



# RED HAT ENTERPRISE LINUX 6
# SECURITY TECHNICAL IMPLEMENTATION GUIDE (STIG) OVERVIEW

Version 1, Release 1

15 May 2013

**Developed by Red Hat, NSA, and DISA for the DoD**

## Trademark Information

# TABLE OF CONTENTS

# LIST OF TABLES

**Page**

# 1. INTRODUCTION

## 1.1 Background

The Red Hat Enterprise Linux 6 (RHEL6) Security Technical Implementation Guide (STIG) is published as a tool to improve the security of Department of Defense (DoD) information systems. The requirements were developed from Federal and DoD consensus, based upon the Operating System Security Requirements Guide (OS SRG).

SRGs are collections of requirements applicable to a given technology area. SRGs represent an intermediate step between Control Correlation Identifiers (CCIs) and STIGs. CCIs represent discrete, measurable, and actionable items sourced from Information Assurance (IA) controls defined in policy, such as those originating in Department of Defense (DoD) Instruction (DoDI) 8500.2 and National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53. STIGs provide product-specific information for validating and attaining compliance with requirements defined in the SRG for the product's technology area. The OS SRG contains general requirements for operating systems; this SRG may be used as a guide for enhancing the security configuration of any operating system.

The consensus content was developed using an open-source project called *SCAP Security Guide*. The project's website is https://fedorahosted.org/scap-security-guide/. Except for differences in formatting to accommodate the DISA STIG publishing process, the content of the RHEL6 STIG should mirror the *SCAP Security Guide* content with only minor divergence as updates from multiple sources work through the consensus process.

The vulnerabilities discussed in this document are applicable to RHEL6 Desktop and Server editions. This document is meant for use in conjunction with the Enclave, Network Infrastructure, Secure Remote Computing, and appropriate application STIGs.

## 1.2 Authority

DoD Directive (DoDD) 8500.1 requires that "all IA and IA-enabled IT products incorporated into DoD information systems shall be configured in accordance with DoD-approved security configuration guidelines" and tasks Defense Information Systems Agency (DISA) to "develop and provide security configuration guidance for IA and IA-enabled IT products in coordination with Director, NSA". This document is provided under the authority of DoDD 8500.1.

Although the use of the principles and guidelines in these STIGs provide an environment that contributes to the security requirements of DoD systems operating at Mission Assurance Categories (MACs) I through III, applicable DoDI 8500.2 IA controls need to be applied to all systems and architectures.

## 1.3 Scope

This document is a requirement for all DoD administered systems and all systems connected to DoD networks. These requirements are designed to assist Security Managers (SMs), Information

Assurance Managers (IAMs), Information Assurance Officers (IAOs), and System Administrators (SAs) with configuring and maintaining security controls. This guidance supports DoD system design, development, implementation, certification, and accreditation efforts.

This RHEL6 STIG Overview document introduces security concepts and terminology used in the OS SRG. This document is not a guide to RHEL6 system administration.

## 1.4 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Code of CAT I, II, or III.

**Table 1-1: Vulnerability Severity Category Code Definitions**

| | DISA Category Code Guidelines | Examples of DISA Category Code Guidelines |
|---|---|---|
| CAT I | Any vulnerability, the exploitation of which will directly and immediately result in loss of Confidentiality, Availability, or Integrity. | Includes **BUT NOT LIMITED** to the following examples of direct and immediate loss:<br><br>1. May result in loss of life, loss of facilities, or equipment, which would result in mission failure.<br>2. Allows unauthorized access to security or administrator level resources or privileges.<br>3. Allows unauthorized disclosure of, or access to, classified data or materials.<br>4. Allows unauthorized access to classified facilities.<br>5. Allows denial of service or denial of access, which will result in mission failure.<br>6. Prevents auditing or monitoring of cyber or physical environments.<br>7. Operation of a system/capability which has not been approved by the appropriate Designated Accrediting Authority (DAA).<br>8. Unsupported software where there is no documented acceptance of DAA risk. |

| | DISA Category Code Guidelines | Examples of DISA Category Code Guidelines |
|---|---|---|
| CAT II | Any vulnerability, the exploitation of which has a potential to result in loss of Confidentiality, Availability, or Integrity. | Includes **BUT NOT LIMITED** to the following examples that have a potential to result in loss:<br><br>1. Allows access to information that could lead to a CAT I vulnerability.<br>2. Could result in personal injury, damage to facilities, or equipment which would degrade the mission.<br>3. Allows unauthorized access to user or application level system resources.<br>4. Could result in the loss or compromise of sensitive information.<br>5. Allows unauthorized access to Government or Contractor owned or leased facilities.<br>6. May result in the disruption of system or network resources degrading the ability to perform the mission.<br>7. Prevents a timely recovery from an attack or system outage.<br>8. Provides unauthorized disclosure of or access to unclassified sensitive, Personally Identifiable Information (PII), or other data or materials. |

| | DISA Category Code Guidelines | Examples of DISA Category Code Guidelines |
|---|---|---|
| CAT III | Any vulnerability, the existence of which degrades measures to protect against loss of Confidentiality, Availability, or Integrity. | Includes **BUT NOT LIMITED** to the following examples that provide information which could potentially result in degradation of system information assurance measures or loss of data:<br><br>1. Allows access to information that could lead to a CAT II vulnerability.<br>2. Has the potential to affect the accuracy or reliability of data pertaining to personnel, resources, operations, or other sensitive information.<br>3. Allows the running of any applications, services or protocols that do not support mission functions.<br>4. Degrades a defense in depth systems security architecture.<br>5. Degrades the timely recovery from an attack or system outage.<br>6. Indicates inadequate security administration.<br>7. System not documented in the site's C&A Package/System Security Plan (SSP).<br>8. Lack of document retention by the Information Assurance Manager (IAM) (i.e., completed user agreement forms). |

## 1.5 SRG Compliance Reporting

All technical NIST SP 800-53 requirements were considered while developing this STIG.

Requirements that are applicable and configurable are included in this STIG. A report marked For Official Use Only (FOUO) is available for those items that did not meet requirements. This report is available to component DAA personnel for risk assessment purposes by request via email to disa.letterkenny.FSO.mbx.stig-customer-support-mailbox@mail.mil.

## 1.6 STIG Distribution

Parties within the DoD and Federal Government's computing environments can obtain the applicable SRGs and STIGs from the Information Assurance Support Environment (IASE) website. This site contains the latest copies of any SRG, as well as STIGs, scripts, and other related security information. The Non-classified Internet Protocol Router Network (NIPRNet) Uniform Resource Locator (URL) for the IASE website is http://iase.disa.mil/.

## 1.7 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: disa.letterkenny.FSO.mbx.stig-customer-support-mailbox@mail.mil. DISA Field Security Operations (FSO) will coordinate all change requests with the relevant DoD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA FSO maintenance release schedule.

# 2. ASSESSMENT CONSIDERATIONS

## 2.1 Command Examples

Some check and fix procedures contain example commands that can be used to obtain information regarding compliance with a requirement or to change a setting to attain compliance with a requirement. These example commands assume use of a standard UNIX shell operating as the root user. If the software used by these commands is not present on the system, the SA or the reviewer is responsible for determining compliance with the requirement using the tools available on the system. Check procedures also contain instructions for evaluating compliance based on the output of these commands.

## 2.2 Alternate Software

RHEL6 systems offer extreme flexibility in replacing components provided by Red Hat with other software to meet operational needs. Many of the check and fix procedures in the RHEL6 STIG assume the use of the software provided by Red Hat. If alternate software is used to provide a function ordinarily provided by a default system application, the specific check and fix information for that function is no longer valid. The SA or the reviewer is responsible for evaluating the requirements based on documentation available for the alternate software. The system accreditation package must contain information pertaining to the use of alternate software.

## 2.3 Requirements for Disabled Functions

The RHEL6 STIG defines requirements for the further hardening and configuration of system functions that are required to be disabled. These requirements exist to address vulnerabilities in the system resulting from accidental activation, malicious intentional activation, or intentional activation of the system function based on acceptance of risk by the Designated Accrediting Authority (DAA). Requirements for a system function remain applicable even when the system function is disabled. Requirements pertaining to software that is not installed on the system, and which has no remaining configuration files on the system, may be evaluated as NA.

# 3. CONCEPTS AND TERMINOLOGY CONVENTIONS

The RHEL6 STIG assumes familiarity with some common UNIX concepts and terminology. Some of these concepts and terms are defined and explained in this document in order to facilitate uniform interpretation of the requirements.

## 3.1 UNIX File Permissions

Files are assigned access permissions with standard UNIX access modes or additionally with access control lists (ACLs). The sometimes cumbersome and restrictive nature of the standard UNIX file access modes is not always suitable for certain environments. ACLs provide a greater degree of file access control and a more granular level of file protection, allowing certain privileges to either specific users or specific groups of users. This granular level of file protection provides more flexibility for ensuring file and directory access restrictions.

### 3.1.1 File Ownership

UNIX files have two ownership attributes representing the user and group owners. The user owner of a file retains access to a file regardless of access control settings. Generally, user owners are not permitted to change the user ownership of a file, but they can change the group owner of files they own.

### 3.1.2 Access Modes

A UNIX access mode represents the standard and special access modes associated with a file. Access modes can be represented as a 3- or 4-digit octal number, a symbolic string as produced by **ls –l**, or as a symbolic combination of classes and permissions as used by **chmod(1)**. Table 3-1 provides some examples of different representations of access modes.

**Table 3-1: Access Mode Examples**

| Permissions | Octal Access Mode | `ls -l` symbolic | `chmod` symbolic |
|---|---|---|---|
| User read, write, execute | 700 or 0700 | `-rwx------` | u+rwx |
| User, group, and other have read, write, execute | 777 or 0777 | `-rwxrwxrwx` | a+rwx |
| SetUID; User full access; Group and other read and execute | 4755 | `-rwsr-xr-x` | u+rwxs,go+rx |

#### 3.1.2.1 Standard Access Modes

The standard UNIX file protection model assigns three permissions (read, write, and execute) for each of three classes of users (user, group, and other). Each class may be granted access to a file using any combination of the read, write, and execute permissions.

The “read” permission allows the ability to read a file or list the contents of a directory. The “write” permission allows the ability to modify a file or add or delete a directory entry. The “execute” permission allows the ability to execute a file or traverse a directory.

The “user” class represents only the assigned owner of the file. The “group” class represents any member of the group owning the file. The “other” class, also referred to as “world,” represents any user on the system not covered by the “user” or “group” classes.

#### 3.1.2.2 Special Access Modes

The fourth (left-most) octal digit of an access mode is used to represent three settings used to alter how permissions are processed: setuid, setgid, and the sticky bit.

##### 3.1.2.2.1 Set User ID

The “set user ID” attribute is also commonly known as “setuid” or “suid.” When a file with the setuid attribute is executed, it inherits the privileges of the owner of the file and not the user executing the file. If the owner of the file is root, then the user, while executing that file, has all the powers of root, at least for the scope of the program being executed. Setuid is ordinarily used for system utilities, such as **passwd(1)**, that allow users to modify files, such as /etc/shadow, for which they have no permissions. It is, therefore, extremely important that any file that has the setuid bit set be of known origin and scope.

Refer to the RHEL6 manual pages (“man <setuid program name>”) for details concerning setuid programs. Commercial and government-supplied applications may also contain programs with the setuid bit set. If so, the vendor/proponent instructions must be followed.

##### 3.1.2.2.2 Set Group ID

The “set group ID” attribute is also known as “setgid” or “sgid.” Similar to the setuid attribute, when a file with the setgid attribute is executed, it inherits the privileges of the group owner of the file. It is, therefore, extremely important that any file that has the setgid bit set be of known origin and scope.

The set group ID has another function when applied to a directory. New files that are created in a directory with the setgid bit set are group-owned by the directory’s group, instead of the group of the process creating the file. This can simplify the use of shared directories. The setgid bit conveys no additional privileges when set on a directory.

Refer to the RHEL6 manual pages (“man <setgid program name>”) for details concerning setgid programs. Commercial and government-supplied applications may also supply programs with the setgid bit set. If so, then vendor/proponent instructions must be followed.

#### 3.1.2.2.3 Sticky Bit

Directories that are world-writable (that is, the "other" user class has write permission) can be accessed and changed by any user with access to the system. Users could populate these directories with erroneous, malicious, and harmful files. In addition, users could also delete files belonging to other users contained in these directories. In the event a directory is required to allow all users permission to write to this directory, such as the case of public directories (e.g., /tmp), the sticky bit must be set.

The sticky bit protects the files within this directory by preventing a user from deleting other users' files also located in this public directory. When the sticky bit has been set on a directory, the owner of the file, owner of the directory, or root, may only delete a file. For that reason, world-writable directories will only be allowed if they are public directories and have the sticky bit set.

### 3.1.2.3 Comparing Access Modes

The RHEL6 STIG uses the terms "less permissive" and "more permissive" when comparing the access modes of files to a defined value. An access mode is considered "more permissive" than the defined value if it has any permissions set that the defined value does not. If the mode is not equal to the defined value and is not "more permissive," it is considered "less permissive" than the defined value.

Access modes defined in the RHEL6 STIG are expressed as 4-digit octal numbers. To determine whether a given mode number is more or less permissive than another, the binary representation of the modes can be compared.

For example:
A check that states "The /etc/example file must have mode 0600 or less permissive." If the /etc/example file on the system has a mode of 0466, this is a finding, as a mode of 0466 is more permissive than 0600. Consider the binary representation of the octal mode numbers:

| **Defined Access Mode** | | | | **Actual Access Mode** | | | |
|---|---|---|---|---|---|---|---|
| 0 | 6 | 0 | 0 | 0 | 4 | 6 | 6 |
| 000 | 110 | 000 | 000 | 000 | 100 | 110 | 110 |
| | rw- | --- | --- | | r-- | rw- | rw- |

The file's mode has read and write permissions for the "group" and "other" classes, while the defined mode does not have these permissions.

#### 3.1.2.4 Umask

The umask is a kernel variable that controls the file access permissions assigned to newly created files and directories. Newly created files or directories will be accessible to unauthorized and possibly malicious users if the umask is too permissive, but some applications may not function correctly if the umask is too restrictive.

The umask controls the standard access mode associated with new files and directories. By default, the system creates files with mode 666 and directories with mode 777. To determine what permissions a given umask will assign to a newly created file, subtract the umask from 666. A 022 umask, for instance, would result in mode 644 for newly created files, with the file creator having read and write permissions while assigning read permissions to group and other.

### 3.1.3 Access Control Lists (ACLs)

RHEL6 supports ACLs to provide more granular control of file permissions. ACLs enhance security by allowing the assignment of permissions to multiple specific users and groups.

As ACLs are not a traditional UNIX concept, their implementation varies significantly between platforms. Two ACL models used by multiple platforms are NFSv4 ACLs and POSIX ACLs. NFSv4 ACLs are defined in RFC 3530 and are similar to the Windows NT ACL model. POSIX ACLs are based on the withdrawn draft POSIX standards Institute of Electrical and Electronics Engineers (IEEE) 1003.2c and 1003.1e. The ACL models available are also dependent on the file systems used.

RHEL6 systems report the existence of additional or alternate permission mechanisms, such as ACLs, with an additional character, typically a '+', at the end of the permissions field in the output of ls –l.

ACLs have the ability to grant additional discretionary permissions beyond those represented in the file's access mode. It is, therefore, essential that the presence and content of ACLs be considered when evaluating file permissions.

### 3.1.4 Links

From a file system perspective, links refer to multiple paths that point to a single physical file. RHEL6 file systems generally support two kinds of links: hard links and symbolic links.

#### 3.1.4.1 Hard Links

Hard links are directory entries pointing to the same physical file within a single file system. There is no distinction between the original file and a hard link pointing to the file. Ownership and modes are attributes of the physical files and, therefore, only need to be checked or changed using one path. If permissions on intermediate traversed directories are used to restrict access to a file, each path to the file must be examined to verify correct configuration.

#### 3.1.4.2 Symbolic Links

Symbolic links, often referred to as symlinks, are special files that reference another path on the system. Unlike hard links, symlinks may link to paths on different file systems and have ownership attributes that are distinct from the referenced file. As with ordinary files, symlinks may be removed or deleted by any user with permission to write to the directory containing the file. The only permission given to the symlink owner is the ability to remove or rename a link when located in a directory with the sticky bit set. All other permissions are controlled by the mode of the target file. When evaluating the ownership or mode of a path that is a symlink, the attributes of the referenced path must be evaluated instead of the attributes of the symlink itself. This can be accomplished using the "-L" option of ls (1).

### 3.2 Services and Daemons

As a multi-tasking server operating system, RHEL6 has a concept of background processes that provide services to clients, both local to the system and remotely through the network. The RHEL6 STIG uses the terms "service" and "daemon" to refer to this concept. Services are commonly started automatically by the system run control scripts and may have processes in place to provide for automatic restart in the event of a software failure. Some daemons provide the capability for reloading their configuration upon receiving a signal from another process, permitting configuration changes without interruption of service.

# 4. SOFTWARE PATCHING GUIDELINES

Maintaining the security of a RHEL6 system requires frequent reviews of security bulletins. Many security bulletins and IAVM notifications mandate the installation of software patches to overcome noted security vulnerabilities. The SA will be responsible for installing all such patches. The IAO will ensure the vulnerabilities have been remedied. FSO guidelines for remediation, including IAVMs, are as follows:

- Apply the applicable patch, upgrade to required software release, or remove the binary/application to remediate the finding.

- Or, the mode of the vulnerable binary may be changed to 0000 to downgrade the finding (for example, a CAT I finding may be downgraded to a CAT II).

SAs and IAOs will regularly check Red Hat's vendor and third-party application vendor websites for information on new vendor-recommended updates and security patches that are applicable to their site. All applicable vendor-recommended updates and security patches will be applied to the system. A patch is deemed applicable if the product is installed, even if it is not used or is disabled.

## 5. OPEN SOURCE SOFTWARE (OSS) POLICY

DoD has clarified policy on the use of OSS to take advantage of the capabilities available in the Open Source community, as long as certain prerequisites are met. DoD no longer requires that operating system software be obtained through a valid vendor channel and have a formal support path if the source code for the operating system is publicly available for review.

*From the DoD CIO Memo, Open Source Software (OSS) in Department of Defense (DoD), 28 May 2003:*

> "DoD Components acquiring, using or developing OSS must ensure that the OSS complies with the same DoD policies that govern Commercial-Off-The-Shelf (COTS) and Government-Off-The Shelf (GOTS) software. This includes, but is not limited to, the requirements that all information assurance (IA) or IA-enabled IT hardware, firmware and software components or products incorporated into DoD information systems whether acquired or originated within DoD;
>
> - Comply with the evaluation and validation requirements of National Security Telecommunications and Information Systems Security Policy Number 11 and;
>
> - Be configured in accordance with DoD-approved security and configuration guidelines at http://iase.disa.mil/ and http://www.nsa.gov/."

OSS takes several forms and may be acceptable or unacceptable depending on the form:

1. A utility that has publicly available source code is acceptable.
2. A commercial product that incorporates OSS is acceptable because the commercial vendor provides a warranty.
3. Vendor-supported OSS is acceptable.
4. A utility that comes compiled and has no warranty is not acceptable.

DoDD 8500.1 states "Public domain software products, and other software products with limited or no warranty, such as those commonly known as freeware or shareware, shall only be used in DoD information systems to meet compelling operational requirements. Such products shall be thoroughly assessed for risk and accepted for use by the responsible DAA."