

**Defense Information Infrastructure (DII)
Common Operating Environment (COE)
Security Features Developers Guide (SFDG)
(Final)**

22 December 1999

Prepared for:

**Joint Interoperability and Engineering Organization
Defense Information Systems Agency**

This page intentionally left blank.

Table of Contents

1. INTRODUCTION.....	1
1.1 PURPOSE	1
1.2 SCOPE	1
1.3 APPLICABLE DOCUMENTS	2
2. SECURITY PHILOSOPHY.....	5
2.1 LEAST PRIVILEGE	6
2.2 SECURITY AUDIT.....	6
2.3 SYSTEM INTEGRITY TOOLS	6
3. SECURITY REQUIREMENTS	11
4. COE 4.X SECURITY ARCHITECTURE BASELINE	13
4.1 CROSS-PLATFORM	16
4.1.1 COE Administrative Domain.....	16
4.1.2 Accounts and Profile Manager.....	17
4.1.3 Password Policy.....	17
4.1.4 Common Data Store	17
4.1.5 Binding	18
4.1.6 COE Security Services Architecture Framework	22
4.2 PLATFORM-SPECIFIC SECURITY POLICIES.....	25
4.2.1 UNIX Platforms (Solaris and HP-UX).....	26
4.2.2 NT Platform.....	31
5. SECURITY COMPLIANCE	41
5.1 UNIX PLATFORMS (SOLARIS AND HP-UX)	41
5.1.1 Compliance Process.....	41
5.1.2 Compliance Requirements.....	42
5.2 NT PLATFORM.....	45
5.2.1 Compliance Process.....	45
5.2.2 Compliance Requirements.....	46
APPENDIX A. ACRONYMS/ABBREVIATIONS	53
APPENDIX B. NT BASELINE SECURITY CONFIGURATION.....	57
APPENDIX C. NT WORKSTATION SECURITY TEMPLATES	63

PREFACE

The following conventions are used in this document:

Bold	Used for information that is typed, pressed, or selected in executables and instructions. For example, select connect to host .
<i>Italics</i>	Used for file names, directories, scripts, commands, user IDs, document names, and Bibliography references; and any unusual computerese. For example, <i>iconified</i> , <i>setup.exe</i> , <i>DII/IC Document Delivery and Process and Style Guide</i> .
<u>Underline</u>	Used for emphasis. Also used for e-mail and web addresses. For example, The user <u>must</u> use these conventions or <u>www.saic.com</u> .
Arrows < >	Used to identify keys on the keyboard. For example, <Return>.
“Quotation Marks”	Used to identify informal, computer-generated queries and reports, or coined names; and to clarify a term when it appears for the first time. For example, “Data-Generation Report”.
Courier Font	Used to denote commands, command lines, and/or screen dumps as they appear on the screen. For example, <code>tar xvf dev/rmt/3mm</code> .
CAPITALIZATION	Used to identify keys, screen icons, screen buttons, field, and menu names. For example, the MAIN MENU button or hit the <F1> key.

This page intentionally left blank.

1. INTRODUCTION

This document provides software application developers and system integrators with a guide to the security features of Defense Information Infrastructure (DII) Common Operating Environment (COE) Version (V) 4.x. It is one of a series of documents prepared by the DII COE Security Engineering Office for the V4.x DII COE (see Applicable Documents in Section 1.3). Taken together, these documents provide a systems security engineering approach to the security of the DII COE, from requirements definition through operation and maintenance.

An understanding of the security features of the DII COE is essential for the development and integration of security-compliant software. This document provides insight into the DII COE Security Engineering Office's security philosophy for the V4.x DII COE, as well as the requirements that govern kernel and application development and integration. This document also outlines the security compliance process used by the Center for Integration (CFI) for the evaluation of developers' segments.

1.1 Purpose

The purpose of this document is to:

- 1) To explain the security philosophy that underlies the V4.x DII COE;
- 2) To reference the security requirements that apply to the DII COE kernel and applications;
- 3) To detail the security features available in the V4.x DII COE; and
- 4) To describe the security compliance process used to evaluate developer segments.

1.2 Scope

This document describes the security features contained in the V4.x DII COE that comprise the security baseline of the COE. Many of these security features are fundamentally related to the following operating system (OS) platforms:

- Sun Solaris 7;
- Hewlett Packard (HP) – UNIX (UX) V10.20;
- Microsoft Windows New Technology (NT) V4.0.

Specific information is provided for the UNIX (Solaris 7 and HP-UX 10.20) and NT platforms in separate subsections.

1.3 Applicable Documents

The following documents should be consulted for more information regarding security for the DII COE:

1. *Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Software Requirements Specification*, V4.1, 15 October 1999, DISA.

This document describes software requirements for the Defense Information Infrastructure (DII) Common Operating Environment (COE) security services. Security services comprise one of six platform services defined in the *Architectural Design Document for the Defense Information Infrastructure Common Operating Environment* (Defense Information Systems Agency [DISA], 1996), which is intended to be compliant with the *Technical Architecture Framework for Information Management* (DISA, 1995), including Volume 6, the *Department of Defense Goal Security Architecture* (DISA, 1994). The security services will function in a heterogeneous environment in the following six areas: accountability, availability, access control, confidentiality, integrity, and non-repudiation. The security requirements specified in this document will be allocated across the COE platform services areas.

2. *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS)*, V4.0, October 1999, DISA.

This document describes the technical requirements for using the Defense Information Infrastructure (DII) Common Operating Environment (COE) to build and integrate systems. It provides implementation details that describe, from a software development perspective, the following:

- the COE approach to software reuse,
 - the COE runtime execution environment,
 - the definition and requirements for achieving DII compliance,
 - the process for automated software integration into the COE or into a COE-based system, and
 - the process for electronically submitting/retrieving software components to/from the DII repository.
3. *Defense Information Infrastructure (DII) Common Operating Environment (COE) UNIX Kernel Developer's Security Guidance*, June 1999, DISA.

The primary purpose of this document is twofold: to provide recommendations to DII COE kernel developers in configuring the UNIX operating system in a secure manner and to ensure that those developers have the information necessary to develop software that does not compromise a secure configuration. A secondary purpose is to provide guidance to developers of software that resides above the COE kernel. Those developers must consider other areas that are not discussed in this document, but much of the guidance included herein is also applicable to them.

4. *Defense Information Infrastructure (DII) Common Operating Environment (COE) Windows NT Application and Kernel Developers' Security Guidance*, July 1999 (Draft), DISA.

The purpose of this document is to provide recommendations to DII COE application software developers and Kernel software developers for designing secure applications for the DII COE NT environment. Application software should not compromise or weaken the security posture of a properly configured DII COE NT installation. Kernel software should increase the security posture of a NT OS installation and provide additional security services required for DII COE environments.

5. *Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Administrator's Manual (SECAM) for Kernel Version 4.1.2.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)*, December 1999, Jet Propulsion Laboratory.

This document describes the Security Administration utilities available with the DII COE Kernel on HP-UX 10.20, Solaris 7, and Windows NT 4.0. The capabilities and functions of the Account and Profile Manager, or APM, are discussed in depth. Additional Security Administration operations, such as configuring system security settings and maintaining audit log files, are also described in this document.

6. *Security Compliance Checking In VerifySeg*, Function Design Document (FDD), Rev 1.1, March 31, 1999, Inter-National Research Institute.

This document describes the requirements and a notional design for security compliance checking to be performed by the COE tool, `VerifySeg`. The intent is to extend the `VerifySeg` tool so that it will analyze a segment to the fullest extent possible to ensure that the segment complies with the security requirements identified in the *Integration and Runtime Specification (I&RTS) 4.0* document. The tool shall be used on all 4.x or later segments, and may be used against earlier segments at the appropriate Program Manager's discretion.

7. *DII COE 4.1.2.0 Security Features User's Guide (SFUG)*, 22 December 1999, DISA.

The SFUG describes the COE security features available to the user and user security responsibilities and activities within a COE system. The document provides a regular system user with the fundamental information required to access and operate securely in the COE system environment. The SFUG also includes information and recommendations on how to recognize and minimize security risks. Finally, the document augments the security-relevant portions of the commercial off-the-shelf (COTS) manuals for the systems and utilities that comprise the DII COE. Where additional or more detailed information is available, reference is made to the appropriate manual.

8. *Security Test Procedures for DII COE Kernel Version 4.1.2.0*, December 1999 (Draft), DISA.

These test procedures are designed to identify, test and document the security posture of the DII COE kernel. The procedures were prepared for the DII COE independent of the manner in which the COE may be used for mission-critical systems. The test procedures are based upon the security requirements found in the *Security Software Requirements Specification* document, and the results are used to develop a Requirements Traceability Matrix for the kernel. The primary audiences for this document are the DII COE Security Engineer and the COE kernel developer. System integrators and site administrators may use the test procedures as part of their system/site accreditation process.

2. SECURITY PHILOSOPHY

The security philosophy underlying the DII COE emphasizes, to the maximum extent possible, reliance upon commercial off-the-shelf (COTS) functionality, within the kernel and Infrastructure Services layer of the COE, for the implementation of security features. For example, this means that access to files and directories is enforced by the underlying OS and not by Government off-the-shelf (GOTS) software within the COE. Applications must still be designed appropriately to cooperate with the underlying OS, but decisions about whether or not a specific user can access a capability is controlled by groups, Access Control Lists (ACLs), etc., as provided by the native OS. This approach is intentional because it is more cost effective than building Department of Defense (DoD)-unique solutions, because it conforms to accepted commercial standards, and because it prevents the COE from extending the boundaries of the Trusted Computing Base (TCB).

There are a few exceptions to this philosophy in areas where the DII COE Security Engineering Office has determined that commercial solutions are not totally adequate to address DoD concerns. These exceptions, in the present DII COE implementation, are primarily¹ in the areas of user account lockout, encryption services for communications, and user inactivity timeouts. Acceptable commercial solutions for these areas are being sought and will eventually replace the present GOTS solutions. Refer to the DII COE Chief Engineer and the *DII COE Buildlist Worksheet* for current status information.

The DII COE security philosophy is based on an approach that emphasizes security configuration and control, then detection of aberrant activities. System/security administrators will be able to make use of the security features provided by the OSs, and COTS/GOTS tool products, to aid in the implementation and maintenance of the security configuration.

The security philosophy consists of three principal elements:

1. Least Privilege,
2. Security Audit,
3. System Integrity Tools.

The least privilege element is designed to protect the security posture; the security audit and system integrity tools elements are intended to detect and recover from any compromise of the security posture. Each of these elements is explained in the subsections below.

Procedural issues such as proper labeling of electronic media, requirements for maintaining paper trails showing originating authority, etc., are not addressed. These procedural issues lie outside

¹ Profiling as defined and provided by the COE is *not* intended to be a security enforcement technique. Its purpose is solely to reduce the Graphical User Interface (GUI) desktop declutter by displaying only those icons, menus, etc., to the user that the user should have access to. Actual granting or denying access to system resources is arbitrated by the underlying OS, database management system, network software, etc., and not by COE profiling software.

the scope of the DII COE and are more properly addressed by traditional system security policy and Concept of Operations (CONOPS) documents.

2.1 Least Privilege

The least privilege concept states that users should be provided with the minimum amount of information or access necessary to perform their specific job functions. For example, files and directories with world-read, -write and -execute permissions do not enforce the least privilege concept. Providing these permissions to the owner of the files and directories, and discrete group access if required (e.g., read permission), would be an example of exercising least privilege. Implementation of the least privilege concept begins with the DII COE kernel, and must be extended to each segment that comprises the DII COE (or a specific system such as Global Command and Control System [GCCS]) to maintain the security posture. The *DII COE 4.0 I&RTS* mandates the implementation of the least privilege concept through numerous compliance requirements (e.g., the UNIX umask setting of 027).

2.2 Security Audit

Security audit is a means to create a chronological record of system and application security-related activities that can be used to reconstruct, review, and examine a sequence of activities or events. The DII COE UNIX kernels are configured to enable underlying security modules (e.g., BSM for Solaris, C2 for HP) that are required for audit, but the precise configuration is system specific. System engineers are responsible for configuring how much auditing is performed, and what is audited, within the OS. Segment developers, however, must ensure that their segment can operate in an environment in which the OS security services are enabled.

Security relevant events to be audited that relate to a segment shall be written to the OS audit trail, an OS log (e.g., UNIX's *syslog*, *NT Application Log*) or a log created by the segment. Selection of a particular log may be driven by non-security considerations (e.g., functional or performance events to be captured). For the UNIX platform, the *syslog* is preferred as an easy-to-read central repository if no other considerations exist. Rationale for the selected log should be provided by the segment developer during design reviews and included in the segment's *Software Version Description (SVD)* documentation or its equivalent.

Segments that write audit information to a log shall include the segment prefix in the output. This is required so that audit information can be traced to a specific segment.

2.3 System Integrity Tools

The following tools are available in the DII COE to maintain the security posture of COE-based systems:

- Security Profile Inspector (SPI) – SPI is a security inspection tool developed by Lawrence Livermore National Laboratory. SPI supports multi-host system security inspections managed from a designated "command host." These inspections include access control testing, system file authentication, file system change detection, password testing, and checks for a variety of common system vulnerabilities.

- Crack – Crack is a UNIX password-guessing program that performs a dictionary attack on a password file. The Crack program encrypts each word in its dictionary (as well as permutations of the dictionary words and permutations of the user names found in the password file) using the same encryption algorithm used to encrypt user passwords and compares the results with the encrypted entries in the password file. If the encrypted dictionary word matches a user's encrypted password, the corresponding plain-text dictionary word is identified by the Crack program as the user's password.
- Satan – Satan is a tool for finding system and network vulnerabilities. Because the tool is freeware, maintenance support and upgrades are limited. Satan should therefore not be relied upon to identify the most current UNIX vulnerabilities.
- Tripwire – Tripwire detects unauthorized modifications to files. It does this by initially building a master database of unique fingerprints for each file, and later comparing current fingerprints with the master's. Differences indicate a file change that the system/security administrator should be aware of. The system/security administrator can designate a specific set of files for monitoring. The tool can notify the system administrator of corrupted or tampered files so that damage control measures can be taken.
- TCP Wrapper – TCP Wrapper is used to selectively log and filter access to services offered by *inetd* on a UNIX system. TCP Wrapper (*tcpd*) is installed between *inetd* and the network server programs (such as *telnetd*, *ftpd*, *fingerd*). When a new connection arrives, *inetd* passes it to *tcpd* which performs the appropriate access control tests and, if those tests are passed, invokes the network server program. TCP Wrapper's access control tests are typically governed by rules contained in the */etc/hosts.allow* and */etc/hosts.deny* files. For each new connection, *tcpd* checks the remote system's network address against the contents of *hosts.allow* for a rule that explicitly grants access to that address. If this fails, *tcpd* checks *hosts.deny* for a rule that prevents access. If no rules from either database apply, access to the service is granted.
- UNIX Security Policy Configuration Tool (SPCONFIG) – SPCONFIG is designed to configure and analyze Unix workstations and servers according to a specified security policy. An initial security configuration can be applied to a Unix system and then periodically checked for deviations. If needed the original policy can be reapplied to bring a system back into compliance. A key objective for the SPCONFIG tool is to engineer a secure baseline that complies with a specified security policy. Users can invoke a standard security policy such as the DII COE or a policy tailored to an organization's specific requirements.
- Microsoft Security Configuration Editor – Microsoft Security Configuration Editor is a Microsoft Management Console (MMC) snap-in tool designed to reduce costs associated with security configuration and analysis of the Windows NT operating system. The Security Configuration Editor allows one to configure security for a Windows NT-based system, and then perform periodic analysis of the system to ensure that the configuration remains intact.

- Windows NT Security Utilities (SECUTL) Infrastructure Services Component – The SECUTL infrastructure services component provides a transparent set of security utilities that augment and/or enhance the native services provided by Microsoft's Windows NT 4 operating system. This set of utilities was developed for the Top Secret/SCI environment, but have important applicability at all security levels. The SECUTL set of utilities provides maximum flexibility to the NT System Administrator to choose the appropriate set of functionality for the given local security and configuration requirements.

The permissions and group identifiers for these tools have been configured to restrict unauthorized access.

This page intentionally left blank.

3. SECURITY REQUIREMENTS

The principal document for security requirements is the *DII COE Security Services Software Requirements Specification* (SRS). The security attributes and capabilities required by this document are intended to establish a consistent security configuration and capability across the DII COE. However, the security features provided by the DII COE do not currently meet all of the requirements defined in the SRS, particularly in the areas of Multiple Security Levels (MSLs) or Multi-Level Security (MLS) and secure applications (e.g., mail). The V4.1 DII COE's compliance with the SRS will be evaluated and documented in a *DII COE 4.1.2.0 Requirements Traceability Matrix*.

The *4.0 DII COE I&RTS* provides more detailed security requirements for the DII COE kernel and applications (segments). These requirements are the foundation for determining security compliance (see Section 4. of this document), and hence Government acceptance criteria. To comply with these requirements, segment developers will need to incorporate security into the requirements, design and analysis, and development phases of the software development lifecycle.

The *UNIX Kernel Developer's Security Guidance* (KDSG) document provides specific security guidelines to be implemented during the software development lifecycle. These guidelines are intended to provide a practical and detailed enumeration of the security philosophy and high-level security requirements contained in the SRS. Segment developers should become familiar with this document because:

- It describes, from a security perspective, the integration and runtime environment, in which segment developers must operate; and
- The majority of the guidance is applicable to segment developers and required by the *4.0 DII COE I&RTS*.

The UNIX KDSG document will also provide segment developers with the rationale for many of the security requirements contained in the *4.0 DII COE I&RTS*.

The *Windows NT Application and Kernel Developers' Security Guidance* (AKDSG) document performs a similar purpose for the NT platform. This document, however, is specifically targeted to both the application (segment) and kernel developer audiences. The DII COE NT security architecture, which may be less well understood by the DII COE community because of the platform's relatively recent introduction, is cogently summarized. This document details, from a security perspective, the integration and runtime environment in which segment developers must operate on the NT platform.

Collectively, the security requirements contained in the documents cited above provide a baseline security configuration for the DII COE that the Commanders-in-Chief (CINCs), Services and Agencies (C/S/A) can augment. If a specific operational requirement mandates additional security features above and beyond those provided by the DII COE, it is the responsibility of that system's owner, designers, and integrators to incorporate those additional features. The security

configuration and security features of the DII COE may not be sufficient to meet all systems' security policy and requirements.

Implementation of these security requirements provides system designers and integrators with the basic building blocks to include sufficient security features in their systems so that system developers/integrators can build on these basic security building blocks to successfully achieve accreditation to operate in a system high environment. The security attributes and features of the DII COE can be used to support this process, along with supporting documentation for individually certified components. The DII COE approach states minimal security requirements because it is ultimately up to the cognizant Program Manager to make the final decision on what is acceptable or not acceptable for the end system.

The SRS does not dictate implementation mechanisms or certification and accreditation (C&A) requirements to individual program systems. In fact, systems will likely support a unique security policy and set of security requirements based on program security level, discretionary control, and operational requirements. The combination of the security services provided by the DII COE and the requirements defined by the SRS provides a basis for a security implementation that will supplement or possibly meet the security requirements of individual programs. In addition to the SRS and services, the DII COE provides a set of security test procedures for the DII-approved products and platforms that support secure system configuration and operation.

4. COE 4.x SECURITY ARCHITECTURE BASELINE

The security architecture baseline for DII COE V4.x consists of the security-related documentation and software, features and capabilities, and security engineering processes that support a fielded release. This baseline is the sum of the efforts undertaken during the COE security engineering lifecycle.

The security architecture baseline is best enumerated by identifying its principal individual elements. These elements, while not exhaustive, constitute a “Cordon COE”, a core collection of security products, features, capabilities and processes that define the COE’s security posture.

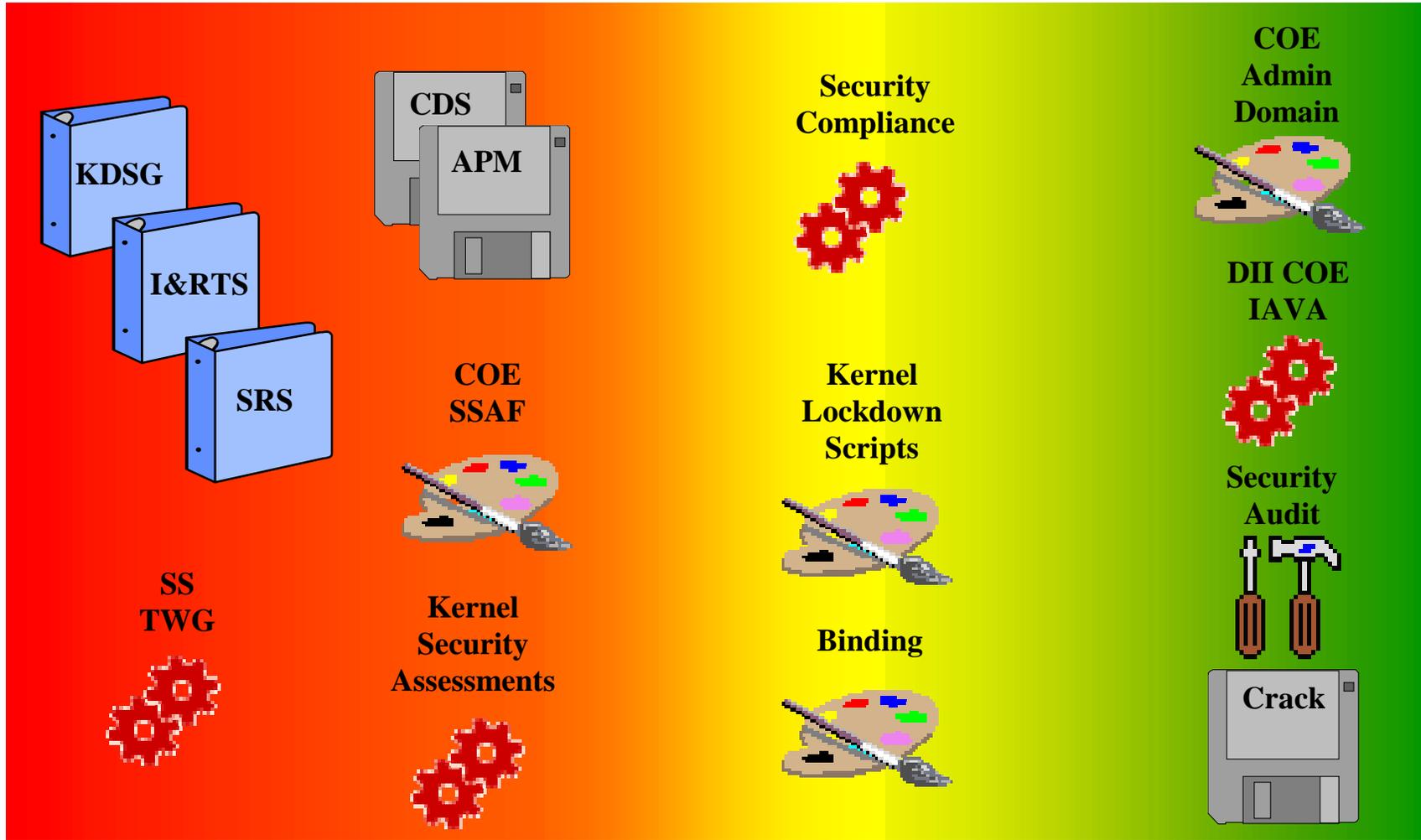
Table 4.-1 Principal Elements of the COE Security Architecture Baseline

Documentation
Security Requirements Specification
Integration & Runtime Specification
Kernel Developer’s Security Guidance (UNIX)
Application and Kernel Developers’ Security Guidance (NT)
Security Features Developers Guide
Security Administrator’s Manual
Security Features User’s Guide
Software
APM
CDS
Security Patches
Operating System Patches
System Integrity Tools
VerifySeg Security Module
Features
COE Admin Domain
Kernel Lockdown Scripts
Binding
User Account Lockout
Deadman
COE SSAF
Capabilities
Identification and Authentication
Discretionary Access Control
Security Audit
Encrypted Communication
Security Engineering Processes
Kernel Security Assessments
Security Compliance (Segments)
Security GSPR Analysis and Resolution
DII COE IAVA Implementation Process

Security Services Technical Working Group
Hardware and Software Security Assessments

The development of the security architecture baseline can be depicted by representing example individual elements within the appropriate phase of the COE security engineering lifecycle. This illustration underscores the contribution of each phase of the security engineering lifecycle to the baseline.

**Figure 4.-1 Development of the COE Security Architecture Baseline:
The COE Security Engineering Lifecycle**



**Requirements
Definition**

**Design &
Development**

Implementation

**Operation &
Maintenance**

	Documentation		Feature		Security Engineering Process
	Software		Capability		

The security architecture baseline for DII COE V4.x consists of both COE-wide and platform-specific elements. COE-wide elements pertain to a cross-platform environment; platform-specific elements are designed to address vulnerabilities or other considerations unique to the UNIX or NT OS.

4.1 Cross-Platform

The security architecture baseline consists of a number of elements that are cross-platform. These COE-wide elements are defined in the following subsections.

4.1.1 COE Administrative Domain

A COE administrative domain (AD) is a defined collection of Internet Protocol (IP)-addressed hosts running DII COE V4.x that are managed as a unit. An AD can contain any number of individual hosts (Solaris, HP-UX and NT), NT domains and Network Information Service Plus (NIS+) domains.

The AD is managed through use of the Accounts and Profile Manager (APM) application (See Section 4.1.2.). Each AD contains a single host that is identified as an Account and Profile Manager (APM) Master. The APM Master is responsible for maintaining the definitions of all the accounts and profiles defined within the AD.. Each host within the AD is an APM Server (Local), with an APM Client application. The APM Client application provides the same user interface for account and profile management, independent of the host platform

The three tier architecture (APM Master, APM Server (Local), and APM Client) of the AD has several advantages:

- Individual accounts and profiles are defined once. Once defined, any host or network account manager (e.g., NIS+ or NT domains) can be instructed to implement the account or profile. Effectively, the master definition acts like a template. This streamlines the process of account/profile management and ensures that the definitions are consistent across the AD.
- Individual accounts and profiles can be defined on one, all, or only a subset of the hosts within the AD.
- An identical account/profile can be defined on HP, Solaris and NT hosts. APM works with the native OSs to ensure that the account/profile is properly implemented.
- The information concerning an account/profile is maintained on the host that serves the account information. This distributes the processing load and reduces the changes of a single point failure.
- Both the Master and Server APMs maintain account and profile information in the Common Data Store (CDS) (See Section 4.1.4.).

4.1.2 Accounts and Profile Manager

The APM provides the ability for authorized users to create, delete, and maintain user accounts and groups, as well as define profiles that provide users with easy access to the executables and icons they need to perform their duties.

Refer to the *Security Administrator's Manual (SECAM) for Kernel Version 4.1.2.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)* for specific information regarding the operation of APM.

4.1.3 Password Policy

A common password policy has been established for the DII COE that is based upon the native capabilities provided by the Solaris, HP-UX and NT operating systems. The password policy enforces the following rules on an individual host machine and within an AD:

- Password must be at least 8 characters long;
- Password must include at least one numeric, case change, or special character (e.g., 0-9, &, %);
- User names are prohibited within a password;
- Minimum password age before change is 7 days;
- Maximum password age before expiration is 91 days;
- The user shall receive a warning 7 days prior to password expiration;
- The initial login password is expired, requiring the user to change this password immediately upon login;
- A graphical user interface (GUI), the COE Change Password tool, is provided for changing passwords.

To ensure a consistent password throughout the AD, specific steps should be followed when a user changes his/her password due to imminent password expiry or for safety's sake. These steps are documented in the *Security Features User's Guide*.

4.1.4 Common Data Store

The CDS provides a persistent repository of data for the DII COE. In COE V4.x, the CDS is used by the APM to store user, group, profiles, feature and host data, and by the COE Installer to store information about segments and processes. The CDS is not a distributed database. Data from the CDS is only available to applications running on the same host as the repository.

The repository is made up of five logical storage areas, each of which has its own access controls. The five areas are:

1. *LocalHostPrivate* – contains information that is shared by applications on the local machine, but which should only be accessed by the most privileged applications. Information in this repository may be read and written only by the user “root” (UNIX) or “Administrator” (NT). Objects stored in this area must have */LocalHostPrivate* as the first element of their class name.
2. *LocalHost* – contains information that is shared by applications on the local machine. Information in this repository may be read by any application, and may be written by any application that is running within the group “admin” (UNIX) or Administrators (NT). Objects stored in this area must have */LocalHost* as the first element of their class name.
3. *MasterHost* – contains information about the administrative domain (a collection of machines managed by a single server). Information in this repository may be read by any application, and may be written by the user root (UNIX) or Administrator (NT). Objects stored in this area must have */MasterHost* as the first element of their class name.
4. *UserPrivate* – contains information that is shared by applications run by a particular user, but which should only be accessed by applications run by that user. There is a separate storage area for each user/host login. Information in this repository may be read and written only by the user who owns the repository). Objects stored in this area must have */UserPrivate* as the first element of their class name.
5. *User* – contains information that is shared by applications run by a particular user. There is a separate storage area for each user/host login. If a user logs in on Host1, he has a separate CDS User area from his login as the same user on Host2. Information in this repository may be read by any application, and may be written only by the user who owns it. Objects that do not match the fully qualified object names for any of the other four areas of CDS will be put in this area, but it is considered better form if the object has */User* as the first element of the class name.

Segment developers should refer to the *Programmer's Guide and Reference Manual (PGRM) for Kernel Version 4.1.2.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)* document for more detailed security-relevant information about the CDS.

4.1.5 Binding

Binding, a new feature within DII COE V4.x, provides the ability for integrators and site administrators to tailor the access permissions (e.g., users, groups) delivered by the segment developer to meet their specific mission and security needs. A binding is comprised of two parts: a symbolic name and one or more actual names. The segment developer associates files with symbolic owners and symbolic groups. The integrator or site administrator uses a binding segment to define which actual owners and actual groups should be used when the segment is loaded.

Segment developers specify symbolic users and groups using keywords in the FileAttribs descriptor. The integrator examines the symbolic groups and users specified in FileAttribs and decides:

- Which actual groups and users to create; and
- The mapping of the *symbolic* groups and users to the *actual* groups and users.

Figure 4.1.5-1 shows two sample segments. Each segment contains a number of files that the segment developer has associated with an owner and group. Segment A assigns files to five different symbolic owners and three different symbolic groups. Likewise, segment B uses four different symbolic owners and three symbolic groups. Note that both segments use the group admin even though it is unclear whether symbolic group admin, defined in segment A, is the same symbolic group as defined in segment B. To minimize ambiguity, the segment prefix is always associated with symbolic bindings (e.g., Segment A’s prefix is SEGA, Segment B’s prefix is SEGB).

Segment A			Segment B		
File	Owner	Group	File	Owner	Group
FileA	tom	admin-support	FileL	kirk	admin
FileB	dick	admin-support	FileM	spock	sub-command
FileC	harriet	admin	FileN	scotty	sub-command
FileD	sysadm	admin	FileO	zulu	helm
FileE	tracker	trackops			

Figure 4.1.5-1. Two Sample Segments

The system integrator or site administrator determines that he/she doesn’t want to use all of the groups based upon security or administrative requirements. In fact, it is determined that only three groups are needed:

1. Admin,
2. Management,
3. Operations.

Note: These three groups could be groups already used by other segments installed on the host or could be new groups either created by the APM or via the \$GROUPS directive.

To make these changes, the integrator or administrator creates a binding segment that specifies that:

- Segment A’s admin-support and admin symbolic groups should be combined into the actual group admin;
- Segment B’s admin symbolic group should be mapped to the actual group management; and

- Segment A's trackops symbolic group and Segment B's sub-command and helm symbolic groups should be combined into the actual group operations;
- The actual groups defined by the binding segment will be used in the COE AD (DOMAIN). (If a parameter is not specified, the default is LOCAL.)

The following syntax, using the Bind segment descriptor, is used in the binding segment's SegInfo file for the mapping described above:

```
[Bind]
$GROUP:admin:36:DOMAIN
SEGA:admin-support
SEGA:admin
$GROUP:management:400:DOMAIN
SEGB:admin
$GROUP:operations:350:DOMAIN
SEGA:trackops
SEGB:sub-command
SEGB:helm
```

Figure 4.1.5-2 Bind Segment Descriptor

See Section 6.5.3.2, Bind of the *4.0 DII COE I&RTS* for additional information about the use of the Bind descriptor.

There are two important lessons in this example:

1. Bindings can occur within segments and across segments; and
2. Bindings can be used to combine groups and owners specified by the developer. However, they cannot be used to cause two files owned by the same user or group to be assigned to different actual users or groups. Users or groups can be combined, but cannot be broken apart.

Figure 4.1.5-3 depicts the binding process.

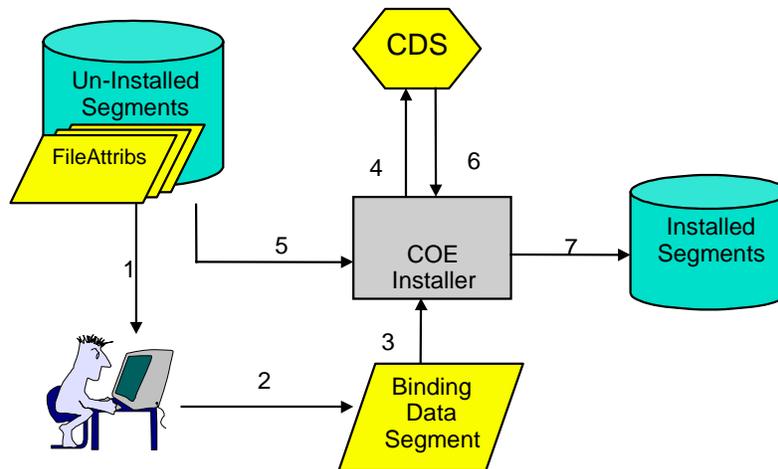


Figure 4.1.5-3. The Binding Process

The integrator analyzes the segments he/she wishes to bind (Step 1). The integrator constructs a data segment (see the *I&RTS* for the exact format) specifying the bindings and any accounts/groups that need to be created (Step 2). The integrator loads the data segment via the COE Installer (Step 3). This process creates any necessary user accounts and groups in addition to storing the binding information in the local and/or master CDS (Step 4). The integrator then loads the developer-delivered segment using COE Installer (Step 5). The COE Installer references the binding information stored in CDS to retrieve the mapping of accounts and groups defined for the segment being installed (Step 6). The COE Installer installs the segment using the actual users and groups defined by the mapping (Step 7).

Segment developers must use judgment when assigning files to symbolic owners and symbolic groups during the development process. For example, if a segment provides three distinct capabilities, then the segment developer should associate each capability with a distinct “symbolic” owner and group. This allows the integrator to either assign each delivered capability to an actual group, or to combine several “symbolic” groups into a single actual group. Additionally, it is important that segment developers do not assign files and applications that must work together to different symbolic groups/owners. For example, if application A accesses file B, then either A and B should be associated with the same symbolic group or owner, or the segment’s documentation should clearly detail the relationship. Otherwise, the integrator may associate each with a different actual group that may prevent the application from running correctly.

Segment developers and system integrators should note the following:

- An integrator can take the little pieces delivered by a segment developer and make big pieces (i.e., they can combine multiple symbolic groups or owners into a smaller number of actual groups or owners). However, the reverse is not true. When it doubt, err on the side of using more symbolic owners and groups; and
- If a segment developer requires strict usage of the owners and groups as specified in the files delivered with a segment, then these should be noted in the segment

documentation. In fact, developers should use the segment documentation to inform the integrators of any special binding considerations (i.e., security concerns, why the files are partitioned as they are, etc.).

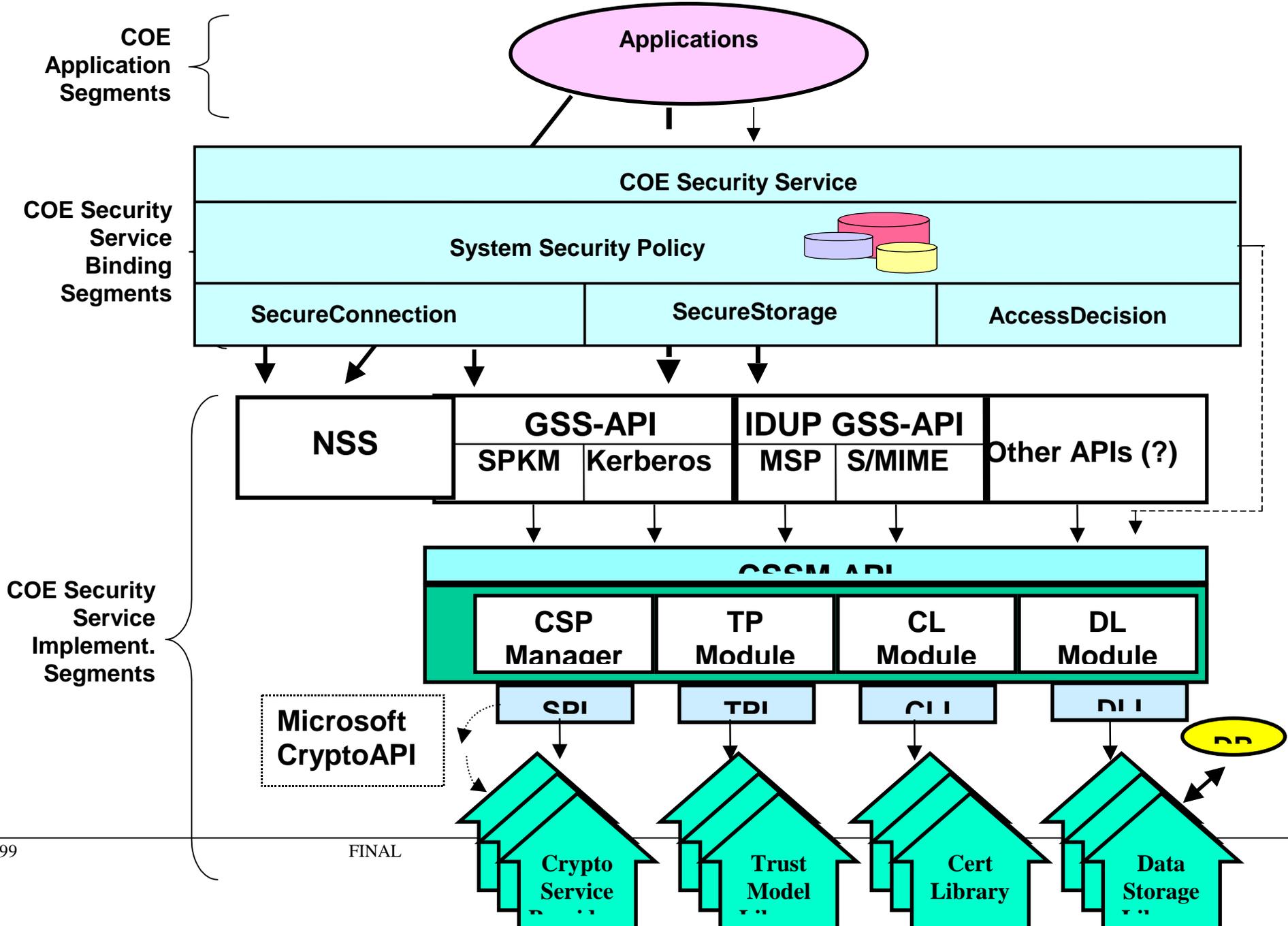
4.1.6 COE Security Services Architecture Framework

The DII COE Engineering Office is developing a DII COE Security Services Architecture Framework (SSAF) that includes an application programming interface (API) specification and an associated binding implementation. The three main areas of effort are:

1. COE SSAF and DoD Public Key Infrastructure (PKI) technical coordination;
2. Development of an API specification for the COE 4.x baseline; and
3. Design and development of an initial Netscape Security Services (NSS) binding implementation.

The SSAF is depicted in Figure 4.1.6-1.

Figure 4.1.6-1 Security Services Architecture Framework



The COE SSAF APIs are a library of functions that can be used to secure communication between two processes. These APIs are explicitly designed for distributed applications that use point-to-point Transmission Control Protocol/Internet Protocol (TCP/IP) communication. Thus, in the present DII COE V4.x implementation, the API specification does not support in-place encryption, digital signing, etc. The APIs provide facilities for authentication, confidentiality and integrity of data transmitted across a TCP/IP socket.

The COE SSAF APIs are designed to be used by those distributed applications that are central to the DII COE kernel or at the lower levels of the COE. The APIs are also available to any developer to meet security requirements using the functionality provided by the APIs.

A binding to NSS, using the COE SSAF APIs, is currently under development. Netscape has developed a product that represents the libraries used by the browser and server product lines to implement the Secure Sockets Layer (SSL). SSL makes use of X.509v3 certificates to authenticate communicating principals with each other and to encrypt and validate transmissions between the principals.

The NSS Binding of the COE SSAF APIs can utilize the DoD PKI. The DoD PKI is the source of X.509v3 certificates that will be used to identify DoD personnel. Both the NSS Binding and the DoD PKI make use of the key and certificate databases implemented by Netscape Communicator to install and access the user's private key, public key, and certificate. This information can be exported to PKCS #12 format file (*.p12) if they are needed by applications that are SSL-enabled but do not use the NSS libraries or the COE SSAF APIs.

The NSS Binding is provided in two languages, C and Java. It also provides an administrative interface through which a System Security Administrator can specify which NSS-supported encryption algorithms to use and under what circumstances to allow unencrypted and/or unauthenticated communication.

The COE SSAF V1.0 will be fielded in a future COE version. The architecture will be made available to segment developers prior to release in a COE version.

4.2 Platform-Specific Security Policies

As part of the COE V4.x security architecture, security policies have been defined and implemented for the UNIX and NT platforms to "lock-down" the kernel upon installation. These security policies address configuration and implementation considerations such as discretionary access control, access to privileged roles and programs, etc. The locked-down kernel provided upon installation represents the security architecture baseline for DII COE V4.x.² Segments should be designed and developed to operate on this baseline without modification. The security policies for the UNIX and NT platforms are described in the subsections below.

² The security baseline for the NT platform requires installation of security templates (*.inf files) via Microsoft's Security Configuration Editor after kernel installation. Details are provided in Appendices B and C.

4.2.1 UNIX Platforms (Solaris and HP-UX)

4.2.1.1 UNIX Lockdown Scripts

The security policy for the UNIX platform is implemented through the execution of multiple scripts during kernel installation. These scripts are located in the */h/COE/bin/security* directory. Access to this privileged directory is limited to *root* and members of the *admin* group. Segment developers and system integrators are encouraged to review the content of these scripts to become familiar with the security architecture baseline.

Table 4.2.1.1-1. UNIX Lockdown Scripts

Script File Name	Description
<i>etc_mods</i>	Modifies several files under the <i>/etc</i> directory.
<i>Kernel_permission_list</i>	This script is an input file for <i>set_permissions</i> .
<i>OS_mods</i>	Modifies operating system files.
<i>OS_permission_list</i>	This script is an input file for <i>set_permissions</i> .
<i>Set_permissions</i>	Sets kernel and operating system file permissions.
<i>Set_perms.sh</i> (Solaris only)	Sets accounts with no password to a <i>/bin/false</i> shell.

The */h/COE/bin/security* directory also contains scripts to “un-lock” the security policy. Prior to executing these scripts, system integrators should inspect the scripts to determine their impact on the security posture of the system.

Table 4.2.1.1-2. UNIX Unlock Scripts

Script File Name	Description
<i>Security.conf</i>	The configuration file for enabling and disabling specific aspects of the kernel lockdown.
<i>SecuritySetup.pl</i>	The script executed to enable/disable specific aspects of the kernel lockdown.

4.2.1.2 Discretionary Access Control

Discretionary access control (DAC) is the principal means to implement the least privilege element of the DII COE security philosophy. This access control is implemented on the UNIX platform through UNIX’s owner, group and world directory and file permissions. Emphasis has been placed on removing unrestricted world access and controlling access at the group level through the use of discrete group identifiers.

4.2.1.2.1 Kernel Directory and File Permissions

The DII COE V4.1 kernel is pre-configured as follows:

- */h* permissions are set to 755.
- */h/data*, */h/data/global*, and */h/data/local* permissions are set to 755.
- */h/AcctGrps/SecAdm* and */h/AcctGrps/SysAdm* permissions are set to 750.
- */h/COE* permissions are set to 755.
- */home1*, */home2*, etc., directory permissions are set to 755.
- All other COE subdirectories created by the kernel are set to 755.
- COE kernel executables are set to 755 and COE kernel data files are set to 644. (Kernel directories and files have limited world access permissions for basic kernel resources required by all system users; this is an approved exception to Table 4.2.1.2-1 below and group-level access controls.)

All other COE subdirectories and files are set in accordance with Table 4.2.1.2.2-1 below.

4.2.1.2.2 Segment Directory and File Permissions

Segment directory and file permissions, whether created at install time or runtime, shall not be less restrictive than identified in Table 4.2.1.2.2-1 to achieve Level 6 compliance. Any such directories or files that do not meet the permissions identified in the table shall be documented in the *SVD* document or its equivalent. The *VerifySeg* security module uses this table to check permissions, and all warnings from the check are explained in the *VSOOutput* file.

Table 4.2.1.2.2-1. UNIX Segment Directory and File Permissions

Dir Name	Dir Permissions	File Permissions
<i>SegDir</i>	750	N/A
Scripts	750	750
SegDescrip	750	750
bin	750	750
lib	750	750
data	770	660
DBS_files	770	660
install	750	750
man	750	440
<i>all other</i>	750	750

4.2.1.3 System Startup Environment, umask, and Search Path

In DII COE 4.x, the *umask* is set to 027 which means that newly created files are not accessible by world, and are writable only by the file owner. The DII COE kernel establishes the *umask* 027 both at system startup time and at user login time. DII compliance specifically prohibits ordinary segments from altering the *umask* setting, although with Chief Engineer approval and properly documented in the *SVD* or its equivalent, an account group segment may establish a different *umask* setting.

The startup environment (whether at boot time or at user login) also includes search path rules to locate executables (e.g., *path*) and shared libraries (e.g., *LD_LIBRARY_PATH*). Any files or directories that are specified by these rules, or are directly accessed, are security relevant. Such files, if modified, could introduce security vulnerabilities by introducing malicious software in place of legitimate services. These environment variables are established by the COE and may only be changed by Chief Engineer approval.

The search path must also be protected so that it does not include relative directory names (e.g., “~”) or the current directory (e.g., “.”) since this would otherwise introduce easily exploitable security vulnerabilities. In addition, the search path rules should be initialized at the beginning of each startup script. It is best to use full pathnames for all commands, rather than rely on the search path.

4.2.1.4 Network Services

Network services that are vulnerable to exploitation have been disabled in COE V4.1 by default. Refer to Section 4.2.2.5 of the *I&RTS* for a list of disabled services and the accompanying rationale.

4.2.1.5 Privileged Roles and Programs

4.2.1.5.1 Root Account

The DII COE kernel is configured to disallow direct login to *root*. Some system integrators may find that this kernel configuration setting is too restrictive for their CONOPS. A system integrator may elect to enable direct *root* login without being non-DII compliant, but he/she should be aware of the potential security vulnerabilities of doing so. If a system integrator enables *root* login, he/she shall document the change.

4.2.1.5.2 System and Security Administrator Accounts

The subdirectories for the System and Security Administrator accounts, */h/AcctGrps/SysAdm* and */h/AcctGrps/SecAdm* respectively, have restricted permissions of 750 to prevent unauthorized access. In addition, access to privileged programs in these directories has been restricted at the group level.

4.2.1.6 Shell Scripts

4.2.1.6.1 SUID/SGID Shell Scripts

Segments shall not contain or create any shell scripts that *SUID* or *SGID* to root to achieve Level 5 compliance. Compiled code is used for privileged programs. The *VerifySeg* security module will issue a warning for any segment that violates this requirement.

4.2.1.6.2 C Shell Scripts

Segments shall not contain or create C shell scripts to achieve Level 5 compliance. The Bourne or Korn shell is used for shell scripts. The *VerifySeg* security module will issue a warning for any segment that uses C shell scripts.

Restrictions upon C shell scripts are necessary because a user's *.cshrc* file is executed by default when a C shell program is executed. A malicious user could insert a command in his/her own *.cshrc* file that would be run everytime a C shell program is executed. Additionally, a C shell script has the ability to suspend a foreground process, potentially giving a user command line access.

4.2.1.7 Security Audit

The *DII COE System Administrator's Manual* and *Security Administrator's Manual* documents provide information regarding the OS's default audit activation and configuration for security. The precise configuration is system specific. System integrators and/or site administrators are responsible for configuring how much auditing is performed, and what is audited, within the OS.

4.2.1.8 Restricted User Environments

A restricted user environment can be used to create user accounts with restricted access to the local system. This can be accomplished using a restricted shell, such as the restricted Korn shell (*rksh*). For proper configuration, the user's home directory, *.profile*, and other startup files shall not be writable by the user. A restricted binary directory, such as */usr/rbin* shall be created, with only a minimal set of commands that the user is authorized to run. Finally, the *.profile* file shall specify a search path of */usr/rbin*. The search path should not contain any directories that are writable by the user. If the restricted shell permits, the following environment variables should be configured to be read-only: *LD_PRELOAD*, *LD_LIBRARY_PATH*, *SHELL*, *IFS*, and *PATH*.

4.2.1.9 Command Line Access Restrictions

The DII COE kernel has been designed so that command-line access is not required. However, *xterm* and *dtterm* have not been disabled. Entry to and exit from command-line mode must be audited as explained in subsection 4.3.2.3 of the *I&RTS*.

4.2.1.10 User Account Lockout

The DII COE kernel includes the Pluggable Authentication Module (PAM) Strike Manager (PSM) facility to manage locking and unlocking user accounts. On Solaris, PSM takes advantage of PAM capabilities provided by Solaris; on HP-UX, PSM uses the native operating system facilities. PSM requires that a user log into a local workstation within a certain number of attempts before being locked out of the system. This facility is disabled by default. Refer to the Introduction to the PAM Strike Manager (PSM) section in the *Security Administrator's Manual (SECAM) for Kernel Version 4.1.2.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)* for specific details to enable and configure this facility.

4.2.1.11 Deadman Capability

The DII COE kernel includes the Deadman feature, which provides additional security for a user's login desktop session. After a specified period of inactivity, the user's session is terminated, preventing it from being accessed by other users. Deadman works by configuring CDE to invoke `COE_deadman` instead of `dtscreen` when the screen saver or screen lock is to be turned on by `dtsession`. The user is required to login again to regain access to the desktop session. This feature is disabled by default. Refer to the Introduction to Deadman section in the *Security Administrator's Manual (SECAM) for Kernel Version 4.1.2.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)* for specific details to enable and configure this feature.

4.2.2 NT Platform

The security architecture baseline for the NT platform is implemented during kernel installation, and through the subsequent application of security templates via Microsoft's Security Configuration Editor. The subsections below detail the security settings implemented during kernel installation; details of the security settings implemented by the security templates can be found in Appendices B and C.

4.2.2.1 NT Lockdown Scripts

The NT lockdown scripts are implemented as part of the kernel installation. The file names and a description of each are provided below:

Table 4.2.2.1-1. NT Lockdown Scripts

Script File Name	Description
Setup.rul	InstallShield script providing installation template instructions. This includes COE Kernel installation instructions and security configuration function calls. The script includes function calls "C2Config()", "SecurePorts", "File ACLS()", "AuditPolicy()", "RegistrySecurity()", "AccountPolicy()", "WinExit", and "APMSecurity".
<i>Regacl.ini</i>	Called by the <i>setup.rul</i> script template to modify the NT registry.
<i>Setacls.cmd</i>	Called by the <i>setup.rul</i> script to configure directory and file

	acls.
<i>Banner.ini</i>	Called by the <i>setup.rul</i> script to configure the NT registry entries for the DISA warning notification.
<i>Pass.cmd</i>	Called by the <i>setup.rul</i> script, the <i>pass.cmd</i> executes APM <i>coepromptpasswd.exe</i> command.
Winreg.ini	Called by <i>setup.rul</i> script to configure the NT registry screen lock capability (not enabled by default).
Winexit.ini	Called by <i>setup.rul</i> script to configure the NT registry capability for the screen saver and deadman setting. The <i>winexit.ini</i> file specifies the screen saver option as <i>winexit.scr</i> also providing the deadman capability (not enabled by default).

4.2.2.2 C2 Configuration Settings

The NT OS provides a number of security-related configuration settings related to the C2 level of security. The following six settings have been configured by default:

1. Portable Operating System Interface for UNIX (POSIX) and OS/2 subsystems disabled (done in registry and at file/directory level);
2. Security log settings configured to “Do Not Overwrite Events (Clear Log Manually)”;
3. DoD warning banner added (accomplished in registry);
4. Last username display disabled (accomplished in registry);
5. Guest account disabled; and
6. Floppy drive and Compact Disk-Read Only Memory (CD-ROM) drive allocated at logon (accomplished in registry).

The following C2 required registry ACLs are set for both servers and workstations (except where otherwise noted):

Table 4.2.2.2-1. C2 Registry ACLs

<p>To disable the OS/2: HKLM\Software\Microsoft\Microsoft\OS/2 Subsystem for NT] [HKLM\System\CurrentControlSet\Control\SessionManager\Environment\Os2LibPath] [HKLM\System\CurrentControlSet\Control\SessionManager\SubSystem\Os2]</p> <p>To disable the POSIX subsystem, the following key will be removed: [HKLM\System\CurrentControlSet\Control\SessionManager\SubSystem\Posix]</p> <p>The following directory and files should be removed: [%systemroot%\SYSTEM32]\os2.exe [%systemroot%\SYSTEM32]\os2ss.exe [%systemroot%\SYSTEM32]\os2srv.exe [%systemroot%\SYSTEM32]\psxss.exe [%systemroot%\SYSTEM32]\posix.exe [%systemroot%\SYSTEM32]\psxdll.dll [%systemroot%\SYSTEM32]\os2 directory [%systemroot%\SYSTEM32]\dos directory</p>	<p>To disable the OS/2 and POSIX subsystem, the listed keys, directories and files will be removed.</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] LegalNoticeCaption (REG_SZ) = <caption message text></p> <p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] LegalNoticeText (REG_SZ) = <message of text></p>	<p>Add legal notice (DoD warning banner)</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] DontDisplayLastUserName (REG_SZ) = 1</p>	<p>Disable last username display</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] AllocateFloppies (REG_SZ) = 1 Allocate</p>	<p>Allocate floppy drive at logon.</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] AllocateCDRoms (REG_SZ) = 1</p>	<p>Allocate CD-ROM drive at logon.</p>

4.2.2.3 Account Policy

The account policy has been set in accordance with Table 4.3.2-1.

Table 4.2.2.3-1. Account Policy

Policy	Value
Maximum password age	91 days
Minimum password age	7 days
Minimum password length	8
Password uniqueness	0 passwords
Use of PASSFILT filter	N/A

4.2.2.4 File System ACLs

The following file system ACLs are set for both servers and workstations:

Table 4.2.2.4-1. File System ACLs

[%systemdrive%] (directory and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read
[%systemdrive%\TEMP] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change
[%systemdrive%\Program Files] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change
[%systemroot%] (directory only) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change
[%systemroot%] (subdirectories and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read

<p>[%systemroot%\REPAIR] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control</p>
<p>[%systemroot%\Security] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control</p>
<p>[%systemroot%\Cookies] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change</p>
<p>[%systemroot%\Help] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change</p>
<p>[%systemroot%\History] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change</p>
<p>[%systemroot%\Temporary Internet Files] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change</p>
<p>[%systemroot%\SYSTEM32]] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control Authenticated Users = Read</p>
<p>The following files under %systemroot%\SYSTEM32 have these permissions:</p> <p>Administrators = Full Control SYSTEM = Full Control</p> <p>At.exe</p>

Backup.exe Cacls.exe ftp.exe ipconfig.exe nbtstat.exe net.exe netstat.exe nslookup.exe ntbackup.exe ping.exe rcp.exe rdisk.exe regedit.exe (under %systemroot%) regedt32.exe rsh.exe syskey.exe telnet.exe
[%systemroot%\SYSTEM32\CONFIG] (directory and files) Administrators = Full Control SYSTEM = Full Control
[%systemroot%\SYSTEM32\Repl\Export] (directory, subdirectories, and files) Administrators = Full Control Authenticated Users = Read SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Read
[%systemroot%\SYSTEM32\Repl\Import] (directory, subdirectories, and files) Administrators = Full Control Authenticated Users = Read SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Change
[%systemroot%\SYSTEM32\Spool\Printers] (directory and files) Administrators = Full Control Authenticated Users = Change SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Change
[%systemdrive%\h] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read

<p>[%systemdrive%\h\COE\data]] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change (Kernel currently sets to Read)</p>
<p>[%systemdrive%\h\COE\data\db] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Admin = Full Control Authenticated Users = Change</p>
<p>[%systemdrive%\h\COE\Comp\APM\lib\APM.jar] (file only) Administrators = Full Control SYSTEM = Full Control Admin = Read</p>
<p>[%systemdrive%\h\COE\Comp\APM\bin\APM.dll] (file only) Administrators = Full Control SYSTEM = Full Control Admin = Read</p>

4.2.2.5 Registry ACLs

The following registry ACLs are set for both servers and workstations (except where otherwise noted):

Table 4.2.2.5-1. Registry Key ACLs

<p>[HKLM\Software\Microsoft\Windows\CurrentVersion]</p> <p>For the Run, RunOnce, Uninstall subkeys:</p> <p>Everyone = READ</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion]</p> <p>Changed the permissions on the Everyone group for the following subkeys to:</p> <p>Everyone = READ</p> <p>AeDebug Compatibility Drivers32 Embedding Fonts FontSubstitutes FontDrivers FontMapper FontCache GRE_Initialize MCI MCI Extensions ProfileList Type 1 Installer Ports WOW Winlogon</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\PerfLib]</p> <p>Removed the Everyone group, replaced with:</p> <p>INTERACTIVE = READ</p>
<p>[HKLM\Software\Windows 3.1 Migration Status]</p> <p>Everyone = READ</p>
<p>\Registry\Machine\Software\Windows 3.1 Migration Status\IniFiles</p> <p>Everyone = READ</p>
<p>[HKLM\System\CurrentControlSet\Services\LanmanServer\Shares]</p>

Everyone = READ
[HKLM\System\CurrentControlSet\Control\SecurePipeServers\winreg] Removed all groups except: Administrators = Full Control
[HKLM\System\CurrentControlSet\Services\LanmanWorkstation] Everyone = READ

4.2.2.6 Registry Security Settings

The following registry security settings are set for both servers and workstations:

Table 4.2.2.6-1. Registry Settings

[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] CachedLogonsCount = 0
[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] DeleteRoamingCache = 1
[HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters] RestrictNullSessAccess = TRUE
[HKLM\System\CurrentControlSet\Services\EventLog] For Application, Security, System, the following value is set: RestrictGuestAccess = 1
[HKLM\System\CurrentControlSet\Control\Lsa] Submit Control = 1
[HKLM\System\CurrentControlSet\Control\Lsa] FullPrivilegeAuditing = 1
[HKLM\System\CurrentControlSet\Control\Lsa] CrashOnAuditFail = 0
[HKLM\System\CurrentControlSet\Control\Lsa] RestrictAnonymous = 1
[HKLM\System\CurrentControlSet\Control\

Session Manager\Memory Management] ClearPageFileAtShutdown = 1
[HKLM\Software\Microsoft\Windows NT\ CurrentVersion\Winlogon] PasswordExpiryWarning = 7

4.2.2.7 Audit Policy

The audit policy has been set as shown in Table 4.3.6-1.

Table 4.2.2.7-1. Audit Policy

Policy	Value
Logon and logoff	Success and Failure
File and object access	Failure
User rights	Success and Failure
User and group management	Success and Failure
Security policy changes	Success and Failure
Restart and shutdown	Success and Failure
Process tracking	Failure
Restart, Shutdown and System	Success and Failure
Log sizes	20MB

4.2.2.8 User Rights

User rights have been implemented through the use of security templates. Details are provided in Appendix C.

4.2.2.9 User Profiles

User profiles have not been implemented in COE V4.x

4.2.2.9 System Policies

System policies (i.e., screen savers) have been implemented through the use of security templates. Details are provided in Appendix C.

4.2.2.10 User Account Lockout

The Windows NT user account strike feature enables locking and unlocking of user accounts on NT machines. When strike management is configured, an account is locked when a user logs in with a specified number of incorrect password attempts (three by default). This can occur from the desktop login screen or the desktop screen saver. This feature is disabled by default. Refer to the Windows NT Strike Management section in the *Security Administrator's Manual (SECAM) for Kernel Version 4.1.1.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)* for specific details to enable and configure this feature.

4.2.2.11 Deadman Capability

The DII COE kernel on Windows NT includes the Deadman feature, which provides additional security for a user's login desktop session. After a specified period of inactivity, the user's session is terminated, preventing it from being accessed by other users. The user is required to login again to regain access to the desktop session. This feature is disabled by default. Refer to the Deadman on Windows NT section in the *Security Administrator's Manual (SECAM) for Kernel Version 4.1.2.0 (HP-UX 10.20/Solaris 7/Windows NT 4.0)* for specific details to enable and configure this feature.

5. SECURITY COMPLIANCE

5.1 UNIX Platforms (Solaris and HP-UX)

To ensure that a segment complies with the security requirements identified in the *Integration and Runtime Specification (I&RTS) V4.0* document, the *VerifySeg* security module tool shall be used on all 4.x or later segments. The *VerifySeg* security module tool is initially being provided as a stand-alone beta program available from the DII COE Security Engineering Web site; an integrated *VerifySeg* tool with the security module will be provided in the first quarter of 2000.

5.1.1 Compliance Process

The *VerifySeg* security module has been developed to analyze security compliance, with a particular emphasis on restricting file and directory permissions. To accomplish this, the *VerifySeg* security module includes both a DII COE Chief Engineer-determined set of minimal compliance requirements and an optional System Integrator-determined set of desired compliance requirements, the intent of the latter being that a system integrator may want to establish more stringent security checks. (These two sets of requirements are for file/directory permissions only and not for the other security compliance checks defined in the following subsections.)

```
SegmentHomeDir:750:warn
"Scripts":750:warn:750:warn
"SegDescrip":750:warn:750:warn
"bin":750:warn:750:warn
"lib":750:warn:750:warn
"data":770:warn:660:warn
"DBS_files":770:warn:660:warn
"install":750:warn:750:warn
"man":750:warn:440:warn
Other:750:warn:750:warn
umask:027:warn
```

Figure 5.1.1-1. Sample DII COE Chief Engineer Compliance Requirements Table

Each row of the table consists of colon-delimited parameters. With two exceptions (*umask* and *SegmentHomeDir*), the first parameter represents a directory name, the second parameter is the minimum required directory permissions, and the third is either "warn" or "fail" to indicate how

the *VerifySeg* security module is to handle segment directories that do not comply with the requirement. Literal directory names are enclosed in double quotes while *SegmentHomeDir* means the segment's home directory and *Other* means the default setting for any directory not otherwise specified. The directory parameter specification applies to all subdirectories underneath the specified directory. In the example given in Figure 5.1.1-1, any subdirectories underneath the segment's "data" directory should also have 770 (or more restrictive) permissions. The *umask* entry indicates what the acceptable UNIX *umask* setting is to be. Segments are not allowed to set the system *umask* setting; the COE itself establishes it for the system.

Following the directory specifications are two parameters that indicate the permissions for files within the directory and how to handle violations ("warn" or "fail"). In the example given, files underneath the segment's "data" directory are to have permissions 660 whether they are directly underneath the data directory, or in a subdirectory underneath the data directory. *SegmentHomeDir* does not allow file permissions to be described because the *I&RTS* does not allow any files directly underneath the segment's home directory. The file permission specified by the files portion of *Other* apply to any file contained in any directory not otherwise covered.

Compliance determination does not mean an exact match on permissions. A segment may have more restrictive settings but may not have less restrictive settings. For example, a segment with permissions 550 on the "data" subdirectory passes the compliance check because it is more restrictive in its permissions for the directory owner. However, a permissions setting of 555 would fail because it is less restrictive for "world" users, even though it is more restrictive for the directory owner.

The *VerifySeg* security module shall examine a segment for security compliance in two passes. In the first pass, the tool uses the DII COE Chief Engineer Compliance Requirements Table (illustrated in Figure 5.1.1-1) to examine a segment and shall generate warnings as stipulated by the Compliance Requirements Table. Upon completion of the first pass, if the optional System Integrator Compliance Requirements Table is provided, the tool shall use the table to re-examine the segment. It shall generate warnings for any areas of non-compliance in the second pass. The *VerifySeg* security module shall clearly indicate that the warnings generated in the second pass are for informational purposes only and are not used to determine a segment's DII COE compliance level.

Security compliance checking is important both before and after a segment is installed because non-compliance may be a side effect of the installation process itself or a result of running the segment. For this reason, the *VerifySeg* security module shall be modified to include a flag that allows the tool to perform the security checks only without performing any other compliance tests. Moreover, when the flag is present, the *VerifySeg* security module shall make no alterations to the segment, such as re-creating the *Installed* file.

5.1.2 Compliance Requirements

This section describes the requirements for specific security checks that the *VerifySeg* security module performs. Two details are important.

1. Analysis of the segment is to include all directories/files underneath the segment's home directory.
2. When evaluating data file/directory requirements, the analysis shall also include files and subdirectories underneath the directories */h/data/local/SegDir* and */h/data/global/SegDir* where *SegDir* is the segment's home directory name.

5.1.2.1 Fundamental Requirements

There are certain requirements that are so crucial to the security posture of a COE-based system that *VerifySeg* shall generate warnings for a segment that does not meet the stated crucial requirements.

- Regardless of requirements established by either Compliance Requirements Table, the tool shall issue a warning if any segment file or directory is found that is world writeable.
- The tool shall issue a warning if the segment contains any SUID/SGID shell script. This particularly includes examination of files in *bin*, *SegDescrip*, and *Scripts*.
- The tool shall issue a warning if the segment contains any C shell scripts.
- The tool shall issue a warning if the segment inserts “.” or “~” into the search path.
- The tool shall issue a warning for any segment other than COTS that fails to contain a *FileAttribs* segment descriptor.

5.1.2.2 Directory and File Permissions

Directory and file permissions shall be evaluated in two passes. The first pass shall use the DII COE Chief Engineer Compliance Requirements Table and issue a warning message as indicated by the table. If the optional Chief Engineer Compliance Requirements Table exists, the second pass shall evaluate the segment against those requirements and issue warnings for violations.

- The tool output shall clearly distinguish between COE compliance requirements (i.e., the DII COE Chief Engineer Compliance Requirements Table) and optional checks requested by the System Integrator.
- The tool shall clearly indicate that violation of the System Integrator requirements is not considered in determining a segment's DII COE compliance level.
- The tool shall indicate a violation if a directory or file permission is found that is less restrictive than that specified by either Compliance Requirements Table.
- Regardless of requirements established by either Compliance Requirements Table, the tool shall issue a warning if any segment file is found that is world executable.

- The tool shall issue a warning if any file/directory has the set userid (SUID) or set groupid (SGID) bit set.

5.1.2.3 Group/Owner Requirements

- The tool shall issue a warning³ if any segment file/directory is found that does not belong to a defined group.
- The tool shall issue a warning if any segment file/directory is found that is not owned.

5.1.2.4 Additional *SegDescrip* Requirements

- The tool shall examine *FileAttribs* to determine if it will cause any violations identified in the preceding subsections.
- The tool shall examine *PostInstall*, *PreInstall*, and *DEINSTALL* to see if they will cause any violations identified in the preceding subsections.

5.1.2.5 Requirements on Shells and Executables

- The tool shall issue a warning⁴ if either *bin*, *Scripts*, or *SegDescrip* uses a shell other than a Bourne or Korn shell.
- The tool shall issue a warning if any executable in *bin*, *Scripts*, or *SegDescrip* uses SGID or SUID.
- The tool shall issue a warning if any script in *bin*, *Scripts*, or *SegDescrip* uses any of the Unix “r” commands.
- The tool shall issue a warning if any script in *bin*, *Scripts*, or *SegDescrip* establishes a *umask* setting.
- The tool shall issue a warning or a failure (as specified by the Compliance Requirements Tables) if any script in *bin*, *Scripts*, or *SegDescrip* establishes a less restrictive *umask* setting.

³ Warnings are stipulated because group and ownership may be established as part of the installation process. It may be difficult to determine if installation ensures group/owner settings.

⁴ The tool unconditionally fails if C shells are used as per a previous subsection. However, there are shells other than Bourne and Korn that may be acceptable (such as the restricted secure shell, *ssh*) while others are unacceptable (such as *dtksh*). Thus the tool issues a warning since it is not feasible to exhaustively list all possible shells that may or may not be acceptable.

5.1.2.6 Installation-Related Requirements

Some checks are easier to perform after installation is accomplished. The following requirements are included when the “segment is installed” flag is specified.

- The tool shall perform all checks identified in previous subsections and issue warnings as applicable.
- The tool shall issue a warning if any file/directory is found that does not have an owner.
- The tool shall issue a warning if any file/directory is found that does not belong to a group.
- The tool shall evaluate the current value of *path* and *LD_LIBRARY_PATH* and issue a warning if either contains “.” Or “~”. The error message should warn the user to check to see if these are inserted by the segment or are part of the environment setting established prior to installing the segment.
- The tool shall evaluate the current *umask* setting and issue a warning if the *umask* is less restrictive than that identified by the Compliance Requirements Tables. The error message shall warn the user to check to see if the *umask* setting is created as a result of installing the segment or is a part of the environment setting established prior to installing the segment.

5.2 NT Platform

When a segment is loaded on an NT system, several new directories, files and registry keys are added to the system. If the permissions on these directories, files and registry keys are not securely configured prior to or as part of segment installation, the security posture of the NT Workstation/Server can be compromised.

5.2.1 Compliance Process

The NT Security Compliance Process (SCP) will be used by the CFI to analyze an NT segment’s security compliance with directory, file and registry access control list (ACL) settings defined in the I&RTS.

The CFI uses the following programs to execute the NT SCP:

- *sysdiff* – a program that provides file and directory structure and registry setting snapshots before and after application installation (provided in the NT Resource Kit);
- *listacl.exe* (part of the NT security third party product line of utilities); and

- *reglistacl.exe* (part of the NT security third party product line of utilities).

The NT SCP is executed as follows:

- The “locked down” COE kernel is installed and configured on an NT platform.
- A listing, or snapshot, of the baseline NT file system file names and registry keys is taken using *sysdiff* with the /snap option.
- The NT segment is installed⁵. The current Service Pack and hotfixes are re-installed.
- The application is launched via its icon as an Authenticated User.
- Another snapshot of the NT file system file names and registry keys is taken using *sysdiff* with the /diff option.
- Compare the snapshots taken before and after segment installation. Identify deltas in the snapshots by using *sysdiff* with the /dump option.
- A pass/fail decision is made for security compliance based upon the I&RTS requirements.

5.2.2 Compliance Requirements

The ability to maintain the security configuration of an NT segment’s directory and file permissions is dependent upon the parent directory’s permissions. The \h directory, and the \h\Coe, \h\Cots, \h\Data and \h\Users subdirectories have the permissions listed below:

Administrators	Full Control
Authenticated Users	Read (RX)(RX) ⁶
System	Full Control

The Everyone group is not assigned any permissions.

The required permissions for an NT segment are shown in Table 5.2.2-1. These permissions enable an Authenticated User to write to the directory and have full control over any file created by that user in the directory; however, the Authenticated User will only be able to read and execute files created by other users.

⁵ The NT SCP is applied only after all segments that comprise an application are installed.

⁶ The first set of permissions shown (RX) are directory permissions, while the second set of permissions are for files (RX).

Seg Dir Names	Dir Permissions		File Permissions	
SegDir Scripts SegDescrip Bin Lib Install Man all other	Administrators Authenticated Users Creator Owner System	Full Control Add & Read (RWX) Full Control Full Control	Administrators Authenticated Users Creator Owner System	Full Control Read (RX) Full Control Full Control
Data DBS_files	Administrators Authenticated Users Creator Owner System	Full Control Change (RWXD) ⁷ Full Control Full Control	Administrators Authenticated Users Creator Owner System	Full Control Change (RWXD) ⁸ Full Control Full Control

Table 5.2.2-1: NT Segment Directory and File Permissions

For data directories within a segment or shared directories such as \h\data\local or global, Authenticated Users may require Change (RWXD) permissions. Change permissions for directories will be required to add and delete any files in the directory. Segments that require Authenticated Users to have Change permissions must document this requirement in the SVD.

In no instance will an Authenticated User be granted Full Control over a file/directory. Full Control only offers two permissions more than Change (RWXD). Those permissions are take ownership and change permissions. An Authenticated User should not have to perform either of these operations for a segment to function.

After installing a NT segment, the CFI will perform a recursive search to identify all files of the following types:

*.exe, *.com, *.bat, *.cmd, *.dll, *.inf, *.hlp

⁷ Directory Change permissions for Authenticated Users enable them to delete any files in the directory (including existing files). If Authenticated Users do not need to delete any files in this directory, directory permissions should be Add and Read (RWX)(RX).

⁸ File Change permissions for Authenticated Users enable them to delete files added to the directory. If Authenticated Users do not need to delete files that have been added to the directory, file permissions should be Read (RX).

These files shall have the following permissions to meet compliance:

Administrators	Full Control
Authenticated Users	Read (RX)(RX)
System	Full Control

For registry keys, all keys below HKEY_LOCAL_MACHINE\Software shall have the following permissions:

Administrators	Full Control
Creator Owner	Full Control
System	Full Control
Authenticated Users	Special Access ⁹

The inherited registry permissions on a system with its registry secured in accordance with the I&RTS should be sufficient for the application to function.

⁹ The least restrictive special access permissions that Authenticated Users are to be granted are as follows: Query Value, Set Value, Create Subkey, Enumerate Subkeys, Notify, Delete, and Read Control

This page intentionally left blank.

APPENDIX A
ACRONYMS/ABBREVIATIONS

This page intentionally left blank.

A. ACRONYMS/ABBREVIATIONS

ACL	Access Control Lists
AD	Administrative Domain
AKDSG	<i>Application and Kernel Developers' Security Guidance</i>
API	Application Programming Interface
APM	Account and Profile Manager
C&A	Certification and Accreditation
C/S/A	CINCs/Services/Agencies
CD-ROM	Compact Disk-Read Only Memory
CDS	Common Data Store
CFI	Center for Integration
CINC	Commander-in-chief
COE	Common Operation Environment
CONOPS	Concept of Operations
COTS	Commercial off-the-shelf
DAC	Discretionary Access Control
DII	Defense Information Infrastructure
DoD	Department of Defense
FDD	Function Design Document
GCCS	Global Command and Control System
GOTS	Government off-the-shelf
GUI	Graphical User Interface
HP	Hewlett Packard
I&A	Identification and Authentication
IP	Internet Protocol
KDSG	<i>Kernel Developer's Security Guidance</i>
LAN	Local Area Network
LSA	Local Security Authority
MLS	Multi-Level Security
MSL	Multiple Security Levels
NIS+	Network Information Service Plus
NSS	Netscape Security Services
NT	New Technology
OS	Operating System

PGRM	<i>Programmer's Guide and Reference Manual</i>
PKI	Public Key Infrastructure
POSIX	Portable Operating System Interface for UNIX
SAM	Security Account Manager
SECAM	<i>Security Administrator's Manual</i>
SFDG	Security Features Developers Guide
SMB	Server Message Block
SPI	Security Profile Inspector
SRS	Software Requirements Specifications
SS	Security Services
SSAF	Security Services Architecture Framework
SSL	Secure Sockets Layer
SVD	Software Version Description
TCB	Trusted Computing Base
TCP/IP	Transmission Control Protocol/Internet Protocol
UX	UNIX
V	Version

APPENDIX B
NT BASELINE SECURITY CONFIGURATION

This page intentionally left blank.

B. NT BASELINE SECURITY CONFIGURATION

The baseline security configuration for the NT platform requires the application of security templates (*.inf files) after kernel installation using Microsoft's Security Configuration Editor. The order of segments applied after kernel installation is depicted in Figure B.-1.

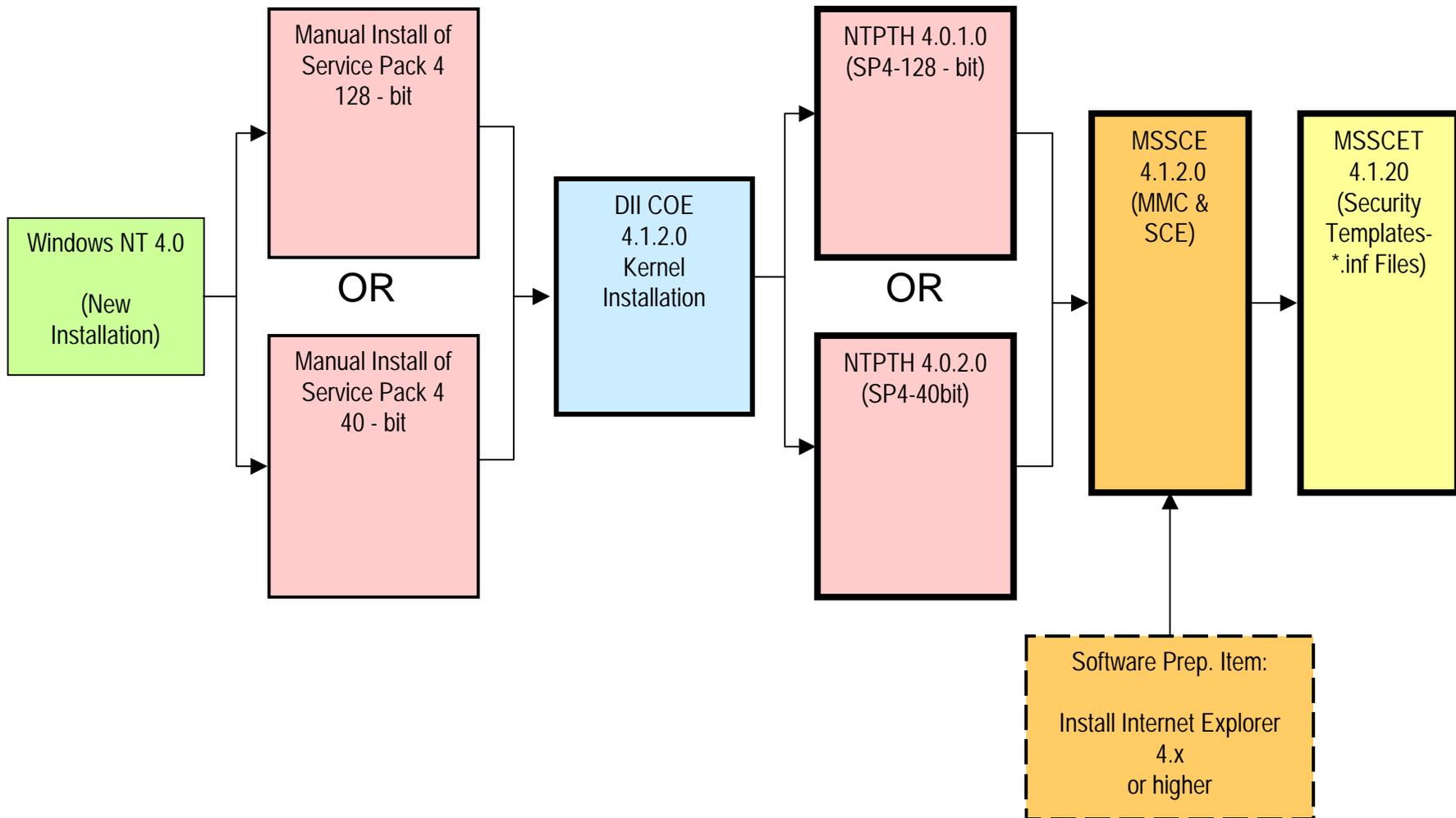


Figure B.-1 Installation Sequence for DII COE 4.1.2.0 for Windows NT

Security templates are provided for both workstation and domain controller installations (primary or back-up). Unlock templates are also provided that restore the kernel to the minimum security configuration that still enables operation.

APPENDIX C
NT WORKSTATION SECURITY TEMPLATES

This page intentionally left blank.

C. NT WORKSTATION SECURITY TEMPLATES

The following table compares the security settings contained in the DII COE 4.1.2.0 NT kernel to the Security Configuration Editor (SCE) security templates for the NT workstation. Two types of templates are provided: one for “locking” security settings and one for “unlocking” security settings.

In general, the security templates contain many of the same settings as the NT kernel: C2 configuration, file system and registry ACLs, account policies, audit policies. Settings contained in the templates that are not configured by default in the 4.1.2.0 kernel include: user rights and additional registry security keys (implemented through *.reg files).

Blank fields in the following table indicate specific settings that are not currently addressed by the kernel or security templates.

NOTE: Security templates for standalone servers (e.g., print servers, file servers) and domain controllers (e.g., PDCs, BDCs) have also been developed; details will be provided as a change page to this document.

Table C.-1 NT COE 4.1.2.0 Kernel and Security Template Settings

NT COE Kernel Security Settings (Workstation)	COE_WKSTN Security Template Settings	UNLOCK_COE_WKSTN Security Template Settings
C2 Configuration		
POSIX/OS2 subsystems disabled	POSIX/OS2 subsystems disabled	POSIX/OS2 subsystems disabled
Configure event logs to “Do Not Overwrite Event (Clear Log Manually)”	Configure event logs to “Do Not Overwrite Event (Clear Log Manually)”	Configure event logs to “Do Not Overwrite Event (Clear Log Manually)”
Add legal notice (DoD warning banner)	Add legal notice (DoD warning banner)	Add legal notice (DoD warning banner)
Disable last username display	Disable last username display	Disable last username display
Disable Guest account	Disable Guest account	Disable Guest account
Allocate floppy and CDROM drives at logon	Allocate floppy and CDROM drives at logon	Allocate floppy and CDROM drives at logon

File System ACLs		
[%systemdrive%] (directory and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read	[%systemdrive%] (directory and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = READ	[%systemdrive%] (directory and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = READ Users = READ
[%systemdrive%\TEMP] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	[%systemdrive%\TEMP] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read&Add	[%systemdrive%\TEMP] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Everyone = Read/Write
[%systemdrive%\Program Files] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	[%systemdrive%\Program Files] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	
[%systemroot%] (directory only) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	[%systemroot%] (directory level only) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read&Add	[%systemroot%] (directory level only) Administrators = Full Control SYSTEM = Full Control Everyone = Change
[%systemroot%] (subdirectories and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read	[%systemroot%] (subdirectories and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read	

[%systemroot%\REPAIR] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control	[%systemroot%\REPAIR] (directory and files) Administrators = Full Control SYSTEM = Full Control	
[%systemroot%\Security] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control	[%systemroot%\Security] (directory and files) Administrators = Full Control SYSTEM = Full Control	
[%systemroot%\Cookies] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	[%systemroot%\Cookies] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	
[%systemroot%\Help] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	[%systemroot%\Help] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read&Add	
[%systemroot%\History] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change	[%systemroot%\History] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read&Add	
[%systemroot%\Temporary Internet Files] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control	[%systemroot%\Temporary Internet Files] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control	

<p>Authenticated Users = Change</p> <p>[%systemroot%\SYSTEM32]] (directory, subdirectories, and files)</p> <p>Administrators = Full Control</p> <p>SYSTEM = Full Control</p> <p>Authenticated Users = Read</p>	<p>Authenticated Users = Read&add</p> <p>[%systemroot%\SYSTEM32]] (directory, subdirectories, and files)</p> <p>Administrators = Full Control</p> <p>SYSTEM = Full Control</p> <p>Authenticated Users = Read</p>	<p>[%systemroot%\SYSTEM32]] (directory, subdirectories, and files)</p> <p>Administrators = Full Control</p> <p>SYSTEM = Full Control</p> <p>Everyone = Change</p>
<p>The following files under %systemroot%\SYSTEM32 have these permissions:</p> <p>Administrators = Full Control</p> <p>SYSTEM = Full Control</p> <p>At.exe</p> <p>Backup.exe</p> <p>Cacls.exe</p> <p>ftp.exe</p> <p>ipconfig.exe</p> <p>nbtstat.exe</p> <p>net.exe</p> <p>netstat.exe</p> <p>nslookup.exe</p> <p>ntbackup.exe</p> <p>ping.exe</p> <p>rcp.exe</p> <p>rdisk.exe</p> <p>regedit.exe (under %systemroot%)</p> <p>regedt32.exe</p> <p>rsh.exe</p> <p>syskey.exe</p>	<p>The following files under %systemroot%\SYSTEM32 have these permissions:</p> <p>Administrators = Full Control</p> <p>SYSTEM = Full Control</p> <p>At.exe</p> <p>Backup.exe</p> <p>Cacls.exe</p> <p>ftp.exe</p> <p>ipconfig.exe</p> <p>nbtstat.exe</p> <p>net.exe</p> <p>netstat.exe</p> <p>nslookup.exe</p> <p>ntbackup.exe</p> <p>ping.exe</p> <p>rcp.exe</p> <p>rdisk.exe</p> <p>regedit.exe (under %systemroot%)</p> <p>regedt32.exe</p> <p>rsh.exe</p> <p>syskey.exe</p>	<p>The following files under %systemroot%\SYSTEM32 have these permissions:</p> <p>Administrators = Full Control</p> <p>SYSTEM = Full Control</p> <p>Users = READ</p> <p>At.exe</p> <p>Cacls.exe</p> <p>ftp.exe</p> <p>ipconfig.exe</p> <p>nbtstat.exe</p> <p>net.exe</p> <p>netstat.exe</p> <p>nslookup.exe</p> <p>ntbackup.exe</p> <p>ping.exe</p> <p>rcp.exe</p> <p>rdisk.exe</p> <p>rsh.exe</p> <p>telnet.exe</p>

telnet.exe	telnet.exe	
[%systemroot%\SYSTEM32\CONFIG] (directory and files) Administrators = Full Control SYSTEM = Full Control	[%systemroot%\SYSTEM32\CONFIG] (directory and files) Administrators = Full Control SYSTEM = Full Control	[%systemroot%\SYSTEM32\CONFIG] (directory and files) Administrators = Full Control SYSTEM = Full Control
[%systemroot%\SYSTEM32\Repl\Export] (directory, subdirectories, and files) Administrators = Full Control Authenticated Users = Read SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Read	[%systemroot%\SYSTEM32\Repl\Export] (directory and files) Administrators = Full Control Authenticated Users = Read SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Read	
[%systemroot%\SYSTEM32\Repl\Import] (directory, subdirectories, and files) Administrators = Full Control Authenticated Users = Read SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Change	[%systemroot%\SYSTEM32\Repl\Import] (directory, subdirectories, and files) Administrators = Full Control Authenticated Users = Read SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Change	
[%systemroot%\SYSTEM32\Spool\Printers] (directory and files) Administrators = Full Control Authenticated Users = Change SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Change	[%systemroot%\SYSTEM32\Spool\Printers] (directory and files) Administrators = Full Control Authenticated Users = Change SYSTEM = Full Control CREATOR OWNER = Full Control Replicator = Change	
[%systemdrive%\h] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read	[%systemdrive%\h] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read	[%systemdrive%\h] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read

<p>[%systemdrive%\h\COE\data]] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change (Kernel currently sets to Read)</p>	<p>[%systemdrive%\h\COE\data]] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Read & Add</p>	<p>[%systemdrive%\h\COE\data]] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Authenticated Users = Change</p>
<p>[%systemdrive%\h\COE\data\db] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Admin = Full Control Authenticated Users = Change</p>		<p>[%systemdrive%\h\COE\data\db] (directory, subdirectories, and files) Administrators = Full Control SYSTEM = Full Control CREATOR OWNER = Full Control Admin = Full Control Authenticated Users = Change</p>
<p>[%systemdrive%\h\COE\Comp\APM\lib\APM.jar] (file only) Administrators = Full Control SYSTEM = Full Control Admin = Read</p>		<p>[%systemdrive%\h\COE\Comp\APM\lib\APM.jar] (file only) Administrators = Full Control SYSTEM = Full Control Admin = Read</p>
<p>[%systemdrive%\h\COE\Comp\APM\bin\APM.dll] (file only) Administrators = Full Control SYSTEM = Full Control Admin = Read</p>		<p>[%systemdrive%\h\COE\Comp\APM\bin\APM.dll] (file only) Administrators = Full Control SYSTEM = Full Control Admin = Read</p>
Audit Policy		
<p>Enabled auditing: Logon and Logoff (Success and Failure) File and Object Access (Failure) Use of User Rights (Success and Failure) User and Group Mgmt (Success and</p>	<p>Enabled auditing: Logon and Logoff (Success and Failure) File and Object Access (Failure) Use of User Rights (Success and Failure) User and Group Mgmt (Success and Failure)</p>	<p>Enabled auditing: Logon and Logoff (Success and Failure) File and Object Access (Failure) Use of User Rights (Success and Failure) User and Group Mgmt (Success and</p>

Failure) Security Policy Changes (Success and Failure) Process Tracking (Failure) Restart, Shutdown and System (Success and Failure)	Security Policy Changes (Success and Failure) Process Tracking (Failure) Restart, Shutdown and System (Success and Failure)	Failure) Security Policy Changes (Success and Failure) Process Tracking (Failure)
Log sizes set to 20MB	Log sizes set to 20MB	Log sizes set to 5056 KB
Registry Security Settings		
[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] CachedLogonsCount = 0	[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] CachedLogonsCount = 0	[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] CachedLogonsCount = 0
[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] DeleteRoamingCache = 1	(Implemented through roamcache.reg file) [HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] DeleteRoamingCache = 1	[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] DeleteRoamingCache disabled
[HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters] RestrictNullSessAccess = TRUE	(Implemented through nullsessaccess.reg file) [HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters] RestrictNullSessAccess = TRUE	[HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters] RestrictNullSessAccess disabled
[HKLM\System\CurrentControlSet\Services\EventLog] For Application, Security, System, the following value is set: RestrictGuestAccess = 1	[HKLM\System\CurrentControlSet\Services\EventLog] For Application, Security, System, the following value is set: RestrictGuestAccess = 1	[HKLM\System\CurrentControlSet\Services\EventLog] For Application, Security, System, the following value is set: RestrictGuestAccess = 1
[HKLM\System\CurrentControlSet\Control\Lsa] Submit Control = 1	[HKLM\System\CurrentControlSet\Control\Lsa] Submit Control = 1	
[HKLM\System\CurrentControlSet\Control\	[HKLM\System\CurrentControlSet\Control\L	

Lsa] FullPrivilegeAuditing = 1	sa] FullPrivilegeAuditing = 1	
[HKLM\System\CurrentControlSet\Control\Lsa] CrashOnAuditFail = 0	[HKLM\System\CurrentControlSet\Control\Lsa] CrashOnAuditFail = 0	
[HKLM\System\CurrentControlSet\Control\Lsa] RestrictAnonymous = 1	[HKLM\System\CurrentControlSet\Control\Lsa] RestrictAnonymous = 1	[HKLM\System\CurrentControlSet\Control\Lsa] RestrictAnonymous = 1
[HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management] ClearPageFileAtShutdown = 1	[HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management] ClearPageFileAtShutdown = 1	[HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management] ClearPageFileAtShutdown = 1
[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] PasswordExpiryWarning = 7	[HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon] PasswordExpiryWarning = 7	
Registry ACLs (In this section, only the permissions for the Everyone group are modified)		
[HKLM\Software\Microsoft\Windows\CurrentVersion] For the Run, RunOnce, Uninstall subkeys: Everyone = READ	[HKLM\Software\Microsoft\Windows\CurrentVersion] For the Run, RunOnce, Uninstall subkeys: Authenticated Users = READ	[HKLM\Software\Microsoft\Windows\CurrentVersion] For the Run, RunOnce, Uninstall subkeys: Everyone = READ
[HKLM\Software\Microsoft\Windows NT\CurrentVersion] Changed the permissions on the Everyone group for the following subkeys to: Everyone = READ AeDebug	[HKLM\Software\Microsoft\Windows NT\CurrentVersion] Changed the permissions on the Everyone group for the following subkeys to: Authenticated Users = READ AeDebug	[HKLM\Software\Microsoft\Windows NT\CurrentVersion] Changed the permissions on the Everyone group for the following subkeys to: Everyone = READ AeDebug

<p>Compatibility Drivers32 Embedding Fonts FontSubstitutes FontDrivers FontMapper FontCache GRE_Initialize MCI MCI Extensions ProfileList Type 1 Installer Ports WOW Winlogon</p>	<p>Compatibility Embedding Fonts FontSubstitutes GRE_Initialize MCI MCI Extensions ProfileList Ports WOW</p>	<p>Compatibility Embedding Fonts FontSubstitutes GRE_Initialize MCI MCI Extensions ProfileList Ports WOW</p>
<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\PerfLib]</p> <p>Removed the Everyone group, replaced with: INTERACTIVE = READ</p>	<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\PerfLib]</p> <p>Removed the Everyone group, replaced with: INTERACTIVE = READ</p>	<p>[HKLM\Software\Microsoft\Windows NT\CurrentVersion\PerfLib]</p> <p>Removed the Everyone group, replaced with: INTERACTIVE = READ</p>
<p>[HKLM\Software\Windows 3.1 Migration Status]</p> <p>Everyone = READ</p>	<p>[HKLM\Software\Windows 3.1 Migration Status]</p> <p>Authenticated Users = READ</p>	<p>[HKLM\Software\Windows 3.1 Migration Status]</p> <p>Everyone = READ</p>
<p>\Registry\Machine\Software\Windows 3.1 Migration Status\IniFiles</p> <p>Everyone = READ</p>	<p>\Registry\Machine\Software\Windows 3.1 Migration Status\IniFiles</p> <p>Authenticated Users = READ</p>	
<p>[HKLM\System\CurrentControlSet\Services\LanmanServer\Shares]</p>	<p>[HKLM\System\CurrentControlSet\Services\LanmanServer\Shares]</p>	<p>[HKLM\System\CurrentControlSet\Services\LanmanServer\Shares]</p>

Everyone = READ	Authenticated Users = READ	Everyone = READ
[HKLM\System\CurrentControlSet\Control\SecurePipeServers\winreg] Removed all groups except: Administrators = Full Control	[HKLM\System\CurrentControlSet\Control\SecurePipeServers\winreg] Removed all groups except: Administrators = Full Control System = Full	
[HKLM\System\CurrentControlSet\Services\LanmanWorkstation] Everyone = READ	[HKLM\System\CurrentControlSet\Services\LanmanWorkstation] Authenticated Users = READ	[HKLM\System\CurrentControlSet\Services\LanmanWorkstation] Everyone = READ
Account Policy		
Maximum password age = 91 days	Maximum password age = 91 days	Maximum password age = 91 days
Minimum password age = 7 days	Minimum password age = 7 days	Minimum password age = 7 days
Minimum password length = 8 characters	Minimum password length = 8 characters	Minimum password length = 8 characters
		Password Uniqueness = 2 passwords
Use of PASSFILT filter	Use of PASSFILT filter	
ScreenSaver		
Use of logon screen saver (password protected, enacts after 15 minutes of inactivity)	Use of logon screen saver (password protected, enacts after 15 minutes of inactivity)	
User Rights		
Access this computer from the network Administrators Everyone	Access this computer from the network Administrators Everyone	Access this computer from the network Administrators Everyone
Act as part of the operating system	Act as part of the operating system	Act as part of the operating system

Administrators Admin	Administrators Admin	Administrators Admin
Back up files and directories	Back up files and directories	Back up files and directories
Administrators Backup Operators	Administrators Backup Operators	Administrators Backup Operators
Bypass traverse checking	Bypass traverse checking	Bypass traverse checking
Authenticated Users	Authenticated Users	Everyone
Change the system time	Change the system time	Change the system time
Administrators	Administrators	Administrators
Create a pagefile	Create a pagefile	Create a pagefile
Administrators	Administrators	Administrators
Debug programs	Debug programs	Debug programs
Force shutdown from a remote system	Force shutdown from a remote system	Force shutdown from a remote system
Administrators	Administrators	Administrators
Increase quotas	Increase quotas	Increase quotas
Administrators Admin	Administrators Admin	Administrators Admin
Increase scheduling priority	Increase scheduling priority	Increase scheduling priority
Administrators	Administrators	Administrators
Load and unload device drivers	Load and unload device drivers	Load and unload device drivers
Administrators	Administrators	Administrators
Log on locally	Log on locally	Log on locally
Not modified by kernel	Administrators	Administrators

	Authenticated Users	Authenticated Users Domain Users Users
Manage auditing and security log Administrators Admin	Manage auditing and security log Administrators Admin	Manage auditing and security log Administrators
Profile single process Administrators	Profile single process Administrators	Profile single process Administrators
Profile system performance Administrators	Profile system performance Administrators	Profile system performance Administrators
Replace a process level token Administrators Admin	Replace a process level token Administrators Admin	Replace a process level token Administrators Admin
Restore files and directories Administrators Backup Operators	Restore files and directories Administrators Backup operators	Restore files and directories Administrators
Shut down the system Administrators Authenticated Users	Shut down the system Administrators Authenticated Users	Shut down the system Administrators Authenticated Users Domain Users Users
Take ownership of files or other objects Administrators	Take ownership of files or other objects Administrators	Take ownership of files or other objects Administrators