Linux Security on HP Servers: Security Enhanced Linux

Technical introduction
This white paper – one in a series of Linux security white papers – discusses Security Enhanced Linux (SELinux), a mandatory access control system that supplements and greatly enhances the security of the discretionary access control provided by traditional Linux ownership and permissions.

Abstract
The power and flexibility of Linux can be employed to greatly enhance its security in many ways. This series of white papers discusses how to take full advantage of the many security features available in the Red Hat Enterprise Linux and SUSE Linux Enterprise Server operating systems, to ensure your servers and data center are as secure as possible.

This white paper, Linux Security on HP Servers: Security Enhanced Linux, discusses how to enhance the security of your system by supplementing the discretionary access controls provided by traditional Linux file ownerships and permissions with the much more secure mandatory access controls of Security Enhanced Linux (SELinux). SELinux provides comprehensive fine-grained access control, which can protect resources beyond just files and directories (for example, sockets, system calls, and network interface reads and writes).

Intended Audience
This white paper series is intended for administrators of HP Linux-based servers needing to better secure those servers by gaining a greater understanding of the full arsenal of security tools that the Red Hat Enterprise Linux (RHEL), and SUSE Linux Enterprise Server (SLES) operating systems provide; by those evaluating Linux on HP servers to determine its security capabilities; and, by those comparing Linux security technologies against what is available on other operating systems.
Table of Contents

Table of Contents................................................................................................................................. 2
Overview............................................................................................................................................ 3
  Introduction ..................................................................................................................................... 3
  Focus on Server Security .............................................................................................................. 3
Introduction to SELinux ....................................................................................................................... 4
  Advantages of SELinux ................................................................................................................. 4
  How SELinux Works ....................................................................................................................... 4
  Special Considerations When Using SELinux ..................................................................................... 6
    Displaying SELinux Security Contexts .......................................................................................... 7
    File Management and Data Archival ............................................................................................. 7
    Roles and The Superuser .............................................................................................................. 8
SELinux Configuration Tools .................................................................................................................. 8
  Adjusting the Default Configuration ............................................................................................. 9
  Changing Booleans ........................................................................................................................ 9
  Managing Policy Modules ........................................................................................................... 9
Using HP Software with SELinux ......................................................................................................... 10
  HP Insight Control Environment .................................................................................................. 10
  HP ProLiant Support Pack ......................................................................................................... 10
Custom Policy Development ................................................................................................................. 10
  Writing Your Custom Policy ......................................................................................................... 10
    Modular SELinux Policy ........................................................................................................... 11
  Policy Development Tools .......................................................................................................... 11
    Development Wizards .............................................................................................................. 11
    Policy Development Environments ........................................................................................... 11
Troubleshooting SELinux ..................................................................................................................... 12
Summary .......................................................................................................................................... 13
For more information.......................................................................................................................... 13
Additional Linux Security Resources................................................................................................... 13
Overview

Introduction

Unless your server is locked in a computer room and not connected to any network, it is potentially vulnerable to unauthorized access, primarily via its network ports. Unless it is properly protected, the data on your server might be retrieved, deleted, or altered by intruders.

In order to maximize the security of an HP Linux-based server, it must be properly configured. This white paper will discuss what is probably the single most effective thing you can do to secure a Red Hat Enterprise Linux Server: enable SELinux, and running in enforcing mode. If your HP Linux server is running the SUSE Linux Enterprise Server (SLES) operating system, use AppArmor, a technology similar to SELinux for implementing mandatory access control of your server’s files and directories on SLES-based servers.

In addition to using the mandatory access control provided by either SELinux or AppArmor, there are many other steps you can (and should) take to secure your HP Linux servers. The following topics are covered by other white papers in this series:

Linux Security on HP Servers: General Security Topics:

- Regularly install the latest available operating system and application software patches to ensure you are running the latest available security patches.
- Carefully control access to all superuser accounts.
- Enforce good password policy, ensuring that all of your user accounts have strong passwords that are changed regularly.
- Harden your server’s file systems by ensuring permissions are properly set so that access to each file, directory, and file system is restricted to only those users that must have it.
- Encrypt sensitive data so that if your data is accessed without proper authorization it will be difficult for the intruder to use it.
- Restrict the number of services your server provides to only those that are essential for its operation.
- Audit and log events on your server so that if intrusions do occur, you can determine what was compromised, when, and by whom.
- Use an intrusion detection system to regularly verify that your server’s installed software packages and their associated configurations have not been altered.

Linux Security on HP Servers: Securing the Network Boundary:

- Configure network applications (services) running on your server to limit their use to only the users that you authorize.
- Run a properly configured firewall.
- Encrypt sensitive network traffic traveling to and from your server.

In general, do not run anything that is not essential, tightly control network access, isolate processes and users from each other so that they can only access system resources that are necessary for their function, log all activity in case you need to reconstruct a chain of events, and encrypt sensitive data to limit its exposure if your system is compromised.

Focus on Server Security

While all systems, including workstations and PCs, can benefit from many of the recommendations in this white paper, this paper will focus primarily on servers such as those found in large data centers. These machines frequently serve many users over large geographic areas, require nearly 100% uptime, often contain sensitive and personal data that must be protected (in many cases, by law), and are frequently targets of malicious attacks.
Introduction to SELinux

SELinux is a security subsystem that implements mandatory access control, a mechanism that enforces pre-defined security policy on a server. Properly configured mandatory access control supplements the discretionary access control provided by the standard Linux permissions and ownership. The default SELinux policy restricts access to many of the services provided by Red Hat Enterprise Linux, allowing access only to what is required to operate.

Advantages of SELinux

In its default form (using the “targeted” policy that ships with Red Hat Enterprise Linux), SELinux is easy to configure and provides significant improvement over traditional Linux security mechanisms. It prevents network-visible services from accessing system resources that they have no need to access, preventing damage from many common types of network attacks.

Specifically, the “targeted” policy protects the following daemons and the services they provide:

- dhcpd – Dynamic Host Configuration Protocol daemon
- httpd – Apache web server daemon
- named – Internet Domain Name System (DNS) daemon
- nscd – Name Service Cache daemon
- ntpd – Network Time Protocol daemon
- portmap – DARPA port to RPC program number mapper
- snmpd – Simple Network Management Protocol daemon
- squid – Web caching proxy daemon
- syslogd – System Logging daemon

Best Practice:

Using the default “targeted” policy that ships with Red Hat Enterprise Linux SELinux is nearly transparent to normal daily use of your system, yet it significantly enhances the security of your system.

How SELinux Works

SELinux uses a system wide security policy (set of rules) to restrict the actions of applications running on a server in addition to whatever restrictions are provided by the traditional Linux security mechanisms of permissions and ownership. This additional layer of security is important because permissions and ownership are often not granular enough to fully contain the behavior of running applications. For example, they can leave resources that are owned by the user of an application vulnerable to the actions of that application, even when the application has no legitimate need to access those resources.

SELinux security does not replace traditional Linux security. In fact, for resources secured by permissions and ownership, if the permissions and ownership do not permit access to a user or process the additional SELinux security controls do not factor into the access decision; access is denied before SELinux. As Figure 1: Security Process Flow demonstrates, SELinux is consulted only after traditional permissions and ownership requirements are satisfied, to ultimately determine whether access to the resource will be granted.
One of the strengths of SELinux over traditional Linux security mechanisms is that it can protect resources that permissions and ownership do not. SELinux can also protect items such as system calls, sockets, network reads and writes, and inter-process communications.

SELinux uses policy rules to determine whether or not to permit the user or application access to a requested resource. By default, SELinux denies access to an object unless the policy rules specifically allow it. Policy rules specify whether or not access to a resource is allowed based on:

- The type of access being requested
- The security context of the user or process requesting the access
- The security context of the requested resource

SELinux policy is loaded directly into the running kernel at boot time, making it difficult to change (both accidentally and deliberately) once a system is online. Certain aspects of the policy can be controlled by switches, called booleans that allow you to choose between several predefined behaviors. For more information on booleans, see Changing Booleans.
When a user or process requests access to a resource (for example, when opening a file), SELinux uses the three items in the previous list (the type of access requested, the security context of the process, and the security context of the requested resource) to determine whether to allow access.

The security context of a process, commonly called a domain, determines what the process can do (what it can access, and what actions it can take). Users have security contexts too; a user’s security context can be viewed and set using the `semanage` tool.

The security context of an object, commonly called a type, is similar to the security context of a user or process. It represents what the security attributes of the object are; that affects which users and processes security contexts can access the object (and how).

**Special Considerations When Using SELinux**

Under the targeted policy that ships with Red Hat Enterprise Linux, SELinux is virtually transparent to users of a secured server. It does not interfere with normal, legitimate, use of the server; although there are some important things to know about systems running SELinux.
Displaying SELinux Security Contexts

On file systems that support extended attributes (for example, ext3), SELinux contexts for files and directories are stored in the file’s or directory’s extended attributes on disk. On file systems that do not support extended attributes, a common security context can be defined for every file in the file system using mount options. The `mount` command has several useful options for assigning contexts:

<table>
<thead>
<tr>
<th>mount Option</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>context=ThisContext</td>
<td>Associates the context <code>ThisContext</code> with the files and directories on the specified file system. Example: mount context=&quot;system_u:object_r:removable_t&quot; -r /dev/dvd /dvd</td>
</tr>
<tr>
<td>fscontext=context</td>
<td>Specifies a context for the file system itself (as opposed to the contents of the file system)</td>
</tr>
<tr>
<td>defcontext=context</td>
<td>Specifies a default context for any object that otherwise does not have one, overriding (for the files and directories on this file system) the value set for unlabeled files in the policy.</td>
</tr>
</tbody>
</table>

You can see the security context of a file or directory by using the `-Z` option of the `ls` command:

```
# ls -dZ /usr/sbin
drwxr-xr-x  root root  system_u:object_r:bin_t /usr/sbin
```

Best Practice:

Wherever possible, use file systems that support extended attributes. SELinux stores security context information in the extended attributes area of files and directories. When using file systems that do not support extended attributes, SELinux is forced to assign all objects within the affected file systems a common security context, which makes it difficult to perform per-file access control on the file system.

Files and directories inherit their security context from their parent directory unless policy specifically overrides this via a type transition based on the security context of the process and the destination directory.

Running processes obtain their security context by inheriting it from their parent process unless policy overrides this via a domain transition or the parent process explicitly specifies the security context for new processes. For example, in the targeted policy domain transitions specify that specific daemons run with special contexts that severely restrict what they can do and which resources they can access. Under the targeted policy, all other processes run with a special context known as `unconfined_t` which gives them exactly the same access privileges they would have on a system not running SELinux.

To see what domain a process is running in, use the `-Z` option to the `ps` command:

```
# ps -Z
```

### File Management and Data Archival

Because of the way SELinux assigns security contexts to files and directories, there are some important extra things you have to consider when managing files.
The Difference between Move and Copy

There is an important difference between how the `mv` command and the `cp` command assign security contexts to their respective destination files. To understand why there are differences consider the basic intent of each command.

The intent of the `mv` command is to relocate an existing file or directory. Therefore, the goal of the `mv` command with respect to the context of the destination is to maintain the context of the file or directory being moved. Sometimes this can cause minor problems when a file has a context different than other files found in the destination directory.

The intent of the `cp` command is to create a new file or directory. Therefore, the destination file is created, depending on the SElinux policy, with a context based either on the parent directory or the combination of the context of the `cp` process and the destination directory. However, there are several options to the `cp` command that can alter this basic behavior. The `-p` option attempts to preserve the attributes, including the security contexts of the source files and the `-Z` option lets you specifically specify the security contexts of the destination files.

Archiving and Restoring SElinux Contexts

The security context of a file or directory is stored in the extended attributes portion of that file or directory on disk. Therefore, the important thing to ensure when creating archives of files on an SElinux system is to use a utility that includes the extended attributes in the archives (if you need to retain the original context when restoring those files). The `tar` utility will do this if you use the `--selinux` option.

Roles and The Superuser

In the default targeted policy, SElinux roles are not particularly significant. That is, although they are fully operational, they are not defined to largely affect the day to day use of the system or what users can and cannot access. If your security needs require it, policy can be added, or other policy can be used, to better enable the SElinux role based access controls.

SElinux roles are used to further define what a given SElinux user can do by restricting which domains they can transition to (and as a result which applications they can run) based on their current role. The roles act like operational modes. For example, the targeted policy defines SElinux roles called `staff_r` and `sysadm_r`, with associated domain types `staff_t` and `sysadm_t`, respectively: the latter domain being allowed to perform critical system administration functions (those typically associated with the root user on a non-SElinux system). One or more SElinux users can be created and assigned both the `staff_r` and `sysadm_r` roles, with the users normally running in the `staff_r` role and only transitioning to the `sysadm_r` role when they need to perform administrative duties. Linux users would then be assigned these SELinux users with both roles and configured to start in the `staff_r` role. While operating in the `staff_r` role, they will be unable to perform administrative commands, avoiding unintentional administrative changes to the system. When they need to intentionally administer the system, they would transition from the `staff_r` role to the `sysadm_r` role. This role change allows the change from the `staff_t` domain which is not permitted to administer the system to the all important `sysadm_t` domain which is permitted to administer the system.

Roles can also be used for other purposes, but these are beyond the scope of this white paper.

SElinux Configuration Tools

There are several important tools for configuring the behavior of your SElinux based system. Policy can be written with internal “switches” known as booleans that can enable or disable various behaviors, and new SElinux policy modules can be written to extend policy to further tighten security.
Adjusting the Default Configuration

As shipped with Red Hat Enterprise Linux, SELinux is configured to use the targeted policy. The targeted policy protects key networking services that are frequent targets of intruders.

In its default configuration, the targeted policy does a very good job of protecting your server. Additionally, SELinux is highly configurable and can be customized to meet almost any security requirement you have by both altering booleans to turn on or off protections for specific services, and by developing your own custom policy modules (or using third party modules from a trusted source).

Changing Booleans

SELinux policies can be written to check the current value of special variables called booleans, and can adjust policy behavior based on the boolean values. This provides a mechanism for easily enabling or disabling various components of the configured policy without having to rewrite, compile, and install a new set of policy. To use this feature, the installed policy must be written to support booleans. Policy that does not include boolean definitions cannot have its behavior altered while the system is running.

Booleans can be managed using the `getsebool` and `setsebool` commands. To see which booleans are currently in use on your server, use the command:

```
# getsebool -a
```

To see the current value for a specific boolean, replace the `–a` option in the above command with the name of the boolean you are interested in. For example,

```
# getsebool user_ping
```

To set the value for a specific boolean, use the `setsebool` command. For example,

```
# setsebool user_ping off
```

This will set the value for the boolean until the system is rebooted. To make the change persistent across reboots, add the `–P` option to the `setsebool` command:

```
# setsebool –P user_ping off
```

For complete details on these commands, see their respective manpages, `getsebool(8)` and `setsebool(8)`.

Managing Policy Modules

Beginning with Red Hat Enterprise Linux 5, SELinux policy is modular; enabling the customization, addition or removal of individual policy modules to meet your own security requirements. Modular policy management means that portions of policy can be installed and removed as needed without having to reboot your system. You can:

- Install policy modules for testing
- Remove modules that are causing problems
- Select which standard modules to use, using only the ones that meet your security requirements

You manage the modules on your system using the `semodule` command. `semodule` enables you to install, remove, upgrade, and list the policy modules that are used by your system. Unlike dynamically loaded kernel modules, SELinux modules installed by `semodule` will survive across boots.

Standard Red Hat Modules

Red Hat Enterprise Linux ships with many policy modules for the targeted policy. They are located in the `/usr/share/selinux/targeted` directory. To list the modules that are currently active on your system, in addition to the always loaded base policy, use the command:

```
# semodule -l
```
Custom or Third-Party Modules

In addition to the standard SELinux policy modules, you can install your own custom policy modules. These modules might come from a third-party, be custom modifications created by you, or be created by you via a command like `audit2allow` in response to audit denial messages in your log files. However the policy modules are created, if they are in the properly compiled policy package format you can install custom policy modules the same way you install standard policy modules, using `semodule`.

Using HP Software with SELinux

Some software packages, particularly those that assist you with system administration, require adjusting your security settings in order to use them; because, by their very nature, they need access to files and services that are not normally required by users or applications. This is not to say that these packages are inherently insecure, but rather that you need to be careful when configuring them to not degrade the security of your system.

HP Insight Control Environment

The HP Insight Control Environment (HP ICE-Linux) works in conjunction with HP Systems Insight Manager to deploy Linux on HP servers. HP ICE-Linux is compatible with SELinux though you might need to add additional SELinux policy to accommodate its operation. For details on how to do this, see the document `Using SELinux on an ICE-Linux CMS`, located at: [http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c01445000/c01445000.pdf](http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c01445000/c01445000.pdf)

HP ProLiant Support Pack

The HP ProLiant Support Pack is a collection of optimized drivers, utilities and agents that have been tested together to ensure proper installation and operation. They are generally compatible with SELinux, though in some cases you might need to add additional SELinux policy – for example, by using the `audit2allow` command previously described – to make them fully compatible.

Custom Policy Development

Though the targeted policy that ships with Red Hat Enterprise Linux is sufficient for many security needs, occasionally you might need to customize the policy on your system. For example to:

- Allow processes in a specific domain access to a file originally denied by the targeted policy
- Lock down daemons or processes unique to your business
- Eliminate SELinux access control notices that you choose to ignore (if the messages cannot be eliminated by adjusting a boolean value)

Writing Your Custom Policy

The detailed steps involved in writing custom policy are beyond the scope of this white paper, but the general steps are:

1. Determine what processes you want to lock down, and which resources they must have access to in order to properly function. When doing this, keep in mind the principle of “least privilege”; you want to allow access to only the resources that are required for the process to properly function, and no more.
2. Write/Edit the policy source files (see `Policy Module Format` below) to assign each process to a domain, and specify what processes running in that domain are allowed to do. Compile and install the new policy module.
3. Fix any file system labeling for files that will be affected by your new policy.
You can manually create the source files using your favorite text editor, compile them using the `checkmodule` command, package the resulting binary files using the `semodule_package` command, and install the package using the `semodule` command. There are also tools that can walk you through the policy development process and do most of the heavy lifting for you. For information about those tools, see Policy Development Tools.

**Best Practice:**

When writing custom policy, be aware that the default SELinux behavior is to prohibit access. The policy you are developing overrides that behavior (like punching holes in armor). Permit access to only what is absolutely necessary and nothing more.

**Modular SELinux Policy**

Modern SELinux policy is contained in policy modules and the easiest way to extend your system’s SELinux policy is by writing, compiling, and loading new modules into the kernel.

Modular SELinux policy provides the following benefits:

- New modules can be developed and added to an existing SELinux system when new security needs develop
- Modules can be developed around specific security goals keeping them focused, to minimize policy development errors and make SELinux errors easier to isolate and troubleshoot
- Modules can be removed when the security goals they served are no longer applicable so that maximum system performance and security can be maintained.

**Policy Module Format**

Policy modules are binary files packaged from the following types of source files:

- A policy configuration file (.te) containing the SELinux policy rules
- A policy interface file (.if) which defines how other modules can interact with the new module you are creating
- A policy file context file (.fc) which maps SELinux security contexts to file system objects

**Policy Development Tools**

Developing SELinux policy can be done manually, but to minimize errors and make the process easier, consider using some of the following tools.

**Development Wizards**

A GUI SELinux policy generation wizard called `polgengui`, will create the source files mentioned above, along with a shell script that will compile and install the policy for you.

The command `audit2allow` can read your system’s log files (/var/log/messages, the output of /bin/dmesg, or the audit log) and build loadable policy modules from SELinux access denial notification that it finds. This new policy contains the rules needed to prevent these access denials.

**IMPORTANT:** Not all access denial notifications are bad. Many of them reflect SELinux doing its job and reflect intrusion attempts that were successfully blocked. See Troubleshooting SELinux below for more details.

**Policy Development Environments**

In addition to policy development wizards, there are several policy development environments that can help you look at your system’s policy and the interactions and affects of the various policy rules. One such tool is called SLIDE, which is a plug-in for the Eclipse Software Development Kit.
Troubleshooting SELinux

Because SELinux is a security tool and security tools are focused on controlling access to system resources, nearly all SELinux problems will be related to access control; for example, providing access to resources for someone who legitimately needs access, but does not have it; or denying access to someone who should not have access to resources, but does.

To prevent users and applications who can currently access resources they should not be able to access, you will need to identify, and remove (or modify), the policy rules that are permitting the access (or adjust a policy boolean) to correct the condition. Most likely, you will not notice any SELinux error messages indicating that someone who should not have access to a resource successfully accessed it. You will need to use other methods, such as auditing, to determine if unwanted accesses are occurring.

When users and applications should have access to resources but SELinux prevents them from successfully gaining that access, access denial messages will be written to the file /var/log/messages, or /var/log/audit/audit.log. The messages will begin with the string “avc: denied” and will include information about the process that was denied access, including the path to its executable and its security context. Many of these access denial messages are legitimate and alert you to instances where SELinux did its job properly, denying access to users and applications that should not have it. However, sometimes you know that an application requires access to function properly and it is being denied that access. SELinux by default will deny access unless it specifically has a rule instructing it to grant the access. So, in most cases, you will need to add new policy rules (or adjust a policy boolean) to allow the required access.

In addition to the policy development tools mentioned previously, a special tool called audit2why can help you determine why an access request was denied by showing you the policy rules that prohibited the access:

- A boolean setting might be blocking a policy’s allow rule that allows the needed access. If this is the case, you can change the boolean setting (if doing so would not compromise other security aspects of the system).
- A policy allow rule might not yet exist for the type of access you need. Another tool called audit2allow can use the “avc: denied” error messages to generate the required new policy rules.

The SELinux access denial messages can be quite cryptic, and the output of audit2why can be terse. A better tool is available to help administrators new to SELinux understand what happened: a service called SETroubleshoot. SETroubleshoot consists of two parts:

- A daemon: setroubleshootd, that monitors the logging of access denials and analyzes the messages, then reports what went wrong using a much more human-friendly form, along with recommendations on what to do about it.
- A user interface application: sealert, that allows users to browse, or be alerted to, reports from setroubleshootd. sealert can be run as a GUI or from the command line.

You can also have setroubleshootd email you alerts, and sealert can monitor reports from either the local machine or remote machines over the network.
Summary

Security is a crucial element in nearly every computing environment. There are many ways your system and information could come under attack. Using mandatory access control, provided by SELinux, is one of the most effective ways to secure a Red Hat Enterprise Linux based system.

Using the targeted policy, SELinux both significantly enhances the security of your system and remains relatively unobtrusive. If you require even more security, SELinux provides more substantial policy levels (MLS and Strict), though these are not needed for most installations.

If you need mandatory access control to secure a SUSE Linux Enterprise Server based system, you need to use AppArmor.

Though SELinux provides significant protection for the information on your system, there are other technologies you need to deploy to protect your information as it travels to and from your system, especially if that information travels over a public or other untrusted network. Many of these technologies are discussed in Security on HP Servers: Securing the Network Boundary.

There are other important ways to secure your system (for example, ensuring your system’s software and operating system are up-to-date, ensuring the use of secure passwords, and logging and auditing the activities of the users of your system). Many of these technologies are covered in Linux Security on HP Servers: General Security Topics.

For more information

Additional Linux Security Resources

If you require additional information on the topics covered in this white paper, here are some important resources to reference:


If you have further questions about Linux security on HP servers, go to [http://www.hp.com/go/linux](http://www.hp.com/go/linux), and click Contact HP.