Designated officials within the organization review the access logs on a regular basis after closeout. **Power Equipment and Cabling**: The organization protects power equipment and cabling for the information system from damage and destruction. **Emergency Shutoff**: For specific locations within a facility containing concentrations of information system resources (e.g., Data Centers, server rooms, mainframe rooms), the organization provides the capability to shut off power to any information technology component that may be malfunctioning (e.g., due to an electrical fire) or threatened (e.g., due to a water leak) without endangering personnel by requiring them to approach the equipment. **Emergency Power**: The organization provides a short-term uninterruptible power supply to facilitate an orderly shutdown of the information system in the event of a primary power source loss. **Emergency Lighting**: The organization employs and maintains automatic emergency lighting systems that cover emergency exits and evacuation routes and that activate in the event of a power outage or disruption. **Fire Protection**: The organization employs and maintains fire suppression and prevention devices / systems that can be activated in the event of a fire. **Temperature and Humidity Controls**: The organization regularly maintains within acceptable levels and monitors the

Security Control Category	Types of Controls
	temperature and humidity within facilities containing information systems. **Water Damage Protection**: The organization protects the information system from water damage resulting from broken plumbing lines or other sources of water leakage by ensuring that master shutoff valves are accessible, working properly, and known to key personnel. **Delivery and Removal**: The organization controls information system-related items (i.e., hardware, firmware, software) entering and exiting the facility and maintains appropriate records of those items. **Alternative Worksite**: Individuals within the organization employ appropriate information system security controls at alternate work sites.
Contingency Planning	**Contingency Planning Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, contingency planning policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Contingency Planning Policy and associated contingency planning controls. **Contingency Plan**: The organization develops and implements a Contingency Plan for the information system addressing contingency roles, responsibilities, assigned individuals with contact information, and activities associated with restoring the system after a disruption or failure. Designated officials within the organization review and approve the Contingency Plan and distribute multiple copies of the Plan to key contingency personnel. **Contingency Training**: The organization trains personnel in their roles and responsibilities with respect to the Contingency Plan as it applies to the information system and provides refresher training at least once per year. **Contingency Plan Testing**: The organization tests the Contingency Plan for the information system at least once per year to determine the Plan's effectiveness and the organization's readiness to execute the Plan. Appropriate

Security Control Category	Types of Controls
	officials within the organization review the Contingency Plan test results and initiate corrective actions. **Contingency Plan Update**: The organization reviews the Contingency Plan for the information system at least once a year and revises the plan to address system / organizational changes or problems encountered during plan implementation, execution, or testing. **Alternate Storage Sites**: The organization identifies an alternate storage site and initiates necessary agreements to permit the storage of information system backup information. **Alternate Processing Site**: The organization identifies an alternate processing site and initiates necessary agreements to permit the resumption of information system operations for critical mission / business functions within a reasonable timeframe when the primary processing capabilities are unavailable. **Telecommunications Services**: The organization identifies primary and alternate telecommunications services to support the information system, and initiates necessary agreements to permit the resumption of system operations for critical mission / business functions within a reasonable timeframe when primary telecommunications capabilities are unavailable. **Information System Backup**: The organization conducts regularly scheduled backups of user-level and system-level information (including system state information) contained in the information system and stores backup information at an appropriately secured location. **Information System Recovery and Reconstitution**: The organization employs mechanisms with supporting procedures to allow the information system to be recovered and reconstituted to the system's original state after a disruption or failure.
Configuration Management	**Configuration Management Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, configuration management policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented

Security Control Category	Types of Controls
	procedures to facilitate the implementation of the configuration management policy and associated configuration management controls. **Baseline Configuration**: The organization develops, documents, and maintains a current, baseline configuration of the information system and an inventory of the system's constituent components. **Configuration Change Control**: The organization documents and controls changes to the information system. Appropriate organizational officials approve information system changes in accordance with organizational policies and procedures. **Monitoring Configuration Changes**: The organization monitors changes to the information system and conducts security impact analyses to determine the effects of the changes. **Access Restrictions for Change**: The organization enforces access restrictions associated with changes to the information system. **Configuration Settings**: The organization configures the default security settings of information technology products to the most restrictive mode consistent with information system operational requirements. The organization configures the information system to provide only essential capabilities and specifically prohibits the use of unnecessary or unauthorized ports, protocols, and / or services. **Least Functionality**: The organization configures the information system to provide only essential capabilities and specifically prohibits and / or restricts the use of unauthorized functions, ports, protocols, and / or services.
Maintenance	**System Maintenance Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, information System Maintenance Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the information System Maintenance Policy and associated system maintenance controls.

Security Control Category	Types of Controls
	Periodic Maintenance: The organization schedules, performs, and documents routine preventive and regular maintenance on the components of the information system in accordance with manufacture / vendor specifications and / or organizational requirements. **Maintenance Tools**: The organization approves, controls, and monitors the use of information system maintenance tools and maintains the tools on an on-going basis. **Remote Maintenance**: The organization approves, controls, and monitors remotely executed maintenance and diagnostic activities. **Maintenance Personnel**: The organization maintains a list of individuals authorized to perform maintenance on the information system. Only authorized individuals perform maintenance on the information system. **Timely Maintenance**: The organization obtains maintenance support and spare parts for critical information system components within a reasonable timeframe following failure.
System and Information Integrity	**System and Information Integrity Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, System and Information Integrity Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the System and Information Integrity Policy and associated system and information integrity controls. **Flaw Remediation**: The organization identifies, reports, and corrects information system flaws. **Malicious Code Protection**: The information system implements malicious code protection that includes a capability for automatic updates. **Intrusion Detection Tools and Techniques**: The organization employs tools and techniques to monitor events on the information system, detect attacks, and provide identification of unauthorized use of the system. **Security Alerts and Advisories**: The organization receives

Security Control Category	Types of Controls
	information system security alerts / advisories on a regular basis, issues alerts / advisories to appropriate personnel, and takes appropriate actions in response. **Security Functionality Verification**: The information system automatically verifies the correct operation of security functions and takes appropriate action when anomalies are discovered. **Software and Information Integrity**: The information system detects and protects against unauthorized changes to software and information. **Spam and Spyware Protection**: The information system implements spam and spyware protection. **Information Input Restrictions**: The organization restricts the information input to the information system to authorized personnel only. **Information Accuracy, Completeness, and Validity**: The organization checks the information input to the information system for accuracy, completeness, and validity. **Information Input Error Handling**: The organization corrects and resubmits erroneous input to the information system. **Information Processing Error Handling**: The organization identifies erroneous information system transactions before processing to minimize disruption of valid transaction processing. **Information Output Error Handling**: The organization reviews outputs from information system application programs for accuracy and controls errors contained in the outputs. **Information Output Handling and Retention**: The organization handles and retains output from information systems in accordance with organizational policy and operational requirements.
Media Protection	**Media Protection Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Media Protection Policy that addresses

Security Control Category	Types of Controls
	purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Media Protection Policy and associated media protection controls. **Media Access**: The organization ensures that only authorized users have access to information in printed form or on digital media removed from the information system. **Media Labeling**: The organization affixes external labels to removable information storage media and information system output indicating the distribution limitations and handling caveats of the information. The organization may exempt certain types of media or hardware components from labeling so long as they remain within a secure environment. **Media Storage**: The organization physically controls and securely stores information system media, both paper and electronic, based on the highest FIPS 199 security category of the information recorded on the media. **Media Transport**: The organization controls information system media (paper and electronic) and restricts the pickup, receipt, transfer, and delivery of such media to authorized personnel. **Media Sanitization**: The organization sanitizes information system magnetic media using approved equipment, techniques, and procedures. The organization tracks, documents, and verifies media sanitization actions and periodically tests sanitization equipment / procedures to ensure correct performance. **Media Destruction and Disposal**: The organization sanitizes or destroys information system digital media prior to its disposal or release for reuse outside the organization, to prevent unauthorized individuals from gaining access to and using the information contained on the media.
Incident Response	**Incident Response Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Incident Response Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation

Security Control Category	Types of Controls
	of the Incident Response Policy and associated incident response controls. **Incident Response Training**: The organization trains personnel in their incident response roles and responsibilities with respect to the information system and provides refresher training at least once per year. **Incident Response Testing**: The organization tests the incident response capability for the information system at least once a year to determine the plan's effectiveness and documents the results. **Incident Handling**: The organization implements an incident handling capability for security incidents that includes preparation, detection and analysis, containment, eradication, and recovery. **Incident Monitoring**: The organization tracks and documents information system security incidents on an on-going basis. **Incident Reporting**: The organization promptly reports incident information to appropriate authorities. **Incident Response Assistance**: The organization provides an incident support resource that offers advice and assistance to users of the information system for the handling and reporting of security incidents. The support resource is an integral part of the organization's incident response capability.
Awareness and Training	**Security Awareness and Training Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Security Awareness and Training Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Security Awareness and Training Policy and associated security awareness and training controls. **Security Awareness**: The organization trains all personnel (including managers and senior executives) in basic information system security awareness prior to authorizing access to the system and at least once a year thereafter. **Security Training**: The organization identifies personnel with

Security Control Category	Types of Controls
	significant information system security roles and responsibilities, documents those roles and responsibilities, and provides appropriate information system security training prior to authorizing access to the system and on a regular basis thereafter. **Security Training Records**: The organization documents and monitors individual information system security training activities including basic security awareness training and specific information system security training.

Table 3: Technical Security Controls

Security Control Category	Types of Controls
Identification and Authentication	**Identification and Authentication Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Identification and Authentication Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Identification and Authentication Policy and associated identification and authentication controls. **User Identification and Authentication**: The information system accurately identifies and authenticates users (or processes acting on behalf of users). **Device and Host Identification and Authentication**: The information system identifies and authenticates specific devices before establishing connections. **Identifier Management**: The organization manages user identifiers by: (i) uniquely identifying each user; (ii) verifying the identity of each user; (iii) receiving authorization to issue a user identifier from an appropriate organization official; (iv) ensuring that the user identifier is issued to the intended party; (v) disabling the user identifier after a pre-defined period of inactivity; and (vi) archiving user identifiers. **Authenticator Management**: The organization manages information system authenticators (e.g., tokens, passwords, key cards) by: (i) defining initial authenticator content; (ii) establishing administrative procedures for initial authenticator distribution, for lost / compromised or damaged authenticators,

Security Control Category	Types of Controls
	and for revoking authenticators; and (iii) changing default authenticators upon information system installation. **Authenticator Feedback**: The information system provides feedback to a user during an attempted authentication that does not weaken the strength of the authentication mechanism. **Cryptographic Module Authentication**: For authentication to a cryptographic module, the information system employs authentication methods that meet the requirements of FIPS 140-2.
Access Control	**Access Control Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Access Control Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Access Control Policy and associated access controls. **Account Management**: The organization manages information system accounts, including establishing, activating, modifying, reviewing, disabling, and removing accounts. The organization reviews information system accounts on a regular basis. **Account Management**: The information system enforces assigned authorizations for controlling access to the system in accordance with applicable policy. **Information Flow Enforcement**: The information system enforces assigned authorizations for controlling the flow of information within the system in accordance with applicable policy. **Separation of Duties**: The information system enforces separation of duties through assigned access authorizations. **Least Privilege**: The information system enforces the most restrictive set of rights / privileges or accesses needed by users (or processes acting on behalf of users) for the performance of specified tasks. **Unsuccessful Log-on Attempts**: The information system enforces a limit of a certain number of consecutive invalid access attempts by a user during a predetermined time period.

Security Control Category	Types of Controls
	The information system automatically restricts access to the user account when the maximum number of unsuccessful attempts is exceeded. **Privacy Policy Notification**: The information system displays the organization's privacy policy prior to granting system access. **Previous Log-on Notification**: The information system notifies the user, upon successful log-on, of the date and time of the last log-on, the location of the last log-on, and the number of unsuccessful log-on attempts since the last successful log-on. **Concurrent Session Control**: The information system limits the number of concurrent sessions for any user to a reasonable number. **Session Lock**: The information system prevents further access to the system by initiating a session lock that remains in effect until the user re-establishes access using appropriate identification and authentication procedures. **Session Termination**: The information system automatically terminates a session after a reasonable period of inactivity. **Supervision and Review – Access Control**: The organization supervises and reviews the activities of users with respect to the enforcement and usage of information system access controls. **Permitted Actions without Identification and Authentication**: The organization identifies specific user actions that can be performed on the information system without identification or authentication. **Automated Marking**: The information system marks output using standard naming conventions to identify any special dissemination, handling, or distribution instructions. **Automated Labeling**: The information system appropriately labels information in storage, in process, and in transmission. **Remote Access**: The organization documents, monitors, and controls all methods of remote access (e.g., dial-up, Internet) to

Security Control Category	Types of Controls
	the information system including remote access for privileged functions. Appropriate organization officials authorize each remote access method for the information system and authorize only necessary users for each access method. **Wireless Access Restrictions**: The organization documents, monitors, and controls wireless access to the information system. **Access Control for Portable and Mobile Systems**: The organization establishes connection criteria for allowing portable and mobile information systems access to organizational networks. **Personally Owned Information Systems**: The organization restricts the use of personally-owned information systems for official U.S. Government business involving the processing, storage, or transmission of federal information.
Audit and Accountability	**Audit and Accountability Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, Audit and Accountability Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the Audit and Accountability Policy and associated audit and accountability controls. **Auditable Events**: The information system generates audit records for a defined set of events as defined in ARS policy. **Content of Audit Records**: The information system captures sufficient information in audit records to establish what events occurred, the sources of the events, and the outcomes of the events as described in the SSP. Audit Logs: A chronological record of system activities which enables the reconstruction and examination of the sequence of events and activities surrounding or leading to an operation, a procedure or an event in a transaction from its inception to final results. The audit log also serves as the chain of custody for the history of use of a record. This term is synonymous with **Audit Records** and **Audit Trail**. **Audit Storage Capacity**: The organization allocates sufficient

Security Control Category	Types of Controls
	audit record storage capacity to reduce the potential for such capacity being exceeded. **Audit Processing**: In the event of an audit failure or audit storage capacity is reached, the information system alerts appropriate organizational officials and automatically takes appropriate action. **Audit Monitoring, Analysis, and Reporting**: The organization regularly reviews / analyzes audit records for indications of inappropriate or unusual activity; investigates suspicious activity or suspected violations; and reports findings to appropriate officials in accordance with policy. The organization investigates suspicious activities on the information system and takes appropriate actions. **Audit Reduction and Report Generation**: The information system provides an audit reduction and report generation capability. **Time Stamps**: The information system provides time stamps for use in audit record generation. **Protection of Audit Information**: The information system protects audit information and audit tools from unauthorized access, modification, and deletion. **Non-Repudiation**: The information system provides the capability to determine whether a given individual took a particular action (e.g., created information, sent a message, approved information [e.g., to indicate concurrence or sign a contract] or received a message). **Audit Retention**: The organization retains audit logs for a sufficient length of time to provide support for after-the-fact investigations of security incidents and to meet regulatory and organizational information retention requirements.
System and Communications Protection	**System and Communications Protection Policy and Procedures**: The organization develops, disseminates, and reviews / updates periodically: (i) a formal, documented, System and Communications Protection Policy that addresses purpose, scope, roles, responsibilities, and compliance; and (ii) formal, documented procedures to facilitate the implementation of the System and Communications Protection Policy and

Security Control Category	Types of Controls
	associated system and communications protection controls. **Application Partitioning**: The information system separates user functionality (including user interface services) from information system management functionality. **System Function Isolation**: The information system isolates security functions from non-security functions. **Information Remnants**: The information system prevents unauthorized and unintended information transfer via shared system resources. **Denial-of-Service Protection**: The information system reasonably protects against Denial-of-Service attacks. **Resource Priority**: The information system limits the use of resources by priority. **Boundary Protection**: The information system monitors and controls communications at the external boundary of the information system and at key internal boundaries within the system. **Transmission Integrity**: The information system protects the integrity of transmitted information. **Transmission Confidentiality**: The information system protects the confidentiality and privacy of transmitted information. **Network Disconnect**: The information system terminates a network connection at the end of a session or after a reasonable period of inactivity. **Trusted Path**: The information system establishes a trusted communications path between the user and the security functionality of the system. **Cryptographic Key Establishment and Management**: The information system employs automated mechanisms with supporting procedures or manual procedures for cryptographic key establishment and key management. **Use of Validated Cryptography**: When cryptography is

Security Control Category	Types of Controls
	employed within the information system, the system performs all cryptographic operations (including key generation) using FIPS 140-2 validated cryptographic modules operating in approved modes of operation. **Public Access Protections**: For publicly available systems, the information system protects the integrity of the information and applications. **Collaborative Computing**: The information system prohibits remote activation of collaborative computing mechanisms (e.g., video and audio conferencing) and provides an explicit indication of use to the local users (e.g., use of camera or microphone). **Transmission of Security Parameters**: The information system reliably associates security parameters (e.g., security labels and markings) with information exchanged between information systems. **Public Key Infrastructure Certificates**: The organization develops and implements a certificate policy and certification practice statement for the issuance of public key certificates used in the information system. **Mobile Code**: The organization restricts the deployment of mobile code based on its potential to cause damage to the information system if used maliciously. Appropriate organizational officials authorize the use of mobile code. **Voice Over Internet Protocol**: The organization restricts the use of Voice Over Internet Protocol (VOIP) technology based upon operational requirements. Appropriate organizational officials authorize the use of VOIP.

2.2.3 PRIORITIZE SECURITY CONTROLS

Based upon the documented security control requirements, the seventeen (17) security control categories shall be prioritized. Each control category shall be assigned a priority of High, Medium, or Low. In order to ensure that time and other resources are allocated to the most critical control categories, the first to be evaluated and validated shall be the High-priority controls, which, if not adequate and effective, may result in the greatest potential harm. Medium-priority controls shall be evaluated during the subsequent testing phase, and Low-

priority controls during the last phase. To prioritize controls, the following factors shall be considered:

1. Sensitivity requirements of the system or application under review;
2. Criticality requirements of the system or application under review;
3. Business risks documented within the IS Business RA Report;
4. Technical risks documented within the IS RA Report; and
5. Potential harm that may result if the control category is inadequate or ineffective.

As a general rule, certain control categories, such as Access Control and Identification and Authentication, will be a higher priority than other categories, such as Awareness and Training and Maintenance. For example, Application "A" may be a financial application supporting a high-profile business function distributed among multiple sites. In the case of Application "A", which has high sensitivity and criticality requirements, ineffective security controls in the Access Control category may directly cause significant financial loss. Ineffective controls in the Awareness and Training category, however, assuming that other control categories are sufficient, would not cause as substantial a risk of financial loss.

In most cases, the actual prioritization of control categories will be system or application-specific. The individual sensitivity and criticality requirements of the business function supported by each system or application, the known business and technical risks, and the potential harm (in terms of financial loss, political damage, public embarrassment, information disclosure, and legal consequences) that CMS might experience will drive the control category prioritization process.

2.2.4 RECENT SECURITY TEST DOCUMENTATION

Documentation from recent security tests shall be reviewed to determine which controls and infrastructure components have been previously evaluated, and to identify vulnerabilities discovered during prior tests. Based upon the controls and infrastructure components previously evaluated, the test plan and test scripts shall be developed to minimize duplication of effort and reuse past results. In addition, the test plan and test scripts shall be developed to validate the effectiveness of completed corrective actions to close or reduce the impact of high-risk vulnerabilities discovered during prior tests, and to re-test open vulnerabilities not corrected.

The following sources of information shall be reviewed:

- ST&E reports;
- Audit reports;
- Vulnerability assessment / penetration test reports; and
- Self-assessment reports.

2.3 ANALYZE AND REPORT GAPS

A Gap Analysis shall be conducted to identify discrepancies between information security documentation resources. For example, the SSP may not contain adequate controls to address

the risks documented in the IS RA, the SSP controls may be inconsistent with CMS organization requirements or NIST guidance, or the components list in the SSP may not be consistent with the system diagram or architecture documentation.

2.3.1 BUSINESS AND IS RISKS GAP ANALYSIS REPORT

Determine whether any additional threats to the business function exist, beyond what is documented in the *IS Business RA Report*. Based upon any additional threats, identify and evaluate additional business risks. The *Business and IS Risks Gap Analysis Report* shall describe any gaps between the documented threats and risks, and any additional threats and risks not documented within the *IS Business RA Report*.

Determine whether any additional system or application threats or vulnerabilities exist, beyond what is documented in the IS RA Report. Based upon any additional threats or vulnerabilities, identify and evaluate additional risks. The *Business and IS Risks Gap Analysis Report* shall describe any gaps between the documented threats, vulnerabilities, and risks, and any additional threats, vulnerabilities, and risks not documented within the IS RA Report.

2.3.2 SECURITY CONTROLS GAP ANALYSIS REPORT

Identify any gaps between the security control requirements documented within the SSP, IS RA, and other security documentation, and the security control guidance detailed in NIST Special Publication 800-53 and the CMS ARS. Any control types or parameters less secure than the NIST guidance or CMS ARS requirements represent gaps that shall be reported in the *Security Controls Gap Analysis Report*. In addition, identify any gaps between the CMS ARS requirements and the CMS CSR requirements, and document these gaps within the *Security Controls Gap Analysis Report*.

Identify business and / or system risks reported in the IS Business RA and IS RA Reports that are not adequately addressed by management, operational, or technical security controls. The *Security Controls Gap Analysis Report* shall describe any gaps between controls required to address risks documented in the IS Business RA and IS RA Reports and actual security control requirements documented within the SSP / IS RA.

3 INFRASTRUCTURE SECURITY ASSESSMENT

A crucial decision point must be resolved before conducting or planning for the infrastructure security test. If the infrastructure's components directly supporting the application under review have been tested recently, and sufficient test results exist to make a determination as to the adequacy of the prior testing, a decision to use these results in lieu of another test shall be made. The independent contractor engaged to conduct security testing or CMS personnel (testing entity) shall conduct a technical analysis of the prior test scope and results to determine if the previous security testing is technically sufficient. The testing entity shall use independent professional judgment as to the adequacy of the prior testing, and recommend to CMS whether re-testing is required. CMS shall make the final decision as to whether re-testing of the infrastructure is required.

If it is decided that the infrastructure components have been adequately tested, the infrastructure security test is not required. In this case, the infrastructure security assessment should only include procedures necessary to verify or validate any open vulnerabilities or completed corrective actions, based upon the information gathered during the Business Process Review. If the infrastructure components have been previously tested and re-testing is not required, proceed to Section 4, Application Security Test.

If the infrastructure components directly supporting the application have not been tested, or the testing performed was not adequate to provide an acceptable level of assurance, infrastructure security testing shall be conducted, based upon the guidance in this section. The infrastructure security test plan and test scripts shall identify and describe test procedures necessary to validate the actual effectiveness of documented security controls (as documented in the SSP / IS RA), and to discover procedural and technical vulnerabilities and weaknesses. To conduct the infrastructure test, the testing entity shall execute the test plan and test scripts.

3.1 IDENTIFY RELEVANT TESTS AND TOOLS

The system platform and technical environment will drive the types of tests to be conducted, and the tools that will be employed to complete the tests. The testing entity shall identify the operating platform(s) for each of the infrastructure components directly supporting the application under review. This information was gathered during the Business Process Review, and further information may be documented within the *Business and IS Risks Gap Analysis Report* and the *Security Controls Gap Analysis Report*. Based upon the system platform, identify relevant tests to be conducted. The tests must reflect the relative priority of the security control categories (as determined during the Business Process Review), and testing shall focus initially on the categories of greatest priority. The identification and prioritization of security control for testing will be made by CMS, SSG through contract task order assignment and / or project scope requirement definitions. Relevant tests include procedures to verify and validate the effectiveness of management, operational, and technical security controls documented within the SSP / IS RA, and procedures to discover and identify procedural and technical vulnerabilities and threats not documented within the SSP / IS RA.

After identifying relevant tests that shall be conducted, identify the tools that will be employed to complete each test. Tools may include technical software, such as port and vulnerability scanners, as well as interview questionnaires and other non-technical instrumentalities that may be employed to gather information, identify vulnerabilities, and assess information security.

3.2 TEST PROCEDURE DOCUMENTATION

3.2.1 TEST PLAN

The test plan documents the processes and procedures that are to be conducted during the infrastructure security test. The test plan shall include the relevant test procedures identified during the previous step, and assign / apply relevant tools, methods, and personnel to achieve the test objective. The test plan shall define a progressive methodology for conducting the test; including that the first phase of testing is conducted with the least information and access, and subsequent phases of testing involve greater knowledge of the infrastructure components and increased access. For example, the first phase of testing may involve remote testing from the perspective of an unauthorized person. The next phase may then involve on-site testing from the perspective of an unauthorized person. The following phase would then involve on-site testing from the perspective of an authorized internal user, and the final phase would involve review of information or access provided to the testing entity by CMS, including firewall configuration(s), router configuration(s), server configuration, etc. This progressive methodology is also referred to as an “outside-in” strategy.

The test plan shall include a requirement whereby the testing entity is to validate the effectiveness of security controls documented within the SSP / IS RA. Specific procedures for conducting these validation checks will be documented within the test scripts. The test plan shall also include the processes to be employed to test for, discover and identify, and validate procedural and technical vulnerabilities in the infrastructure. These processes include, but are not limited to, port scanning, vulnerability scanning, password cracking / discovery, and manual penetration testing / access attempts. The test plan shall also include requirements to re-test open vulnerabilities and to validate completed corrective actions, and to validate the infrastructure component inventory.

The test plan shall be developed by the testing entity, and reviewed and authorized by CMS before the start of any testing. CMS will have an opportunity to comment on the test plan, and request that test procedures be added to, or deleted from, the test plan.

3.2.2 TEST SCRIPTS

The test scripts support the test plan by providing detailed criteria to be used in validating the implementation and effectiveness of documented security controls. The test scripts shall be designed to validate that the security controls documented within the SSP / IS RA are implemented, configured, and operate as expected. If, however, the *Security Controls Gap Analysis Report* reveals gaps between the documented security controls (SSP / IS RA) and the CMS ARS and CSR requirements, the test scripts should include the relevant ARS and / or CSR requirement. Test scripts are, in actuality, a pass / fail assessment system, whereby a certain expectation is made, based upon the documented security control requirements, and during the

infrastructure test the expectation is proven, through testing or other means, to be true or false. Completion of the test scripts may involve technical testing and review of server or device configuration, but may also involve personnel interviews and documentation review.

3.3 TEST PLAN AND TEST SCRIPT EXECUTION

3.3.1 COMPONENTS INVENTORY VALIDATION

The infrastructure components inventory, as documented in the SSP / IS RA, shall be validated against the actual infrastructure environment. The validation process will include personnel interviews, automated system and network scans, and physical review of the infrastructure supporting the application. Any discrepancies between the documented component inventory and the actual environment shall be documented as a finding.

3.3.2 MANUAL VALIDATION OF DOCUMENTED CONTROLS

The test scripts will include detailed criteria against which to measure the actual implementation and effectiveness of documented security controls. Based upon the prioritization of security control categories, determined during the Business Process Review, test procedures shall be conducted to validate the documented security controls operate as expected and required. The prioritization of controls will determine which control categories shall be validated during the current test. Subject to the prioritization of control categories, management, operational and technical security controls implemented for the infrastructure supporting the application shall be validated.

Validation of management controls requires the review of documentation, as well as personnel interviews. The testing entity validating the management controls shall review risk assessment, security planning, system / services acquisition, and certification and accreditation documentation to validate that the required documentation exists and that it is reasonably comprehensive and accurate. Personnel interviews shall supplement the documentation review, and will enable the evaluator to determine how the documented processes and requirements are followed and completed within the CMS environment.

Validation of operational controls requires a combination of documentation review, personnel interviews, and manual testing. Certain control categories, such as Physical and Environmental Protection and System and Information Integrity (see table in Section 2.2.2), will require some manual testing to validate that the security controls are implemented and operate effectively. For example, a physical security assessment is necessary to validate that door and window locks are in place and provide adequate security, and technical testing is necessary to validate that an Intrusion Detection System is implemented and operates as expected.

Technical security controls are validated primarily through the execution of automated and manual test procedures, but also through personnel interviews. To validate technical controls, the testing entity shall conduct port scanning, vulnerability scanning, and password cracking / discovery. The most effective and efficient way, however, to validate technical security controls is to review the operating system (Windows, UNIX, Novell, MVS, etc.) or application platform (Oracle, Apache, DB2, etc.) configuration from the server console.

During the manual validation of documented controls, the test script papers shall be completed. For each test or security check conducted, a pass / fail assessment shall be recorded on the test script document. The test scripts shall be signed by the tester assigned to conduct the testing and by the System Owner or other person responsible for the security of the infrastructure component. The completed and signed test scripts serve as a written record of the test process and acceptance of the test results.

3.3.3 VULNERABILITY DISCOVERY

After validating the security controls documented within the SSP / IS RA, the testing entity shall conduct vulnerability testing to discover and identify procedural and technical vulnerabilities in the infrastructure supporting the application. Technical vulnerabilities may be discovered through a broad range of tools and testing procedures. The following common test procedures and techniques shall be conducted, when appropriate:

- Attempt to access internal network hosts, including the mainframe(s), using common and default user accounts and passwords.
- Obtain a user account with no dataset access privileges, viewing the user account capabilities, and identifying sub-systems present in the configuration.
- Attempt to invoke the security package within MVS mainframes.
- Attempt to set up a new power user or super user account within MVS mainframes or distributed systems.
- Attempt to alter security software parameters within MVS mainframes or distributed systems.
- Scan for sensitive or confidential information.
- Attempt to access various sensitive Medicare data sets.
- Perform automated vulnerability scanning comparable to ISS Internet Scanner policy level L1-L2.
- Attempt to gain password files and passwords hashes using network-sniffing applications.
- Use password cracking applications to discover valid passwords from encrypted password files or through brute force log-on attempts.
- Attempt to create false trust relationships and access network user lists using vendor security tools.
- Perform penetration testing of internal networks, systems, and applications that store, process, or transmit Medicare information.
- Complete internal network security testing involving manual procedures, such as port scanning, vulnerability scanning, password discovery, and malicious code introduction.
- Perform tests to determine CMS' internal intrusion detection capability. These tests may include port scans, vulnerability scans, password cracking attempts, and manual vulnerability exploitation.
- Conduct war dialing to identify active modem connections, and attempt to gain access to active connections.
- Attempt to execute / discover known vulnerabilities associated with default operating system, web server, and database server configurations.

Vulnerability discovery shall be conducted in a progressive manner, as laid out in the test plan. Testing shall begin with little or no access to the infrastructure components, and then progress to subsequent phases that involve testing from the perspective of an authorized user(s) and review of firewall configurations, router configurations, and operating system or application platform configurations.

Procedural vulnerabilities may be discovered through physical assessments (facility walk-through) and personnel interviews. Physical security vulnerabilities shall be discovered through a physical security assessment, whereby evaluators attempt to access protected grounds, buildings, facilities, or rooms. The physical security assessment shall be conducted only to the extent authorized by CMS, and may be limited only to validate that appropriate locks, security guards, and other controls are in place to protect restricted physical areas.

Vulnerabilities in disaster recovery, contingency planning, and personnel roles and responsibilities shall be discovered through personnel interviews. Standard interview questionnaires shall be developed to assist the testing entity in determining whether vulnerabilities exist, particularly in the following areas:

- Disaster planning;
- Disaster preparedness (availability of back-up facility, back-up data, personnel);
- Personnel security;
- Personnel separation of duties;
- User access change management;
- Vendors / technical support access; and
- Controls for users of varying levels of access (authorizing access, assigning access rights and privileges).

During the vulnerability discovery process, all output, test results, communications, and working papers shall be retained. The tester shall capture, document, and retain information sufficient to prove the existence of vulnerabilities discovered through the testing process. This information shall be submitted to CMS with the findings report, and includes, but is not limited to: screenshots, port scan results, vulnerability scan reports, and e-mail communications.

3.3.4 VULNERABLITY VALIDATION

Automated vulnerability scanning tools commonly report “false positive” results, because the scanning tools rely upon system and application version numbers, rather than actual vulnerability exploitation, to determine whether a vulnerability exists. To prevent reporting “false positive” results in the findings report, a potential vulnerability shall not be reported unless it can be verified. Manual validation of vulnerabilities is necessary in the case of many web-associated vulnerabilities, text window access vulnerabilities, and similar vulnerabilities that require user intervention to exploit. Vulnerability validation shall be conducted to determine, to the best of the tester’s knowledge, whether a reported vulnerability is valid (it does exist and could be exploited) or is invalid (the report vulnerability could not be exploited or clearly does not exist). Validation test procedures should only demonstrate that the vulnerability does or does not exist;

the vulnerability should not be exploited, and no test procedures should be conducted that have the potential to disrupt system or business operations, unless expressly authorized by CMS.

During the vulnerability validation process, all output, test results, communications, and working papers shall be retained. The tester shall capture, document, and retain information sufficient to prove the existence or non-existence of vulnerabilities discovered through the testing process. This information shall be submitted to CMS with the findings report, and includes, but is not limited to: screenshots, port scan results, vulnerability scan reports, and e-mail communications.

3.4 FINDINGS REPORT

The findings report to be documented at the end of the testing engagement shall include all vulnerabilities that were discovered during the infrastructure test process. The report shall be developed in accordance with the *CMS Reporting Standard for Information Security Testing*. In the findings report, each vulnerability discovered is documented as a Business Risk. The Business Risk shall include a technical discussion of how the vulnerability was discovered and the potential impact of the vulnerability to CMS' business. Each Business Risk shall also contain suggested corrective actions for closing or reducing the impact of each vulnerability. The processes and requirements for developing, assembling, and submitting the findings report and supporting documentation are contained within the *CMS Reporting Standard for Information Security Testing*. Following completion of the application security test, the results of the infrastructure and application tests shall be combined in a single report and submitted to CMS.

4 APPLICATION SECURITY TEST

Application security testing shall be conducted by an independent contractor or CMS personnel (testing entity) based upon the guidance in this section. The application test scope document, security test plan and test scripts shall identify and describe the testing techniques and test procedures necessary to validate the actual effectiveness of security controls implemented on the system, and to discover procedural and technical vulnerabilities and weaknesses. To conduct the application security test, the testing entity shall execute the test plan and test scripts.

The identification and prioritization of application security testing will be made by CMS, SSG. Various testing techniques can be employed when performing application security testing, including Manual Inspections and Reviews, Scanning (Network and Vulnerability), Password Cracking, Log Reviews, Penetration Testing and Application Source Code Reviews.

Manual Inspections and Reviews
Manual inspections are human-driven reviews that typically test the security implications of people, polices, procedures and processes, but also include inspections of technology decisions supporting application security control requirements and processing platform technical security control parameter(s) settings. Manual inspections and reviews are one of the few ways to test the software development lifecycle process and to ensure that there is an adequate policy or skill set in place to ensure the integrity of the security controls are maintained through the application(s) lifecycle. Manual reviews are particularly useful for testing whether people understand the security process, have been made aware of policy, and have the appropriate skills to design and / or implement a secure application. Other activities accomplished using manual inspections and reviews are: documentation reviews; secure coding policies; security requirements; and architecture designs.

Scanning
Scanning is completed using software tools to identify hosts, open ports and provide information on the conditions found. The two types of scanning used in application security testing are network and vulnerability, which are described in Appendices A and B, respectively.

Password Cracking
Password cracking programs are used to identify weak passwords as described in Appendix-C.

Log Reviews
Log review and analysis provide a dynamic picture of on-going system activities that are described in Appendix-D.

Penetration Testing
Penetration testing, as described in Appendix-E, is security testing in which evaluators attempt to circumvent the security features of an application system.

Application Source Code Reviews
Application source code review, as described in Appendix-G, is the process of manually checking the application's source code for security-related issues and weaknesses. Unlike

testing third party closed software such as operating systems, when testing web applications (especially if they have been developed in-house) the source code is, and should be, almost always available. Source code analysis can be an extremely effective process to find implementation issues such as places where input validation was not performed or when fail open control procedures may be present. However, it is important to note that a comprehensive code review can be an onerous effort, considering budget and resources. Therefore, it may be cost effective to conduct a code review of critical modules to mission-critical systems or shared modules for multiple high-priority systems, only.

Application security code reviews are best handled in a holistic and comprehensive manner to meet the objectives of the review: 1) producing a complete picture of the application's trust model while considering as many avenues of attack as possible, and 2) providing detailed recommendations for improvements. For that reason a five-phase methodology shall be utilized in application security code reviews.

1. Identification of base technologies;
2. Identification of application components;
3. Identification of known vulnerabilities;
4. Review of code; and
5. Test/verify possible exploits.

In each phase of the review, the testing entity should build upon the data collected and derived in the previous phases to refine and clarify the trust model, paths of attack, vulnerabilities and weaknesses, and finally recommendations for improvement(s). Through this review process, the testing entity shall inspect the overall application implementation to ensure that proper controls have been selected and implemented where necessary to ensure the confidentiality, availability and integrity of all aspects of the application processes, and the data.

Identify Base Technologies

This phase consists of two activities. First, catalogue all the application technologies (e.g., Visual Basic, Java, SQL Server, WebSphere, etc.) that support the application processing followed by research of each of the technologies to determine potential current weaknesses that an application may inherit simply by incorporating the technology.

Identify Application Components

Next, the testing entity shall divide the application into its basic components. These include those components intended for workstations, servers, operating systems, network infrastructure, users, administrators, and the application code itself.

Identify Known Vulnerabilities

From this information the testing entity shall seek out known vulnerabilities affecting all aspects of the application implementation. These include all published or generally known defects (bugs) and exploitable holes in the operating system, web server, application server, and other third-party components. Most of these vulnerabilities have existing patches, but hackers often exploit systems where patches have not been applied in a timely fashion.

Review of Code

This is the longest and most costly (budget, time and resources) phase of the application security code review. In this phase, the testing entity performs a line-by-line examination of the application code with the objective of detecting security defects and common implementation errors within the application code base. Defects can be logic errors, anomalies in the code that might indicate an erroneous condition, and / or non-compliance with project or industry standards. The testing entity will review security control implementations within the application code base to ensure consistency and adherence to project and industry standards.

As part of this review all application test scripts (e.g., unit, integration and user tests) and procedures shall be reviewed to ensure that security controls are thorough and effective. This review process will ensure that the application has been adequately tested.

The testing entity shall pay special attention to the six typical classes of security vulnerabilities (see table below) that place the confidentiality, availability, and integrity of application processing at risk.

Vulnerability Class	**Description and Examples**
Data Faults	Incorrect handling of data. • Sensitive data handling • Data encryption
Control Faults	Errors in application logic flow such as improperly applying the results of an authorization check. Software defects (such as race conditions) that allow unauthorized or inadvertent access to system resources or data and result in unauthorized disclosure or modification of private information. • Authentication & access control • Session management
Input/Output Faults	Errors in managing output and input of data. Software defects leading to, or allowing, buffer overflow conditions (commonly known in the security industry to create vulnerabilities in systems) fall into this class of vulnerability. • Input validation • Information disclosure
Interface Faults	Errors in the interface between two components in the application system (e.g., between two modules of application code or between an application code module and an operating system or a database system service). • Parameter manipulation • Administrative interfaces

Storage Management Faults	Errors in managing, allocation, or freeing memory or disk space.
Exception Management Faults	Errors in responding to exceptions in the computing environment (e.g., running out of disk space or memory). Software defects that cause a program to abort resulting in a Denial-of-Service to end-users are in this vulnerability class.

Test / Verify Possible Exploits
The final phase is to probe the application using test examples to verify the possible security flaws. This testing reveals potential vulnerabilities existing in the code that are most feasible to exploit. This testing also uncovers and verifies weaknesses that otherwise might remain unnoticed until some time in the future when they might create a security vulnerability or application error.

4.1 IDENTIFY RELEVANT TESTS AND TOOLS

The nature of the application (web-based, database, e-mail, etc.), the type of user interface, and the technical environment will drive the types of tests to be conducted, and the tools that will be employed to complete the tests. The testing entity shall identify the application type and user interface(s). This information was gathered during the Business Process Review and further information may be documented within the *Business and IS Risks Gap Analysis Report* and the *Security Controls Gap Analysis Report*. Based upon the application type and user interface, the testing entity shall identify relevant tests to be conducted. The tests must reflect the relative priority of the security control categories (as determined during the Business Process Review) and testing shall focus initially on the categories of greatest priority. Relevant tests shall include procedures to verify and validate the effectiveness of management, operational, and technical security controls documented within the SSP / IS RA, and methods to discover and identify procedural and technical vulnerabilities and threats not documented within the SSP / IS RA.

After identifying relevant tests that shall be conducted, the testing entity shall identify the tools that will be employed to complete each test. Tools may include technical software, such as port and vulnerability scanners, code scanners / analyzers, as well as interview questionnaires and other non-technical instruments that may be employed to gather information, identify vulnerabilities, and assess information security. A description of various testing tools can be found in the Appendices to this document.

4.2 TEST PROCEDURE DOCUMENTATION

4.2.1 TEST PLAN

The test plan documents the processes and procedures that are to be executed during the application security test. The test plan shall include the relevant test procedures identified during the previous step, and assign / apply relevant tools, methods, and personnel to achieve the test objective. The test plan shall define a progressive methodology for conducting the test. The first

phase of testing is conducted with the least information and access, and subsequent phases of testing involve greater knowledge of the technical application and increased access. For example, the first phase of testing may involve remote testing from the perspective of an unauthorized person. The next phase may then involve on-site testing from the perspective of an unauthorized person. The following phase would then involve on-site testing from the perspective of an authorized internal user, and the final phase would involve review of information or access provided to the testing entity by CMS. This progressive methodology is also referred to as an "outside-in" strategy.

The test plan shall include a requirement whereby the testing entity is to validate the effectiveness of security controls documented within the SSP / IS RA. Specific procedures for conducting these validation checks will be documented within the test scripts. The test plan shall also include the processes to be employed to discover and identify, and to validate, procedural and technical vulnerabilities in the application. These processes may include, but are not limited to; port scanning, vulnerability scanning, password cracking / discovery, manual penetration testing / access attempts, and technical review, analysis, and evaluation based upon interviews and documentation review. When a second or deeper level of application security testing is required, an actual Application Source Code Review shall be conducted see Appendix-G Application Source Code Review. The test plan shall also include requirements to re-test open vulnerabilities, to validate completed corrective actions, and to validate the application component inventory.

The test plan shall define a set of role-related tests that will be conducted. The purpose of the role-related testing is to validate that proper access permission and restrictions are assigned for each of the application user roles. Role-related tests shall be assigned to each of the relevant application user roles. For example, the testing entity, in developing the test plan may define seven (7) role-related test procedures (i.e., attempt to add user, attempt to change password, attempt to access a certain database, etc.). The role-related tests shall then be assigned to each of the application user roles, based upon the access permissions and expectations associated with each user role. The types of role-related tests to be conducted and the identification of application user roles shall be based upon the application type, user interface, and results / findings from the Business Process Review. The following table is an example of the role-related test mapping:

Roles	Role-Related Tests						
	1	2	3	4	5	6	7
Administrator	-	-	X	X	-	X	X
Help Desk	X	-	X	X	X	X	X
User Level II	X	X	X	X	X	X	X
User Level I	X	X	X	X	X	X	X
Unauthorized User ID	X	X	-	-	-	X	X
No User Account	X	X	-	-	-	X	X

The test plan shall be developed by the testing entity, and reviewed and authorized by CMS, before the start of any testing. The role-related test mapping shall be appended to the application

test plan. CMS will have an opportunity to comment on the test plan, and request that test procedures be added to or deleted from the test plan.

4.2.2 TEST SCRIPTS

The test scripts support the test plan by providing detailed criteria to be used in validating the implementation and effectiveness of documented security controls. The test scripts shall be designed to validate that the security controls documented within the SSP / IS RA are implemented, configured, and operate as expected. If, however, the *Security Controls Gap Analysis Report* reveals gaps between the documented security controls (SSP / IS RA) and the CMS ARS and CSR requirements (or any other test measure established by CMS), the test scripts should include the relevant ARS and / or CSR requirement. Test scripts are, in actuality, a pass / fail assessment system, whereby a certain expectation is defined, based upon the documented security control requirements, and during the application security test, through testing or other means, the expectation is proven to be true or false. Completion of the test scripts will involve personnel interviews and documentation review, but may also involve technical testing and review of application interfaces, code, and scripting to validate the proper implementation of controls. The output and results of the personnel interviews and documentation review will drive the need for manual verification of security controls through testing and compliance reviews.

4.3 TEST PLAN AND TEST SCRIPT EXECUTION

4.3.1 COMPONENTS INVENTORY VALIDATION

The components inventory, as documented in the SSP / IS RA, shall be validated against the actual application environment. The validation process will include personnel interviews, automated scans, and manual test procedures. Any discrepancies between the documented component inventory and the actual environment shall be documented as a “finding”.

There are several different types of security testing which may be employed. The following section describes each testing technique, and provides additional information on the strengths and /or weaknesses of each. Some testing techniques are predominantly manual, requiring an individual to initiate and conduct the test. Other tests are automated and require less human involvement.

The following types of testing are described in their respective appendix:

- Network Scanning.....Appendix-A
- Vulnerability Scanning .....Appendix-B
- Password Cracking.....Appendix-C
- Log Review.....Appendix-D
- Penetration Testing .....Appendix-E
- Application Source Code Review…….Appendix-G

After running each test, the results of the test shall be documented, the system owner shall be informed of the results, and a plan to mitigate the vulnerabilities shall be developed.

Only designated individuals, including network administrators or individuals contracted to perform the network scanning as part of a larger series of tests, shall conduct the tests described in this section. Depending on the extent of the testing, the approval for the tests may need to come from an individual as prominent as the Chief Information Officer (CIO). Since a number of these tests mimic some of the signs of attack, the appropriate managers shall be notified to avoid confusion and unnecessary expense.

4.3.2 MANUAL VALIDATION OF DOCUMENTED CONTROLS

The test scripts will include detailed criteria against which to measure actual implementation and effectiveness of documented security controls. Based upon the prioritization of security control categories, determined during the Business Process Review, test procedures shall be conducted to validate that the documented security controls operate as expected and required. The prioritization of controls will determine which control categories shall be validated during the current test. Subject to the prioritization of control categories, management, operational and technical security controls implemented for the application shall be validated. Management and operational controls evaluated during the infrastructure test shall not be re-tested during the application security test. Only those security controls directly relating to the application shall be evaluated during the application test.

Validation of management and operational controls requires the review of documentation, as well as personnel interviews. The testing entity validating the application-specific management controls shall review risk assessment, security planning, system / services acquisition, and C&A documentation to validate that the required documentation exists and that it is reasonably comprehensive and accurate. Personnel interviews shall supplement the documentation review, and will enable the evaluator to determine how the documented processes and requirements are followed and completed within the CMS environment. Typically, Operational controls, for the most part, shall fall within the scope of infrastructure security testing. Validation of application-specific operational controls, such as configuration management, information integrity, and maintenance controls, shall be validated during the application security test, through the review of documentation and through personnel interviews.

Technical security controls are validated primarily through the execution of automated and manual test procedures, but also through personnel interviews. To validate technical controls, the testing entity shall examine the user interface, evaluate application specific controls (i.e., authentication mechanisms, session control, access restrictions, communications protection, audit logs), and, if necessary, review application code / scripts.

During the manual validation of documented controls, the test script papers shall be completed. For each test or security check conducted, a pass / fail assessment shall be recorded on the test script document. The test scripts shall be signed by both the tester assigned to conduct the testing and by the System Owner, or other person, responsible for the security of the application. The completed and signed test scripts serve as a written record of the test process, and acceptance of the test results.

4.3.3 VULNERABILITY DISCOVERY

After validating the security controls documented within the SSP / IS RA, the testing entity shall conduct vulnerability testing to discover and identify procedural and technical vulnerabilities in the application. A description of vulnerabilities can be found in Appendix-F, Common Application Vulnerabilities of this document. Technical vulnerabilities may be discovered through a broad range of tools and testing procedures. A partial list of application testing tools is provided in Appendix H, Application Testing Tools.

The following common test procedures and techniques shall be conducted, when appropriate:

- Conduct user-role testing, based upon the role-related test mapping attached to the application test plan;
- Evaluate application-specific authentication mechanisms. The tester shall attempt to gain access without a valid user account, and attempt to log-on with default and easily-guessed passwords;
- Evaluate application interfaces / user interface controls;
- Evaluate application input / output security controls;
- Evaluate application privileges / user role configuration. The tester shall attempt to access information resources outside the scope of the authorized user role. The tester shall validate that “Read” / “Write” access is limited to only authorized resources;
- Perform attempts to access application resources in the context of another user.
- Perform attempts to elevate access to a broader role;
- Review application-specific audit log configuration settings;
- Review application logs produced during testing to validate that application logging operates as required;
- Review application administration / management connectivity. Attempt to connect to management ports, services, and interfaces;
- Evaluate application session control management. Attempt to take-over sessions created by other users. Attempt to restore old sessions without re-authenticating.
- Attempt cookie poisoning;
- Test for buffer overflow conditions / validate that forms limit user input;
- Evaluate application error handling. Attempt to gather configuration information and other sensitive information from error messages;
- Attempt to access hidden URLs;
- Attempt to access and manipulate web scripts (CGI, Active Server Pages, Java Server Pages, Cold Fusion, Perl, etc.);
- Attempt to inject commands into web requests and submissions;
- Attempt cross-site scripting;
- Attempt SQL command injection;
- Attempt to manipulate HTML form submissions and hidden fields;
- Inspect code / scripts for vulnerabilities, coding weaknesses, and potential buffer overflow conditions;
- Inspect code / scripts for hard-coded passwords;
- Inspect code / scripts for the existence of back doors; and

- Inspect code / scripts for sensitive information (hidden URLs, IP addresses, server names, SQL commands, etc.).

Vulnerability discovery shall be conducted in a progressive manner, as laid out in the test plan (see Appendix-B, Vulnerability Scanning). Testing shall begin with little or no access to the application, and then progress to subsequent phases that involve testing from the perspective of an authorized user(s) and review of application interfaces, application configuration, and scripting / code.

Procedural vulnerabilities may be discovered through personnel interviews. Vulnerabilities in personnel roles and responsibilities shall be discovered through personnel interviews. Standard interview questionnaires shall be developed to assist the testing entity in determining whether vulnerabilities exist, particularly in the following areas:

- Personnel separation of duties;
- User access change management;
- Change control processes;
- Application administration / management processes;
- Vendor / technical support access; and
- Controls for users of varying levels of access (authorizing access, assigning access right and privileges).

During the vulnerability discovery process, all output, test results, communications, and working papers shall be retained. The testing entity shall capture, document, and retain information sufficient to prove the existence of vulnerabilities discovered through the testing process. This information shall be submitted to CMS with the findings report, and includes, but is not limited to: screenshots, automated scan results, and e-mail communications.

4.3.4 VULNERABILITY VALIDATION

Automated vulnerability scanning tools (see “Tools” at end of Appendix-A, Network Scanning) commonly report “false positive” results, because the scanning tools rely upon system and application version numbers, rather than actual vulnerability exploitation, to determine whether a vulnerability exists. To prevent reporting false-positive results in the findings report, a potential vulnerability shall not be reported unless it can be verified. Manual validation of vulnerabilities is necessary in the case of many web-associated vulnerabilities, text window access vulnerabilities, and similar vulnerabilities that require user intervention to exploit. Vulnerability validation shall be conducted to determine, to the best of the tester’s knowledge, whether a reported vulnerability is valid (it does exist and could be exploited) or is invalid (the report vulnerability could not be exploited or clearly does not exist). Validation test procedures should only demonstrate that the vulnerability does or does not exist; the vulnerability should not be exploited, and no test procedures should be conducted that have the potential to disrupt system or business operations, unless expressly authorized by CMS.

During the vulnerability validation process, all output, test results, communications, and working papers shall be retained by CMS. The testing entity shall capture, document, and retain

information sufficient to prove the existence or non-existence of vulnerabilities discovered through the testing process. This information shall be submitted to CMS with the findings report, and includes, but is not limited to: screenshots, automated scan results, and e-mail communications.

4.3.5 REPORTING "CRITICAL IMMEDIATELY" VULNERABILITIES

If, during the test process, a critical vulnerability is discovered and confirmed to exist, this finding shall be reported to CMS SSG immediately. A critical vulnerability includes any weakness, flaw, bug, configuration error, or other defect, which, if exploited, is likely to cause significant political, financial, and / or legal damage to CMS. Immediate notification of critical vulnerabilities is required where a vulnerability meets the above criteria, the threat exposure is considered high, and security controls are not effectively implemented to reduce the severity of impact if the vulnerability were to be exploited. If there is any question as to whether a vulnerability test is critical, and requires immediate notification, the testing entity shall err on the side of caution and report the vulnerability to CMS SSG immediately.

4.4 FINDINGS REPORT

The findings report to be documented at the end of the testing engagement shall include all vulnerabilities that were discovered during the application security test, and, if conducted, the infrastructure security test. The report shall be developed in accordance with the *CMS Reporting Standard for Information Security Testing*. In the findings report, each vulnerability discovered is documented as a Business Risk. The Business Risk shall include a technical discussion of how the vulnerability was discovered and the potential impact of the vulnerability to the CMS business. Each Business Risk shall also contain suggested corrective actions for closing or reducing the impact of each vulnerability. The processes and requirements for developing, assembling, and submitting the findings report and supporting documentation are contained within the *CMS Reporting Standard for Information Security Testing*. Following completion of the application security test, the results of the infrastructure (if conducted) and application tests shall be combined in a single report and submitted to CMS.

APPENDICES

APPENDIX-A NETWORK SCANNING

Network scanning involves the use of a port scanner to identify all hosts potentially connected to a CMS network, the network services operating on those hosts, such as the file transfer protocol (FTP) and hypertext transfer protocol (HTTP), and the specific application running the identified service, such as WU-FTPD, Internet Information Server (IIS) and Apache for the HTTP service. The result of the scan is a comprehensive list of all active hosts and services, printers, switches, and routers operating in the address space scanned by the port-scanning tool, i.e., any device that has a network address or is accessible to any other device.

Port scanners, such as NMAP, first identify active hosts in the address range specified by the user using Transport Control Protocol / Internet Protocol (TCP / IP) Internet Control Message Protocol (ICMP) ECHO and ICMP ECHO_REPLY packets. Once active hosts have been identified, they are scanned for open TCP and User Datagram Protocol (UDP) ports that will then identify the network services operating on that host. A number of scanners support different scanning methods that have different strengths and weaknesses that are usually explained in the scanner documentation. For example, certain scans are better suited for scans through firewalls and others are better suited for scans that are internal to the firewall.

All basic scanners will identify active hosts and open ports, but some scanners provide additional information on the scanned hosts. The information gathered during this open port scan will often identify the target operating system. This process is called operating system fingerprinting. For example, if a host has TCP port 135 and 139 open, it is most likely a Windows NT or 2000 host. Other items such as the TCP packet sequence number generation and responses to ICMP packets, e.g., the TTL (Time To Live) field, also provide a clue to identifying the operating system. Operating system fingerprinting is not foolproof. Firewalls filter (block) certain ports and types of traffic, and system administrators can configure their systems to respond in non-standard ways to camouflage the true operating system.

In addition, some scanners will assist in identifying the application running on a particular port. For example, if a scanner identifies that TCP port 80 is open on a host, it often means that the host is running a web server. However, identifying which web server product is installed can be critical in identifying vulnerabilities. For example, the vulnerabilities for Microsoft's IIS server are very different from those associated with Apache web server. The application can be identified by "listening" on the remote port to capture the "banner" information transmitted by the remote host when a client (web browser in this example) connects. Banner information is generally not visible to the end-user (for web servers / browsers); however, when it is transmitted, it can provide a wealth of information, including the application type, application version and even operating system type and version. The process of capturing banner information is sometimes called "banner grabbing".

While port scanners identify active hosts, services, applications and operating systems, they do NOT identify vulnerabilities (beyond some common Trojan ports). Vulnerabilities can only be identified by a human who interprets the mapping and scanning results. From these results, a qualified individual can ascertain what services are vulnerable as well as the presence of Trojans.

Although the scanning process itself is highly automated, the interpretation of scanned data is not.

CMS shall conduct network scanning to:

- Check for unauthorized hosts connected to the CMS network;
- Identify vulnerable services;
- Identify deviations from the allowed services defined in the CMS security policy;
- Prepare for penetration testing; and
- Assist in the configuration of the intrusion detection system (IDS) to help mitigate vulnerabilities.

The scanning can also disrupt network operations by consuming bandwidth and slowing network response times. However, network scanning does enable CMS to maintain control of its IP address space and ensure that its hosts are configured to run only approved network services. To minimize disruptions to operations, scanning software should be carefully selected. Network scanning can also be conducted after hours to ensure minimal impact to operations, with the caveat that some systems may not be turned on.

Network scanning results shall be documented and identified deficiencies corrected. The following corrective actions may be necessary as a result of network scanning:

- Investigate and disconnect unauthorized hosts;
- Disable or remove unnecessary and vulnerable services;
- Modify vulnerable hosts to restrict access to vulnerable services to a limited number of required hosts (e.g., host level firewall or TCP wrappers); and
- Modify enterprise firewalls to restrict outside access to known vulnerable services.

Network Sniffer Tools

The tools listed in the tables below are identified for information purposes only; the inclusion of a tool does not constitute an endorsement of that tool. No technical assessment was conducted for any of these tools, beyond investigating product literature (or, in the case of public domain tools, the high-level technical description).

The following are examples of tools that are available to assist in reviews and may not be applicable to all applications to be tested.

Tool	Capabilities	Website	Linux/ Unix	Win32	Cost
Dsniff	Unix sniffer	http://www.monkey.org/~dugsong/dsniff/	✓		Free
Description	*Dsniff is a collection of tools for network auditing and penetration testing. Dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). Arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). Sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKIs.*				
Ethereal	Unix/Windows sniffer with GUI	http://www.ethereal.com/	✓	✓	Free
Description	*Ethereal is a free network protocol analyzer for Unix and Windows. It allows users to examine data from a live network or from a capture file on disk. It can interactively browse the capture data, viewing summary and detail information for each packet. Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session and parse an 802.11 packet.*				
Sniffit	Unix sniffer	http://reptile.rug.ac.be/~coder/sniffit/sniffit.html http://www.symbolic.it/Prodotti/sniffit.html (Windows)	✓	✓	Free
Description	*A freeware general-purpose sniffer for various versions of Linux, Unix, and Windows.*				
Snort	Unix sniffer/IDS	http://www.snort.org	✓	✓	Free
Description	*A freeware lightweight IDS and general-purpose sniffer for various versions of Linux, Unix and Windows.*				
TCPDump	Unix sniffer	http://www-nrg.ee.lbl.gov/	✓		Free
Description	*A freeware general-purpose sniffer for various versions of Linux and Unix.*				
WinDump	Windows sniffer	http://netgroup-serv.polito.it/windump/		✓	Free
Description	*A freeware Windows general-purpose sniffer based on TCPDump.*				

Scanning and Enumeration Tools

Tool	Capabilities	Website	Linux/ Unix	Win32	Cost
DUMPSec	Windows enumeration tool	http://www.systemtools.com		✓	Free
Description	*DumpSec is a security auditing program for Microsoft Windows. It dumps the permissions (DACLs) and audit settings (SACLs) for the file system, registry, printers and shares in a concise, readable listbox format, so that holes in system security are readily apparent. DumpSec also dumps user, group, and replication information.*				
Firewalk	Firewall filter rule mapper	http://www.packetfactory.net/firewalk/	✓		Free
Description	*Firewalking is a technique that employs traceroute-like techniques to analyze IP packet responses to determine gateway ACL filters and map networks. Firewalk the tool employs the technique to determine the filter rules in place on a packet forwarding device.*				
Fscan	Port scanner	http://www.foundstone.com/		✓	Free
Description	*FScan is a command-line port scanner. It will scan for both TCP and UDP ports.*				
LANguard Network Scanner	Port scanner, OS detection	http://www.gfi.com/languard/lanscan.htm		✓	Free
Description	*LANguard Network Scanner is a freeware security and port scanner to audit your network security. It scans entire networks and provides NetBIOS information for each computer such as hostname, shares, logged on user name. It does OS detection, password strength testing, detects registry issues and more. Reports are outputted in HTML.*				
NDS Snoop	Novell Enumeration Tool	http://www.novell.com/coolsolutions/		✓	Free
Description	*Provides the ability to enumerate a variety of NDS objects and values.*				
Nmap	Port scanner, OS detection	http://www.insecure.org/nmap/	✓	✓	Free
Description	*Nmap ("Network Mapper") is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it also works against single hosts. Nmap uses raw IP packets to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and version) they are running, what type of packet filters/firewalls are in use, and other characteristics.*				
Solarwinds	Network enumeration	http://www.solarwinds.net/		✓	$
Description	*A collection of network and management and discovery tools.*				
SuperScan	Port scanner, OS detection, Banner enumeration	http://www.foundstone.com/		✓	Free

APPENDIX-B VULNERABILITY SCANNING

Vulnerability scanners take the concept of a port scanner to a higher level. Like a port scanner, a vulnerability scanner identifies hosts and open ports, but it also provides information on the associated vulnerabilities (as opposed to relying on human interpretation of the results). Most vulnerability scanners also attempt to provide information on mitigating discovered vulnerabilities.

Vulnerability scanners provide system and network administrators with pro-active tools that can be used to identify vulnerabilities before an adversary can find them. A vulnerability scanner is a relatively quick and easy tool with which to quantify CMS' exposure to surface vulnerabilities.

A surface vulnerability is a weakness, as it exists in isolation, independent from other vulnerabilities. The difficultly in identifying the risk level of vulnerabilities is that they rarely exist in isolation. For example, there could be several "low risk" vulnerabilities that exist on a particular network that, when combined, present a high risk.

Vulnerability scanners attempt to identify vulnerabilities in the hosts scanned. Vulnerability scanners can also help identify out-of-date software versions, applicable patches or system upgrades, and validate compliance with, or deviations from, the CMS security policy. To accomplish this, vulnerability scanners identify operating systems and major software applications running on hosts and match them with known exposures. Scanners employ large databases of vulnerabilities to identify flaws associated with commonly used operating systems and applications.

The scanner will often provide significant information and guidance on mitigating discovered vulnerabilities. In addition, vulnerability scanners can automatically make corrections and fix certain discovered vulnerabilities. This assumes that the operator of the vulnerability scanners has "root" or administrator access to the vulnerable host.

However, vulnerability scanners have some significant weaknesses. 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). This means an individual with expertise in networking and operating system security and in administration must interpret the results.

Since vulnerability scanners require more information than port scanners to identify reliably the vulnerabilities on a host, vulnerability scanners tend to generate significantly more network traffic than port scanners. This may have a negative impact on the hosts or network being scanned or network segments through which scanning traffic is traversing. Many vulnerability scanners also include tests for Denial-of-Service (DoS) attacks that, in the hands of an inexperienced tester, can have a considerable negative impact on scanned hosts.

Vulnerability scanners are better at detecting well-known vulnerabilities than the more esoteric ones, primarily because it is difficult to incorporate all known vulnerabilities in a timely manner.

Also, manufacturers of these products keep the speed of their scanners high (more vulnerabilities detected requires more tests which slows the overall scanning process).

Vulnerability scanners provide the following capabilities:

- Identifying active hosts on network;
- Identifying active and vulnerable services (ports) on hosts;
- Identifying applications and “banner grabbing”;
- Identifying operating systems;
- Identifying vulnerabilities associated with discovered operating systems and applications;
- Identifying mis-configured settings;
- Testing compliance with host application usage / security policies; and
- Establishing a foundation for penetration testing.

Vulnerability scanners can be of two types: network-based scanners and host-based scanners. Network-based scanners are used primarily for mapping an organization's network and identifying open ports and related vulnerabilities. In most cases, these scanners are not limited by the operating system of targeted systems. The scanners can be installed on a single system on the network and can quickly locate and test numerous hosts.

Host-based scanners require installation on each host to be tested and are used primarily to identify specific host operating system and application security control configuration discrepancies and vulnerabilities. Because host-based scanners are able to detect vulnerabilities at a higher degree of detail than network-based scanners, they usually require not only host (local) access but also a “root” or administrative account.

Vulnerability scanning results shall be documented and discovered deficiencies corrected. The following corrective actions may be necessary as a result of vulnerability scanning:

- Upgrade or patch vulnerable systems to mitigate identified vulnerabilities as appropriate;
- Deploy mitigating measures (technical or procedural) if the system cannot be patched immediately (e.g., operating system upgrade will make the application running on top of the operating system inoperable), in order to minimize the probability of this system being compromised;
- Improve configuration management program and procedures to ensure that systems are upgraded routinely;
- Assign a staff member to monitor vulnerability alerts and mailing lists, examine their applicability to the CMS environment and initiate appropriate system changes; and
- Modify CMS security policies, architecture, or other documentation to ensure that security practices include timely system updates and upgrades.

Vulnerability Assessment Tools

The tools listed in the tables below are identified for information purposes only; the inclusion of a tool does not constitute an endorsement of that tool. No technical assessment was completed for any of these tools, beyond investigating the product literature (or, in the case of public domain tools, the high-level technical description).

The following are examples of tools that are available to assist in reviews and may not be applicable to all applications to be tested.

Tool	Capabilities	Website	Linux/ Unix	Win32	Cost
CyberCop Scanner	Vulnerability scanner	http://www.pgp.com/products/	✓	✓	$
Description	*CyberCop Scanner is a network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
ISS Internet Scanner	Vulnerability scanner	http://www.iss.net/		✓	$
Description	*ISS Internet Scanner is a network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
Nessus	Vulnerability scanner	http://www.nessus.org/	✓	✓ (client only)	Free
Description	*A freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
SecureScanNX	Vulnerability scanner	http://www.viqilante.com/securescan/		✓	$
Description	*SecureScan NX is a corporate network security assessment tool that proactively probes your organization's network and firewalls to assess vulnerabilities and suggest corrective action.*				
SAINT	Vulnerability scanner	http://www.wwdsi.com/saint/	✓		$
Description	*SAINT is an updated and enhanced version of SATAN, is designed to assess the security of computer networks.*				
SARA	Vulnerability scanner	http://www-arc.com/sara/	✓		Free
Description	*Sara is a freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
SATAN	Vulnerability scanner	http://www.fish.com/satan/	✓		Free
Description	*SATAN is a tool to help system administrators. It recognizes several common networking-related security problems, and reports the problems without actually exploiting them.*				

APPENDIX-C PASSWORD CRACKING

Password cracking programs shall be used to identify weak passwords. Password cracking verifies that users are employing sufficiently strong passwords. Passwords are generally stored and transmitted in an encrypted form called a ”hash”. When a user logs-on to a computer / system and enters a password, a hash is generated and compared to a stored hash. If the entered and the stored hashes match, the user is authenticated.

During a penetration test or a real attack, password cracking employs captured password hashes. Passwords hashes can be intercepted when they are transmitted across the network (using a network sniffer) or they can be retrieved from the targeted system. The latter generally requires administrative or “root” access on the target system.

Once the hashes are obtained, an automated password cracker rapidly generates hashes until a match is found. The fastest method for generating hashes is a dictionary attack that 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. As a result, any “dictionary” word no matter how obscure is weak.

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 attempt a number of variations. The attack tries common substitutes 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$%, etc.).

The most powerful password-cracking method is called the “brute force” method. Although brute force can take a period of time, it usually requires far less time than most password policies specify for password changing. Consequently, passwords found during brute force attacks are also too weak. Brute force randomly generates passwords and their associated hashes. However, since so many possibilities exist it can take months to crack a password. Theoretically all passwords are “crackable” from a brute force attack given enough time and processing power.

A strong Linux / Unix password is one that is long (greater than 10 characters at least) and complex (contains both upper and lower case letters, special characters and numbers). Creating a strong Windows password is somewhat more complicated. Versions of Windows prior to Windows 2000 use LanMan password hashes, which have several associated weaknesses. First, LanMan is not case sensitive, all alphabetic characters are converted to uppercase. This effectively reduces the number of different combinations a password cracker has to utilize. Second, all LanMan passwords are stored as two seven-character hashes. Passwords that are exactly fourteen-characters long will be split into two seven-character hashes. Passwords less than fourteen characters will be padded up to fourteen characters. The splitting of the hash into two segments causes LanMan passwords to be less resistant to password cracking.

Password Crackers

The tools listed in the tables below are identified for information purposes only; the inclusion of a tool does not constitute an endorsement of that tool. No technical assessment was done for these tools, beyond investigating the product literature (or, in the case of public domain tools, the high-level technical description).

The following are examples of tools that are available to assist in code reviews and may not be applicable to all applications to be tested.

Tool	Capabilities	Website	Linux/ Unix	Win32	Cost
Crack 5	Unix password cracker	http://www.crypticide.org/users/alecm/	✓		Free
Description	*Crack is a password guessing program that is designed to quickly locate insecurities in Unix (or other) password files by scanning the contents of a password file, looking for users who have misguidedly chosen a weak login password.*				
IMP 2.0	Novell Netware password cracker	http://www.wastelands.gen.nz		✓	Free
Description	*Imp is a NetWare password cracking utility with a GUI (Win95/NT). It loads account information directly from NDS or Bindery files and allows the user to attempt to compromise the account passwords with various attack methods.*				
John the Ripper	Windows and Unix password cracker	http://www.openwall.com/john/	✓	✓	Free
Description	*John the Ripper is a fast password cracker, currently available for many flavors of Unix, DOS, Win32, and BeOS. Its primary purpose is to detect weak Unix passwords, but a number of other hash types are supported as well.*				
L0pht Crack	Windows password cracker	http://www.securityfocus.com/tools/1005		✓	$
Description	*A password cracking utility for Windows NT, 2000 and XP.*				
Nwpcrack	Novell Netware password cracker	http://ftp.cerias.purdue.edu/pub/tools/novell/		✓	Free
Description	*A password cracking utility for Novell Netware.*				

APPENDIX-D LOG REVIEWS

Various system logs shall be used to identify deviations from the CMS security policy, including firewall logs, Intrusion Detection System (IDS) logs, server logs, and any other logs that are collecting audit data on systems and networks. Log review and analysis can provide a dynamic picture of on-going system activities that can be compared with the intent and content of the security policy. Audit logs shall be used to validate that the system is operating according to policies.

For example, if an IDS sensor is placed behind the firewall (within the enclave), its logs can be used to examine the service requests and communications that are allowed into the network by the firewall. If this sensor registers unauthorized activities beyond the firewall, it indicates that the firewall is no longer configured securely and a backdoor exists on the network.

"Snort" is a free IDS sensor with ample support. It is a network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. Snort can perform protocol analysis, content searching / matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI (Common Gateway Interface) attacks, SMB (System Message Block) probes, and OS fingerprinting attempts. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that uses a modular plug-in architecture.

Snort has a real-time alerting capability as well, incorporating alerting mechanisms for *syslog*, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's *smbclient.* Snort has three primary uses. It can be used as a straight packet sniffer like *tcpdump*, a packet logger (useful for network traffic debugging, etc), or as a complete network intrusion detection system.

The following actions shall be taken if a system is not configured according to policies:

- Remove vulnerable services if they are not needed;
- Reconfigure the system as required to reduce the chance of compromise;
- Change firewall policy to limit access to the vulnerable system or service; and
- Change firewall policy to limit accesses from the IP subnet that is the source of compromise.

APPENDIX-E PENETRATION TESTING

Penetration testing is security testing in which evaluators attempt to circumvent the security features of a system based on their understanding of the system design and implementation. The purpose of penetration testing is to identify methods of gaining access to a system by using common tools and techniques used by attackers.

Penetration testing can be overt or covert. These two types of penetration testing are commonly referred to as Blue Teaming and Red Teaming. Blue Teaming involves performing a penetration test with the knowledge and consent of the CMS IT staff. Red Teaming involves performing a penetration test without the knowledge of the CMS IT staff but with full knowledge and permission of CMS senior management.

A penetration test can be designed to simulate an inside and / or an outside attack. If both internal and external testing is to be performed, the external testing usually occurs first. With external penetration testing, firewalls usually limit the amount and types of traffic that are allowed into the internal network from external sources. Depending on what protocols are allowed through, initial attacks are generally focused on commonly used and allowed application protocols such as FTP, HTTP, or SMTP and POP.

To simulate an actual external attack, the testing entity is not provided with any real information about the target environment other than targeted IP address / ranges and they must covertly collect information before the attack. They collect information on the target from public web pages, newsgroups and similar sites. They then use port scanners and vulnerability scanners to identify target hosts. Since they are, most likely, going through a firewall, the amount of information is far less than they would achieve if operating internally. After identifying hosts on the network that can be reached from the outside, they attempt to compromise one of the hosts. If successful, they then leverage this access to compromise others hosts not generally accessible from outside. This is why penetration testing is an iterative process that leverages minimal access to gain greater access.

An internal penetration test is similar to an external except that the testers are now on the internal network (i.e., behind the firewall) and are granted some level of access to the network (generally as a user but sometimes at a higher level). The penetration testers will then try to gain a greater level of access to the network through privilege escalation. The testers are provided with the information about a network that someone with their provided privileges would typically be granted. This is generally as a standard employee although it can also be anything up to and including a system or network administrator depending on the goals of the test.

Penetration testing consists of four phases:

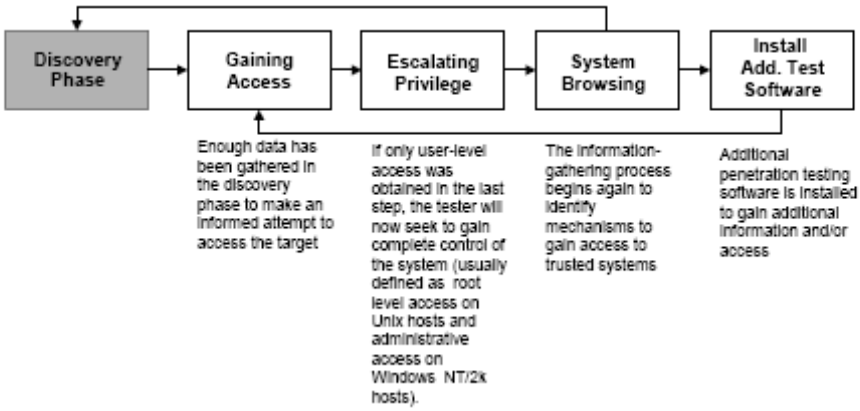
In the Planning Phase, rules are identified, management approval is finalized, and testing goals are set. The Planning Phase sets the groundwork for a successful penetration test. No actual testing occurs in the planning phase.

The Discovery Phase begins the actual testing. Network scanning (port scanning) is used to identify potential targets. In addition to port scanning, other techniques are commonly used to gather information on the targeted network:

- Domain Name System (DNS) interrogation;
- InterNIC (whois) queries;
- Search of the target CMS web server(s) for information;
- Search of the CMS Lightweight Directory Access Protocol server(s) (LDAP) for information;
- Packet capture (generally only during internal tests);
- NetBIOS enumeration (generally only during internal tests); and
- Network Information System ([NIS] generally only during internal tests).

The second part of the Discovery Phase is vulnerability analysis. During this phase, services, applications, and operating systems of scanned hosts are compared against vulnerability databases (for vulnerability scanners this process is automatic). Generally, human testers use their own database or public databases to identify vulnerabilities manually. This manual process is superior for identifying new or obscure vulnerabilities, but is much slower than an automated scanner.

Executing an Attack is at the heart of any penetration test. This is where previously identified potential vulnerabilities are verified 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 that are executed during attack execution 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. In either case, additional analysis and testing is required to determine the true level of risk for the network. This is represented in the feedback loop in figure above between the Attack and Discovery Phase of a penetration test.



While vulnerability scanners only check that a vulnerability may exist, the Attack Phase of a penetration test exploits the vulnerability, confirming its existence. Most vulnerabilities exploited by penetration testing and malicious attackers fall into the following categories:

- **Kernel Flaws**—Kernel code is the core of an operating system. The kernel code enforces the overall security model for the system. Any security flaw that occurs 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, which is usually a result of poor programming practice. When this occurs, arbitrary code can be introduced into the system and executed with the privileges of the running program. This code often can be run as root on UNIX systems and SYSTEM (administrator equivalent) on Windows systems.
- **Symbolic Links**—A symbolic link or *symlink* is a file that points to another file. Often there are programs that will change the permissions granted to a file. If these programs run with privileged permissions, a user could strategically create *symlink*s to trick these programs into modifying or listing critical system files.
- **File Descriptor Attacks**—File descriptors are non-negative integers that the system uses to keep track of files rather than using specific 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 when a program or process has entered into a privileged mode but before the program or process has given up its privileged mode. A user can time an attack to take advantage of this program or process while it is still in the privileged mode. If an attacker manages successfully to compromise the program or process during its privileged state, then the attacker has won the "race." Common race conditions include signal handling and core-file manipulation.
- **File and Directory Permissions**—File and Directory Permissions control the access users and processes have to files and directories. Appropriate permissions are critical to the security of any system. 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
- **Trojans**—Trojan programs can be custom-built or could include programs such as BackOrifice, NetBus, and SubSeven. Kernel root kits could also be employed once access is obtained to allow a backdoor into the system at any time.

- **Social Engineering**—Social engineering is the technique of using persuasion and / or deception to gain access to, or information about, information systems. It is typically implemented through human conversation or other interaction. The usual medium of choice is telephone but can also be e-mail or even face-to-face interaction. Social engineering generally follows two standard approaches. In the first approach the penetration tester poses as a user experiencing difficultly and calls the CMS Help Desk in order to gain information on the target network or host, obtain a log-on ID and credentials, or get a password reset. The second approach is to pose as the Help Desk and call a user in order to get the user to provide his / her UserID(s) and password(s). This technique can be extremely effective.

The Reporting Phase occurs simultaneously with the other three phases of the penetration test. In the Planning Phase, rules of engagement, test plans and written permission are developed. In the Discovery and Attack Phases, written logs are usually kept and periodic reports are made to system administrators and / or management, as appropriate. At the end of the test an overall testing report is developed to describe the identified vulnerabilities, provide a risk rating, and to give guidance on the mitigation of the discovered weaknesses.

Corrective measures can include closing discovered and exploited vulnerabilities, modifying CMS security policies, creating procedures to improve security practices, and conducting security awareness training for personnel to ensure that they understand the implications of poor system configurations and poor security practices.

APPENDIX-F COMMON APPLICATION VULNERABILITIES

There are nine (9) classes of common security flaws that place the confidentiality (including privacy), integrity and availability of an application at risk. These are:

1. Administrative interfaces;
2. Authentication & Access Control;
3. Configuration management;
4. Information gathering;
5. Input validation;
6. Parameter manipulation;
7. Sensitive data handling;
8. Session management; and
9. Cryptographic algorithms.

Within these classes, there are a series of common vulnerabilities that can be identified uniquely.

Inadequate Identification and Authentication
Occasionally, users are not required to enter a password before accessing an application, which can result in an easily circumvented authentication process. This category also includes authentication of users who should be denied access.

Insufficient Access Control
When restrictions on what authenticated users are prevented from doing are not properly enforced, both malicious and inadvertent access to other users' accounts, viewing of sensitive data, or using unauthorized functions may occur.

Improper Integration of Application Components
The application integration process could leave "backdoors" or "security holes" that make it possible for users to bypass access controls, second-level identification and authentication, or other security controls. Improper integration could also enable the ability to read security data passed between components, including incorrect interfaces between the application and cryptographic mechanisms on which the application may depend.

Weak Passwords
Passwords that are too short, not changed frequently enough, easy to guess, or which may be defaults provided by a vendor place the system at risk.

Plain Text Communication of Sensitive Information
Unencrypted, or plain text information may provide a way to circumvent or bypass application controls; e.g., clear text transmission of user passwords. This places both the integrity and confidentiality (including privacy) of the application at risk.

Incorrect Reparsing of Data
Movement of data, without adequate security, into application components where data processing occurs, such as user-provided identification data passed between application and backend server.

Susceptibility to Buffer Overflow
Application components in higher level languages, such as C, C++, etc. may not limit the amount of input properly, thereby allowing the data cache buffer for the application to overfill. When it overfills, the excess data may leak into the processing cache where they can result in a Denial-of-Service or possible exploitation of the application.

Lack of Adequate Parameter Validation
Parameter manipulation occurs when input data (such as query strings or cookies and form fields) are manipulated to cause an unintended action to occur. Parameters should always be validated to ensure the proper formatting and length each time the parameter is passed to the application. If not done, attackers can manipulate parameters in order to created unexpected or undesirable events within the application. This validation is an essential part of session control.

Input Validation of Active Content Data
Insufficient validation could cause active content-based applications to execute unexpected processes and make the application vulnerable to "cross-site scripting." In cross-site scripting attacks, the application can be used to transport an attack to an end user's browser or back-end systems, allowing the attacker to view session tokens, manipulate the remote workstation, or spoof or modify content in a way that the system does not expect or intend. This integrity check is part of Parameter Validation.

Acceptance of Meta Code Embedded Within Input Data
This vulnerability enables "stealth commanding"; i.e., the insertion of shell meta-characters in data input. An example is the character '!' which is used to access the command history in some shells; particularly troublesome in *tcsh*, where '!' can be used not just interactively, but in scripts. Another example is '|' (the "pipe") in Perl. Many Perl programs allow the user to input a filename, and then pass that filename to a program in a shell command. Because the shell may interpret characters differently than the Perl program, if the user includes '!' (the "bang") within the filename, the shell will attempt to execute the rest of the filename as a program. By including control string code (allowing the user to execute unintended actions) after the '!' character, hidden debug code or developer-instituted backdoors may result in security being compromised. This integrity check is also part of Parameter Validation.

Acceptance of Illegal Characters in Structured Query Language Queries (direct command injection)
Database applications that do not correctly validate and / or sanitize the user input can potentially be exploited in several ways. These include 1) changing SQL values; 2) concatenating SQL statements; 3) adding function calls and stored-procedures to a statement; and 4) typecasting and concatenating retrieved data. All applications should be stripped (or cleansed) of any characters

or strings that could possibly be used maliciously. Failure to do so places the confidentiality (including privacy), integrity and availability of the application at risk.

Use of Relative Pathnames
The use of relative pathnames enables users to gather information about the directory structure and content of application systems that can be used to launch other types of attacks. With this knowledge malicious users could remotely access confidential information or execute protected applications.

Remote Directory Listing
If no filename is specified at the end of the pathname, the system may simply list the full directory contents to the user, enable a malicious user to gather information about the application for use in an attack. When coupled with improper access controls, this information could enable the release of confidential data.

APPENDIX-G APPLICATION SOURCE CODE REVIEW

When a second level of application security testing is required, an actual Application Source Code Review is conducted. In this, the code of the application itself is examined for the component problems listed below:

- Denial-of-Service Attacks – overflowing the ability of the application to handle transactions.
- Buffer Overflow Assaults – sending large numbers of characters against the application.
- Session Hijacking – capturing a session for another purpose.
- Session Replay – taking unauthorized control of a previous authorized session.
- Hidden Manipulation – hidden field value changes.
- Stealth Operations – Placing of Trojan Horses.
- Parameter Tampering – altering URL parameters.
- Cross-Site Scripting – entering unauthorized script into authorized web pages.
- Debug Options – Trying Debug Syntax On URLs.
- Cookie Poisoning – altering cookie content.
- Reverse Directory Transversal – extending system access beyond application boundaries.
- Backup Checking – taking control of authorized sessions, or capturing sensitive information through browser "back to previous page" functions.
- Path Truncation – examining the potential for buffer overflow or script injection conditions.
- Hidden Web Paths – identification of paths not publicly advertised or linked.
- Application Mapping – identification of application data flow and backend support applications, including database servers.
- Directory Enumeration – discovery of all directories – including sample, administrative and executable directories.
- SQL Injection – sending unauthorized, unexpected, or malformed database commands.
- Caching – discovery of sensitive information contained in cached pages on server and client systems.

Application security code reviews typically include four classes of security vulnerabilities resulting from weaknesses in the application code. These include:

1. The ability to manipulate parameters to alter hidden input values in HTML code;
2. The possibility for buffer overflows due to invalidated application strings and the URL query string;
3. Potential for undertaking a Denial-of-Services attack due to server input errors; and
4. Improper session management that allows indefinite display of sensitive data.

The goal of an application security code review is to 1) produce a complete picture of the application's trust model while considering as many avenues of attack as possible; and 2) provide detailed recommendations for improvements

APPENDIX-H APPLICATION TESTING TOOLS

The tools listed in the tables below are identified for information purposes only; the inclusion of a tool does not constitute an endorsement of that tool. No technical assessment was completed for any of these tools, beyond investigating the product literature. The following are examples of tools available to assist in code reviews and may not be applicable to all applications to be tested.

Tool	Capabilities	Website	Linux/ Unix	Win 32	Cost
inSpect	Static Analysis	http://www.klocwork.com		√	
Description	Klockwork's inSpect checks for many implementation and security vulnerabilities within source code as well as cyclomatic complexity and coding technique violations. Scans using different options including: architecture, coding violations, metrics, security, and others. The inSpect engine analyzes the code and updates the database with information about the source code. The management console allows the management of reports for each project, tracking the progress of source changes. The tool generates reports in HTML, PDF, text and XML formats. Analyzes C, C++ and Java code.				
Prexis	Contextual analysis	http://www.ouncelabs.com	√	√	
	Through a complex suite of technologies that are built on top of a patent-pending contextual analysis engine, Prexis aims to find vulnerabilities within uncompiled applications or source code. Contextual analysis is defined by the act of determining if an implemented system call is truly vulnerable. Prexis does this through the inferred intelligence and understanding of the individual and interrelationships between the system calls, data elements, modules, processes and links. The Prexis 3.0 engine can quickly analyze C and C++ for Windows, and native Unix, Linux, Java and JSP environments.				
Holodeck Enterprise Edition	Fault Simulation	http://www.sisecure.com		√	
Description	Holodeck uses fault simulation to emulate real-world application and system errors. This allows testers and developers to work in a controlled, repeatable environment to analyze and debug error-handling code in hostile environments. Holodeck performs application monitoring, as well as error-logging and fault simulation, making it a tool for adept testers that need to understand the underpinnings of their applications for reliability testing. Applications that stand up to Holodeck attacks are by definition not fragile. Holodeck works with both Windows and .NET Applications. Holodeck supports reference parameters, private methods, and system reflection. Holodeck exposes UI on a .NET interface and generates code that allows a tester to easily instrument the application being tested. It advertises itself as fully supporting Windows services with the ability to launch or attach to any service regardless of whether it is a standalone service or part of a *svchost* process. It also allows Custom Test Project Creation. Using the Custom Test Project, Holodeck generates test code for .net and win32 APIs which allow testers to create logic for intercepted functions.				

Tool	Capabilities	Website	Linux/ Unix	Win 32	Cost
WebIinspect	Dynamic Analysis	http://www.spidynamics.com	√	√	
Description	WebInspect enables users to perform security assessments for any web application, including the industry leading application platforms: IBM WebSphere; Macromedia ColdFusion; Lotus Domino; Oracle Application Server; Macromedia JRun; BEA Weblogic; and Jakarta Tomcat. WebInspect uses "Adaptive-Agent" technology, a set of heuristics that enables intelligent application level security checks to be applied. WebInspect provides a full programming language and programming tools to write custom rules and extensive reporting capabilities.				
eEye Digitial Retina	Dynamic Analysis	http://www.eeye.com	√	√	
Description	Retina is able to scan UNIX-Based operating systems for vulnerabilities, including Solaris, Linux, and BSD, as well as networked devices (such as routers and firewalls). Retina includes vulnerability auditing modules for the following type of services: NetBIOS, HTTP, CGI, FTP, DNS, POP 3, SMTP, LDAP, TCP/IP, and UDP. Additionally, Retina has modules for checking registry settings, DoS vulnerabilities, Users and Accounts, password vulnerabilities and publishing extensions.				
WebXM	Online Risk Management	http://www.watchfire.com	√	√	
Description	WebXM automates the scanning, analysis and reporting of online security, privacy, quality, accessibility and compliance issues across corporate web properties. WebXM ensures visibility and control by delivering executive dashboards that are used to identify, assign and track the issues impacting online compliance.				
ISS Database Scanner	Database Scanner	http://www.iss.net	√	√	
Description	The ISS Database Scanner assessment tool identifies security vulnerabilities in leading database applications, including Microsoft SQL Server, Oracle and Sybase database servers. The Database Scanner offers security policy generation and reporting functionality, measures policy compliance, and automates the process for securing critical data. Database Scanner audit scans using an inside-out approach that enumerates users, groups, privileges, logins, and a wide number of other objects in the database, identifying incorrect privilege assignments and potential areas for unauthorized use by authorized users..				
Flawfinder	C and C++ Auditing	http://www.dwheeler.com/flawfinder	√	√	
Description	Flawfinder is a Python language program that can be used to assist auditing C and C++ program code. Flawfinder uses a built-in database of C and C++ functions with well-known problems, such as: Buffer Overflow Risks; Format Strings problems; Race conditions; Shell Metacharacter dangers; and poor Random Number Acquisition (e.g. randomness). Potential security vulnerabilities are sorted by risk level based on the function performed and the value of the parameters of the function.				

Tool	Capabilities	Website	Linux/ Unix	Win 32	Cost
Bastille	Linux and Apache Harding		√	√	
Description	The Bastille Linux project has recently been working with the U.S. government to improve and harden the operating system security software. The project is called Fort Knox for Linux (FKL) that created documents on setting up Linux and Apache based on best practices. These documents are specific for Redhat and SUSE Enterprise Servers; however, they are general enough to be used with other types of Unix. The FKL project uses the existing code base of Bastille, incorporates the standards of the DoD, and makes Linux boxes hardened to this standard by running one program. The standard has incorporated the Linux Auditing Subsystem and removes unnecessary users, disables unencrypted services, turns off unused services, and handles many others focus areas as required by the DoD.				

End of Document