Grsecurity

en.wikibooks.org

July 26, 2015

Contents

1 Introduction

grsecurity is a set of patches[1] for the Linux kernel[2] with an emphasis on enhancing security. Its typical application is in web servers and systems that accept remote connections from untrusted locations, such as systems offering shell access[3] to its users.

Released under the GNU General Public License[4], grsecurity is free software[5].

1.1 History

Work on grsecurity began in February 2001 as a port of Openwall Project's[6] security-enhancing patches for Linux 2.4. The first release of grsecurity was for Linux 2.4.1.

1.2 PaX

A major component bundled with grsecurity is PaX[7], which is a patch that, amongst other things, flags data memory, such as that on the stack[8], as non-executable, and program memory as non-writable. The aim is to prevent executable memory pages from being overwritten with injected machine code, which prevents exploitation of many types of security vulnerabilities, such as buffer overflow[9]s. PaX also provides address space layout randomization[10] (ASLR), which randomizes important memory addresses to hinder attacks that rely on such addresses being easily known. PaX is not itself developed by the grsecurity developers, and is also available independently from grsecurity `http://pax.grsecurity.net`.

1.3 Role-based Access Control

Another notable component of grsecurity is that it provides a full Role-based access control[11] (RBAC) system. RBAC is intended to restrict access to the system further than what is

1 `http://en.wikipedia.org/wiki/patch%20%28computing%29`
2 `http://en.wikipedia.org/wiki/Linux%20kernel`
3 `http://en.wikipedia.org/wiki/shell%20account`
4 `http://en.wikipedia.org/wiki/GNU%20General%20Public%20License`
5 `http://en.wikipedia.org/wiki/free%20software`
6 `http://www.openwall.com/`
7 `http://en.wikipedia.org/wiki/PaX`
8 `http://en.wikipedia.org/wiki/stack%20%28data%20structure%29`
9 `http://en.wikipedia.org/wiki/buffer%20overflow`
10 `http://en.wikipedia.org/wiki/address%20space%20layout%20randomization`
11 `http://en.wikipedia.org/wiki/role-based%20access%20control`

normally provided by Unix[12] access control list[13]s, with the aim of creating a fully least-privilege system, where users and processes have the absolute minimum privileges to work correctly and nothing more. This way, if the system is compromised, the ability by the attacker to damage or gain sensitive information on the system can be drastically reduced. RBAC works through a collection of "roles". Each role can have individual restrictions on what they can or cannot do, and these roles and restrictions form a "policy" which can be amended as needed.

A list of RBAC features:

- Domain support for users and groups
- Role transition tables
- IP-based roles
- Non-root access to special roles
- Special roles that require no authentication
- Nested subjects
- Variable support in configuration
- And, or, and difference set operations on variables in configuration
- Object mode that controls the creation of setuid and setgid files
- Create and delete object modes
- Kernel interpretation of inheritance
- Real-time regular-expression resolution
- Ability to deny ptraces to specific processes
- User and group transition checking and enforcement on an inclusive or exclusive basis
- */dev/grsec* special device for kernel authentication and learning logs
- Next-generation code that produces least-privilege policies for the entire system with no configuration
- Policy statistics for gradm
- Inheritance-based learning
- Learning configuration file that allows the administrator to enable inheritance-based learning or disable learning on specific paths
- Full pathnames for offending process and parent process
- RBAC status function for gradm
- */proc/<pid>/ipaddr* gives the remote address of the person who started a given process
- Secure policy enforcement
- Supports read, write, append, execute, view, and read-only ptrace object permissions
- Supports hide, protect, and override subject flags
- Supports the PaX flags
- Shared memory protection feature
- Integrated local attack response on all alerts
- Subject flag that ensures a process can never execute trojaned code
- Full-featured fine-grained auditing
- Resource, socket, and capability support
- Protection against exploit bruteforcing
- */proc/pid* filedescriptor/memory protection
- Rules can be placed on non-existent files/processes

12 `http://en.wikipedia.org/wiki/Unix`
13 `http://en.wikipedia.org/wiki/access%20control%20list`

- Policy regeneration on subjects and objects
- Configurable log suppression
- Configurable process accounting
- Human-readable configuration
- Not filesystem or architecture dependent
- Scales well: supports as many policies as memory can handle with the same performance hit
- No runtime memory allocation
- SMP safe
- O(1) time efficiency for most operations
- Include directive for specifying additional policies
- Enable, disable, reload capabilities
- Option to hide kernel processes

1.4 Chroot Restrictions

grsecurity restricts chroot[14] in a variety of ways to prevent a variety of vulnerabilities, privilege escalation attacks, and to add additional checks and balances.

Chroot Modifications:

- No attaching shared memory outside of chroot
- No `kill` outside of chroot
- No `ptrace` outside of chroot (architecture independent)
- No `capget` outside of chroot
- No `setpgid` outside of chroot
- No `getpgid` outside of chroot
- No `getsid` outside of chroot
- No sending of signals by `fcntl` outside of chroot
- No viewing of any process outside of chroot, even if */proc* is mounted
- No mounting or remounting
- No pivot_root
- No double chroot
- No `fchdir` out of chroot
- Enforced `chdir("/")` upon chroot
- No `(f)chmod +s`
- No `mknod`
- No `sysctl` writes
- No raising of scheduler priority
- No connecting to abstract Unix domain sockets[15] outside of chroot
- Removal of harmful privileges via capabilities

14 `http://en.wikipedia.org/wiki/Chroot`
15 `http://en.wikipedia.org/wiki/Unix_domain_socket`

1.5 Miscellaneous Features

grsecurity also adds enhanced auditing[16] to the Linux kernel. It can be configured to audit a specific group of users, audit mount[17]s/unmounts of devices, changes to the system time and date, chdir[18] logging, amongst other things. Some of these other things allow the admin to also log denied resource attempts, failed fork attempts, and exec logging with arguments.

Trusted path[19] execution is another optional feature that can be used to prevent users from executing binaries that are not owned by the root[20] user, or are world-writable. This is useful to prevent users from executing their own malicious binaries or accidentally executing system binaries that could have been modified by a malicious user (being world-writable).

grsecurity also hardens the way chroot "jails" work. A chroot jail can be used to isolate a particular process from the rest of the system, which can be used to minimise the potential for damage should the service be compromised. However, there are ways to "break out" of a chroot jail. grsecurity attempts to prevent this.

There are also other features that increase security and prevent users from gaining unnecessary knowledge about the system, such as restricting the `dmesg`[21] and `netstat`[22] commands to the root user `http://www.grsecurity.net/features.php`.

List of additional features and security improvements:

- */proc* restrictions that don't leak information about process owners
- Symlink/hardlink restrictions to prevent */tmp* races
- Hardlink restrictions to prevent users from hardlinking to files they do not own
- FIFO[23]/Named pipe[24] restrictions
- `dmesg(8)` restriction
- Enhanced implementation of Trusted Path Execution
- Group-based socket restrictions
- Nearly all options are sysctl-tunable, with a locking mechanism
- All alerts and audits support a feature that logs the IP address of the attacker with the log
- Stream connections across unix domain sockets carry the attacker's IP address with them (on 2.4 kernels only)
- Detection of local connections: copies attacker's IP address to the other task
- Automatic deterrence of exploit bruteforcing
- Pre-defined Low, Medium, High, and Custom security levels
- Tunable flood-time and burst for logging

16 `http://en.wikipedia.org/wiki/auditing`
17 `http://en.wikipedia.org/wiki/mount%20%28Unix%29`
18 `http://en.wikipedia.org/wiki/chdir`
19 `http://en.wikipedia.org/wiki/Trusted%20path`
20 `http://en.wikipedia.org/wiki/superuser`
21 `http://en.wikipedia.org/wiki/dmesg`
22 `http://en.wikipedia.org/wiki/netstat`
23 `http://en.wikipedia.org/wiki/FIFO`
24 `http://en.wikipedia.org/wiki/Named_pipe`

This book uses many different terms, some of which have the same meaning. We have listed some of these terms and their definitions here. The book also contains inline links to relevant Wikipedia articles.

policy

The policy is **a system-wide set of rules** enforced by grsecurity. A very good description is offered in the mandatory access control[25] article: "Any operation by any subject on any object will be tested against the set of authorization rules (aka policy) to determine if the operation is allowed."

access control list

From a related Wikipedia article[26]: "An access control list (ACL) is **a list of permissions attached to an object** . The list specifies who or what is allowed to access the object and what operations are allowed to be performed on the object." In the context of this book, an ACL is used to mean a single role or subject definition, or the whole policy file.

ruleset

Ruleset is used much in the same way as "access control list". It is perhaps more often used to refer to role or subject definitions than the whole policy file.

object

An object is a part of the system that is used by the programs running on the system. It can be an absolute path to a file or a directory; a capability[27]; a system resource[28]; a PaX flag[29]; network access (IP ACLs).

subject

A subject uses and accesses objects and the ruleset of the subject enforces what objects it may use and in what way. In practice a subject is most often a program running on the system. In grsecurity, a subject is defined as an absolute path to the actual program executable (e.g. */sbin/init*) or a directory (e.g. */lib/hal/scripts*).

role

A role is an abstraction that encompasses traditional users and groups that exist in Linux distributions[30] and special roles, that are specific to grsecurity. Roles can be used to split the responsibility of system administration into smaller logical sets of responsibilities, such as "database administrator" or "DNS administrator". Compare this approach to having a single superuser[31] (e.g. *root*) that is used to do every administrative task on the system.

domain

With domains you can combine users that do not belong in the same group as well as groups so that they share a single policy. Domains work just like roles.

25 `http://en.wikipedia.org/wiki/Mandatory_access_control`
26 `http://en.wikipedia.org/wiki/Access_control_list`
27 Chapter 8.16 on page 99
28 Chapter 8.16 on page 105
29 Chapter 8.16 on page 99
30 `http://en.wikipedia.org/wiki/Linux_distribution`
31 `http://en.wikipedia.org/wiki/Superuser`

2 Installation

The following instructions will lead you through the process of downloading all the components necessary for using grsecurity on your system. Download each component to the same directory on your computer.

You need:

- The latest stable version of grsecurity.
- A matching version of gradm, the administration utility for grsecurity.
- Full source code of the Linux[1] kernel.

You also need to have necessary programs for building, configuring and installing a custom kernel for your system. The preferred way, and required tools, to do the installation depend on the Linux distribution you are using. If you encounter problems with configuring or installing the kernel, please consult your distribution's documentation.

2.1 Downloading grsecurity

Point your browser to `http://grsecurity.net/`. Click on the "Download" link and then "Stable". For the purposes of this document, we will be installing the latest stable grsecurity for kernel 3.2.50. Therefore the patch file will be called "grsecurity-2.9.1-3.2.50-201308052151.patch".

> All grsecurity packages have a version string in their names. It contains both the version of the release itself and the kernel version it is meant for. For example, the version string 2.9.1-3.2.50-201308052151 tells us that the version of this grsecurity release is 2.9.1 and it is meant for kernel version 3.2.50. The last section of the version is a timestamp.

In our case we downloaded the following files

- grsecurity-2.9.1-3.2.50-201308052151.patch
- grsecurity-2.9.1-3.2.50-201308052151.patch.sig - This is the digital signature[2] of this release.

1 `http://en.wikipedia.org/wiki/Linux`
2 `http://en.wikipedia.org/wiki/Digital%20signature`

2.2 Downloading gradm

When downloading `gradm` , the administration utility for grsecurity's role-based access control system, you must download the version that matches the version of the grsecurity patch you downloaded. `Gradm` is located on the same download page as grsecurity.

In our case we downloaded the following files

- gradm-2.9.1-201308021745.tar.gz
- gradm-2.9.1-201308021745.tar.gz.sig - This is the digital signature of this release.

2.3 Downloading the Linux Kernel

The grsecurity patches can only be applied to a vanilla[3] kernel. Many distributions modify the official kernel with additional patches, which means that any kernel source packages acquired through their package manager is very likely incompatible with grsecurity.

For this reason we will download the official unmodified kernel from `http://www.kernel.org/`. Download the full kernel source package and its signature (the ".sign" file), and make sure its version matches the version of the grsecurity patch you downloaded. In this document the version is 3.2.50. The required version is most likely not the latest, so you need to get it from the kernel archives[4].

> ⚠ **Warning**
>
> Official support for kernel version 2.6.32.61 closed at the end of 2013.

If you've got a terminal open, you can use the below commands to download both the kernel source and the signature to the current working directory:

```
$ wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.2.50.tar.bz2
$ wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.2.50.tar.sign
```

NOTE : The versions of the grsecurity patch and the kernel must match exactly.

2.4 Verifying the Downloads

The grsecurity and gradm packages have been cryptographically signed so that users can verify that the source code has not been modified since it was packaged. You can find the public key used to sign them from the same download page as grsecurity. Scroll down the page until you see a heading that says "Verify these downloads with GPG". Below the heading is a link to the public key. Download the key to the directory where you placed grsecurity.

3 http://en.wikipedia.org/wiki/Linux_kernel%23Development_model
4 http://www.kernel.org/pub/linux/kernel/

Before you can verify the downloads, you need to import the grsecurity key to your public keyring using Gnu Privacy Guard[5] (GPG). If you are unfamiliar with GPG and wish to know more, please refer to The GNU Privacy Handbook[6].

To import the key, run the following command in the directory where your grsecurity and its key were downloaded.

```
$ gpg --import spender-gpg-key.asc
gpg: key 4245D46A: public key "Bradley Spengler (spender)
 <spender@grsecurity.net>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

After importing the key, verify the downloaded grsecurity and gradm packages by running the below commands in your grsecurity directory:

```
$ gpg --verify grsecurity-2.9.1-3.2.50-201308052151.patch.sig
gpg: Signature made Mon 05 Aug 2013 06:55:44 PM PDT using DSA key ID 4245D46A
gpg: Good signature from "Bradley Spengler (spender) <spender@grsecurity.net>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9F74 393D 7E7F FF3C 6500  E778 9879 B649 4245 D46A

$ gpg --verify gradm-2.9.1-201308021745.tar.gz.sig
gpg: Signature made Fri 02 Aug 2013 02:45:37 PM PDT using DSA key ID 4245D46A
gpg: Good signature from "Bradley Spengler (spender) <spender@grsecurity.net>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9F74 393D 7E7F FF3C 6500  E778 9879 B649 4245 D46A
```

Below is an example of a failed signature verification. The patch file was modified on purpose to make the verification fail.

```
$ gpg --verify grsecurity-2.9.1-3.2.50-201308052151.patch.sig
gpg: Signature made Mon 05 Aug 2013 06:55:44 PM PDT using DSA key ID 4245D46A
gpg: BAD signature from "Bradley Spengler (spender) <spender@grsecurity.net>"
```

As long as `GPG` reports the signature is good, you do not need to worry about the warning about the key not being certified with a trusted signature. If you signed the grsecurity key with your own key, you will not get the warning. If the verification of either file failed (i.e. if you get the "BAD signature" message), re-download the file in question and try again.

The Linux kernel source packages have been signed as well. Please follow the instructions on the Linux kernel website[7] to verify the kernel source package.

When you have successfully verified the downloaded files, you are ready to configure grsecurity.

The following instructions will lead you through the process of patching the Linux kernel with grsecurity, configuring its features and compiling, and installing the patched kernel.

5 http://en.wikipedia.org/wiki/GNU%20Privacy%20Guard
6 http://www.gnupg.org/gph/en/manual.html
7 https://www.kernel.org/signature.html

2.5 Patching Your Kernel with grsecurity

In this document the kernel source archive is called *linux-3.2.50.tar* and the matching grsecurity patch *grsecurity-2.9.1-3.2.50-201308052151.patch* . Both files are in the same directory.

Change to the root user and run the following commands in the directory you downloaded the files to. The first command decompresses the Linux source package, and the second one applies the patch to the kernel. You may need to install the patch program with your preferred package management tool.

```
# tar -xf linux-3.2.50.tar
# cd linux-3.2.50
# patch -p1 < ../grsecurity-2.9.1-3.2.50-201308052151.patch
```

2.6 Configuring the Kernel

The kernel source package contains a generic configuration file that should work without any significant modifications. Your distribution may have its own process and tools for configuring and building the kernel, in which case you should consult their documentation. Nonetheless you should go through the options and make sure they match your hardware and current setup.

To configure the kernel using the default configuration as a base, change into the kernel source directory (e.g. */usr/src/linux-3.2.50*), and execute the below command.

```
$ make menuconfig
```

You may need to install missing packages and libraries - follow the error messages for direction.

The interactive Kernel Configuration menu will launch. In the 3.x and 2.6 kernels the grsecurity options are under Security options » Grsecurity. Detailed descriptions of each option and its effects on the system can be viewed online on the Grsecurity and PaX Configuration Options[8] page or by using the built-in help functionality of the kernel configuration system. Make sure you understand each option before you enable or disable them. Once you have exited the Configuration menu, you can launch it again by rerunning ”make menuconfig”.

It is recommended that you start by setting the Configuration Method option to Automatic and then configuring Usage Type and other options to fit your environment and needs. You can fine-tune all grsecurity and PaX settings in the Customize Configuration section, if needed.

8 Chapter 7.1 on page 53

2.6.1 Suggestions

- Enable the sysctl interface[9] (Grsecurity » Customize Configuration » Sysctl Support). It will enable you to change the options that grsecurity runs with without recompiling the kernel. This is a very helpful feature especially when you are using grsecurity for the first time. "Configuration Method - Automatic" enables this feature by default.
- Some auditing options produce a lot of log messages, most notably Exec and Chdir logging (GRKERNSEC_EXECLOG and GRKERNSEC_AUDIT_CHDIR, respectively). If you enable either of them, make sure your logging system is properly configured to prevent the logs from flooding. Check Grsecurity » Customize Configuration » Logging Options as well.

2.7 Compiling and Installing the Kernel

2.7.1 On Debian and Ubuntu

To compile the kernel and build a Debian package (deb), execute the below commands in the kernel source directory. Ubuntu users should reference the Ubuntu Community Page[10] and decide whether they wish to use the ubuntu-package overlay directory in building. For building on Maverick from a git checkout, see How to compile a Ubuntu 10.10 kernel[11]

```
# fakeroot make deb-pkg
```

To install the newly created Debian package, run:

```
# cd ..
# dpkg -i *.deb
```

For more information about building kernels in Debian, please refer to the Debian Linux Kernel Handbook[12].

2.7.2 Other Distributions

- Gentoo Linux: Gentoo Linux (x86) Handbook[13] and Hardened Gentoo[14]
- CentOS: `http://wiki.centos.org/HowTos/Custom_Kernel`
- Fedora (release 8 and later): `http://fedoraproject.org/wiki/Docs/CustomKernel`

9 Chapter 3.6.3 on page 24
10 https://help.ubuntu.com/community/Kernel/Compile
11 http://blog.avirtualhome.com/2010/11/06/how-to-compile-a-ubuntu-10-10-maverick-kernel/
12 http://kernel-handbook.alioth.debian.org/ch-common-tasks.html#s-kernel-org-package
13 http://www.gentoo.org/doc/en/handbook/handbook-x86.xml?part=1&chap=7
14 http://www.gentoo.org/proj/en/hardened/

2.7.3 Compilation Differences

As you are compiling a kernel patched with grsecurity, you will notice some differences. One of these differences appears towards the end of compilation, and may look similar to:

```
WARNING: modpost: Found 2820 section mismatch(es).
To see full details build your kernel with:
'make CONFIG_DEBUG_SECTION_MISMATCH=y'
```

This warning is harmless. As described by the PaX Team on the grsecurity mailing list:

```
the extra section mismatches are due to my changes, i explicitly
added detection for writeable function pointers which are potential
exploit targets, just to know how many of them there are. we've been
eliminating some of them already but this work will never finish.

as for what they are in general, a mismatch means an unwanted reference
from one section to another. say, accessing init code or data from
normal code/data is not good since init sections are freed up on boot,
so any reference to them must not exist from permanent sections.
```

You will also notice additional warnings emitted by the compiler when compiling a kernel patched with grsecurity. This is due to additional warning flags that have been added to the build process to help spot specific kinds of bugs. You can ignore these additional warnings.

2.8 Conditional Steps

2.8.1 Proprietary NVIDIA Driver Patching

If you're using grsecurity on a desktop and plan to use the proprietary NVIDIA drivers, you'll need to patch them to be able to function correctly with grsecurity. To do this, follow these steps:

- Download the NVIDIA driver .run file from NVIDIA's website.
- Download PaX's patch for the NVIDIA driver from `https://grsecurity.net/~spender/nvidia-drivers-352.09-pax.patch`
- Run sh <name of NVIDIA .run file> -x
- cd 'basename <name of NVIDIA .run file> .run'
- patch -p1 < ../nvidia-drivers-352.09-pax.patch
- Install the driver by running ./nvidia-installer

3 Administration

Gradm , the administration utility for the role-based access control system, is a powerful tool that parses your ACL[1]s (Access Control Lists), performs the enforcement of a secure base policy, optimizes the ACLs, as well as handles parsing of the learning logs, merges them with your ACL set and outputs the final ACLs.

Before you install `gradm` , boot into your patched grsecurity kernel. You can compile `gradm` in any kernel you wish, but the installation will fail if the kernel does not support grsecurity.

3.1 Installation

If your Linux distribution provides ready-made grsecurity kernel packages, they will very likely provide a package for `gradm` too. If that is the case you should consider using it before compiling `gradm` yourself.

Before compiling and installing `gradm` , make sure you have the following applications installed in your system: lex[2] or flex[3] and byacc[4] or bison[5]. If you need Pluggable Authentication Modules[6] (PAM) support, install the header files for your system; The package containing them will very likely be called libpam-dev or similar.

A note should be added to say that if you are compiling gradm on your default linux kernel without grsecurity support the compile will fail and that you will only be able to compile after you reboot into your new grsecurity enabled kernel.

Change to the directory you downloaded `gradm` and grsecurity to earlier. In this document the name of the compressed package is *gradm-3.1-201503211320.tar.gz* . Decompress the package and change to the *gradm* directory by executing the following commands:

```
$ tar xzf gradm-3.1-201503211320.tar.gz
$ cd gradm
```

To install `gradm` with PAM support, as a non-root user run:

```
$ make
```

1 http://en.wikipedia.org/wiki/Access_control_list
2 http://en.wikipedia.org/wiki/Lex_programming_tool
3 http://en.wikipedia.org/wiki/Flex_lexical_analyser
4 http://en.wikipedia.org/wiki/Byacc
5 http://en.wikipedia.org/wiki/GNU_bison
6 http://en.wikipedia.org/wiki/Pluggable_Authentication_Modules

NOTE: Look at the output from `make` . Make sure you do not see a line near the end that says "Unable to detect PAM headers, disabling PAM support." If you do, install the PAM header files and run the `make` command again.

To install `gradm` without PAM support, run:

```
$ make nopam
```

Finally, as root, run:

```
# make install
```

The installation process does the following:

- Installs the `gradm` and `grlearn` programs to */sbin* .
- Creates a directory */etc/grsec* and two files in it (if they are not already present): *learn_ config* and *policy* .
- Installs `gradm` 's man pages[7] to */usr/share/man/man8* .
- (`grlearn` does not come with a man page. It is used internally by `gradm` .)
- Finally, and most importantly, if this is the first time you are installing `gradm` on your system you will be asked to provide the administrative password for the RBAC system. Choose a long password, but one that you will remember (especially if you start `gradm` from an initscript). **Do not use** the same password as your root password.

If you need to change any of the binary or man page locations, modify the *Makefile* .

3.2 Usage

To display all available command-line switches, run `gradm --help` .

```
# gradm --help
gradm 3.1
grsecurity RBAC administration and policy analysis utility

Usage: gradm [option] ...

Examples:
    gradm -P
    gradm -F -L /etc/grsec/learning.logs -O /etc/grsec/policy
Options:
    -E, --enable    Enable the grsecurity RBAC system
    -D, --disable    Disable the grsecurity RBAC system
    -C, --check    Check RBAC policy for errors
    -S, --status    Check status of RBAC system
    -F, --fulllearn Enable full system learning
    -P [rolename], --passwd
            Create password for RBAC administration
            or a special role
    -R, --reload    Reload the RBAC system while in admin mode
                        Reloading will happen atomically, preserving
                        special roles and inherited subjects
    -r, --oldreload Reload the RBAC system using the old method that
```

7 http://en.wikipedia.org/wiki/Man_page

```
                    drops existing special roles and inherited subjects
-L <filename>, --learn
        Specify the pathname for learning logs
-O <filename|directory>, --output
        Specify where to place policies generated from
                    learning mode.  Should be a directory only if
                    "split-roles" is specified in learn_config and
                    full-learning is used.
-M <filename|uid>, --modsegv
        Remove a ban on a specific file or UID
-a <rolename> , --auth
        Authenticates to a special role that requires auth
-u, --unauth    Remove yourself from your current special role
-n <rolename> , --noauth
        Transitions to a special role that doesn't
                    require authentication
-p <rolename> , --pamauth
        Authenticates to a special role through PAM
-V, --verbose   Display verbose policy statistics when enabling system
-h, --help    Display this help
-v, --version    Display version and GPLv2 license information
```

3.3 Learning Mode

The **learning mode** is different than anything found in other security systems. Grsecurity's learning mode can be used on a per-subject or per-role basis, as well as system-wide. When using the learning mode on a single process or role, the rest of the system remains protected as defined by the policy. The learning mode can learn all things that the RBAC system supports: files, capabilities, resources, what IP addresses make use of each role, and socket usage. The learning system performs intelligent reduction of filesystem and network access to reduce policy size, increase readability, and reduce the amount of manual tweaking needed later. Furthermore, the learning system enforces a secure base that is configurable. The */etc/grsec/learn_ config* file gives the administrator the ability to specify files/directories that should be considered protected resources by the learning system. The learning system will ensure that regardless of any rule reduction done, only the processes that access those protected resources through normal usage will be granted access through the generated policy. Furthermore, it will create new subjects for the processes that access the protected resources, creating privilege boundaries that grant those processes additional protection.

3.3.1 Full System Learning

To enable full system learning, run `gradm` as root[8] with the following options:

```
# gradm -F -L /etc/grsec/learning.logs
```

This will enable the Role-based Access Control[9] (RBAC) system and initiate full system learning. That is, `gradm` will monitor and log what your system does. The log can then be used to build a least privilege policy for your system.

8 `http://en.wikipedia.org/wiki/Superuser`

9 Chapter 4 on page 27

Run and use the application(s) that you normally do, several times. This is important, since the learning mode uses a threshold–based system to determine when access should be given to a file or whether it should be given to a directory. If four or more similar accesses are made in a single directory (such as writing to several files in */tmp*), access is granted to that directory instead of the individual files. This reduces the amount of rules you have and ensures that the application will work correctly after the final ACLs are compiled.

> ⚠ **Warning**
>
> Do not perform any administrative tasks outside of the admin role while full system learning is enabled.

To perform administrative tasks while full system learning is enabled, authenticate to the admin role with:

```
# gradm -a admin
```

Remember to exit your shell or unauthenticate from the admin role with `gradm -u` when you are done performing administrative tasks.

Once you feel you've given the system the normal usage it would see in real life, disable the RBAC system with `gradm -D` . Disabling RBAC is a necessary step, as it forces the learning daemon to flush its buffers to disk. Using learning logs obtained before RBAC has been disabled will produce incomplete results. Once RBAC is disabled, execute:

```
# gradm -F -L /etc/grsec/learning.logs -O /etc/grsec/policy
```

This will place the new learned ACLs at the end of your ruleset. You can test the policy by enabling grsecurity (run `gradm -E`), and making sure all applications are functioning the way they're supposed to.

3.3.2 Process and Role-Based Learning

Using this learning mode is very simple. All you have to do is add "l" (the small letter L, not the number 1) to the subject mode of the process, you want to enable learning for. To learn all necessary access for a given binary that does not yet have an established policy, add the following subject:

```
subject /path/of/binary ol
    / h
    -CAP_ALL
    connect disabled
    bind disabled
```

To learn on a given role, add "l" to the role mode. For both of these, to enable learning, enable the system by executing:

```
# gradm -L /etc/grsec/learning.logs -E
```

When you are done, disable the ACL system with `gradm -D` (or alternatively, go into admin mode with `gradm -a`), and use:

```
# gradm -L /etc/grsec/learning.logs -O /etc/grsec/policy
```

This will place the new learned ACLs at the end of your ruleset. Simply remove the old ACLs and you are ready to go.

3.3.3 /etc/grsec/learn_config

This configuration file aids the learning process by tweaking the learning algorithm for specific files and directories. It accepts lines in the form of:

```
<command> <pathname>
```

Where `<command>` can be *inherit-learn* , *no-learn* , *inherit-no-learn* , *high-reduce-path* , *dont-reduce-path* , *protected-path* , *high-protected-path* , and *always-reduce-path* . *inherit-learn* , *no-learn* , and *inherit-no-learn* only affect full system learning, while the others work on all modes of learning.

inherit-learn changes the learning process for the specified path by throwing all learned accesses for every binary executed by the processes contained in the pathname into the subject specified by the pathname. This is useful for `cron` in the case of full system learning, so that scripts that eventually end up executing `mv` or `rm` with privilege don't cause the root policy to grant that privilege to `mv` or `rm` in all cases.

no-learn allows processes within the path to perform any operation that normal system usage would allow without restriction. If a process is generating a huge number of learning logs, it may be best to use this command on that process and configure its policy manually.

inherit-no-learn combines the above two cases, such that processes within the specified path will be able to perform any normal system operation without restriction as will any binaries executed by these processes.

high-reduce-path modifies the heuristics of the learning process to weigh in favor of reducing accesses for this path.

dont-reduce-path modifies the heuristics of the learning process so that it will never reduce accesses for this path.

always-reduce-path modifies the heuristics of the learning process so that the path specified will always have all files and directories within it reduced to the path specified.

protected-path specifies a path on your system that is considered an important resource. Any process that modifies one of these paths is given its own subject in the learning process, facilitating a secure policy.

read-protected-path specifies a path on your system that contains sensitive information. Any process that reads one of these paths is given its own subject in the learning process, facilitating a secure policy.

high-protected-path specifies a path that should be hidden from all processes but those that access it directly. It is recommended to use highly sensitive files for this command.

Note that regular expressions are not supported for pathnames in this configuration file.

This page will introduce you to some additional utilities. They are not required to use a grsecurity-enabled system, but are very useful and thus recommended.

3.4 Controlling PaX Flags (paxctl)

`Paxctl` is a user-space utility for controlling PaX flags of executables (see Appendix/PaX Flags[10] for a list of these flags).

3.4.1 Installation

Download the latest version from the PaX website at `http://pax.grsecurity.net/`. In our case we downloaded *paxctl-0.7.tar.bz2* . `Paxctl` packages are not signed. Change into the directory you downloaded the package to and run the below commands.

```
$ tar xjf paxctl-0.7.tar.bz2
$ cd paxctl-0.7
$ make
$ su
# make install
```

The installation process does the following:

- Installs the `paxctl` program to */sbin* .
- Installs `paxctl` 's man pages to */usr/share/man/man1* .

If you need to change either of these locations, modify the *Makefile* .

3.4.2 Usage

To display all available command-line switches, run `paxctl --help` . Read the man page for more detailed information.

```
# paxctl --help
PaX control v0.7
Copyright 2004,2005,2006,2007,2009,2010,2011,2012 PaX Team
 <pageexec@freemail.hu>

usage: paxctl <options> <files>
```

10 Chapter 8.16 on page 99

```
options:
        -p: disable PAGEEXEC            -P: enable PAGEEXEC
        -e: disable EMUTRAMP            -E: enable EMUTRAMP
        -m: disable MPROTECT            -M: enable MPROTECT
        -r: disable RANDMMAP            -R: enable RANDMMAP
        -x: disable RANDEXEC            -X: enable RANDEXEC
        -s: disable SEGMEXEC            -S: enable SEGMEXEC

        -v: view flags                  -z: restore default flags
        -q: suppress error messages     -Q: report flags in short format
        -c: convert PT_GNU_STACK into PT_PAX_FLAGS (see manpage!)
        -C: create PT_PAX_FLAGS (see manpage!)
```

3.4.3 Examples

Lets query what, if any, PaX flags have been enabled for `/usr/bin/vi` :

```
# paxctl -v /usr/bin/vi
PaX control v0.7
Copyright 2004,2005,2006,2007,2009,2010,2011,2012 PaX Team
 <pageexec@freemail.hu>

file /usr/bin/vi does not have a PT_PAX_FLAGS program header, try conversion
```

As you can see, `paxctl` could not display the flags because `vi` does not have the appropriate program header. We need to convert the header and query the flags again.

> ⚠ **Warning**
>
> Note that `paxctl` does not make backup copies of the files it modifies. It is recommended that you make backups of the binaries you want to modify.

```
# paxctl -c /usr/bin/vi
file /usr/bin/vi had a PT_GNU_STACK program header, converted

# paxctl -v /usr/bin/vi
PaX control v0.7
Copyright 2004,2005,2006,2007,2009,2010,2011,2012 PaX Team
 <pageexec@freemail.hu>

- PaX flags: -------x-e-- [/usr/bin/vi]
        RANDEXEC is disabled
        EMUTRAMP is disabled
```

With the appropriate program header in place, we can query and modify the PaX flags of `vi` .

3.5 Displaying Program Capabilities (pspax)

The `pspax` program displays the run-time capabilities of all programs you have permission for. It is part of the `pax-utils` package. `Pax-utils` can be found at `http://dev.gentoo.`

`org/~vapier/dist/`. It contains many useful tools for PaX[11] but is not as critical as `paxctl` . The `pax-utils` package is maintained by the Hardened Gentoo Project[12].

Programs that the pax-utils package provides:

- `pspax` - Displays the run-time capabilities of all programs you have permission for.
- `scanelf` - Prints out information specific to the ELF[13] structure of a binary.
- `dumpelf` - Converts a ELF file into human readable C code[14] that defines a structure with the same image as the original ELF file.

For more information, see the Gentoo Linux guide to pax-utils[15].

3.5.1 Installation

Gentoo Linux and Debian GNU/Linux[16] users (and possibly others) can install the `pax-utils` package the same way they install any other application in their system. Below are instructions on how to compile and install it from the source.

Download the latest version from `http://dev.gentoo.org/~vapier/dist/`. In our case we downloaded *pax-utils-0.4.tar.xz* , the latest stable release at the time of writing. Change into the directory you downloaded the package to and run the below commands.

```
$ tar xJf pax-utils-0.4.tar.xz
$ cd pax-utils-0.4
$ make
$ su
# make install
```

The installation process does the following:

- Installs the `pspax` , `scanelf` , `dumpelf` and `scanmacho` programs to */usr/bin* .
- Installs README, BUGS, and TODO files to */usr/share/doc/pax-utils/* .
- Installs man pages of `pspax` , `scanelf` and `dumpelf` to */usr/share/man/man1* .

If you need to change any of these locations, modify the *Makefile* .

3.5.2 Usage

To display all available command-line switches, run `pspax --help` . Read the man page for more detailed information.

```
$ pspax --help
* List ELF/PaX information about running processes

Usage: pspax [options]
```

11 http://en.wikipedia.org/wiki/PaX
12 http://www.gentoo.org/proj/en/hardened/
13 http://en.wikipedia.org/wiki/Executable_and_Linkable_Format
14 http://en.wikipedia.org/wiki/C_%28programming_language%29
15 http://www.gentoo.org/proj/en/hardened/pax-utils.xml
16 http://en.wikipedia.org/wiki/Debian

```
Options:
  -a, --all         * Show all processes
  -e, --header      * Print GNU_STACK/PT_LOAD markings
  -i, --ipaddr      * Print ipaddr info if supported
  -p, --pid         * Process ID/pid #
  -u, --user        * Process user/uid #
  -g, --group       * Process group/gid #
  -n, --nx          * Only display w^x processes
  -w, --wx          * Only display w|x processes
  -W, --wide        * Wide output display of cmdline
  -v, --verbose     * Be verbose about executable mappings
  -C, --nocolor     * Don't emit color in output
  -B, --nobanner    * Don't display the header
  -h, --help        * Print this help and exit
  -V, --version     * Print version and exit
```

`Pspax` shows the PaX flags of a single program as a string of characters (e.g. "peMRS"). Lowercase character means the flag is disabled, uppercase means it is enabled. Below is a table that shows these characters and their corresponding PaX flags used by grsecurity. The "Details" column contains a link to a detailed explanation of each flag.

pspax flag	**grsecurity's PaX flag**	**Details**
E	PAX_EMUTRAMP	emutramp.txt[17]
M	PAX_MPROTECT	mprotect.txt[18]
P	PAX_PAGEEXEC	pageexec.txt[19]
R	PAX_RANDMMAP	randmmap.txt[20]
S	PAX_SEGMEXEC	segmexec.txt[21]

3.5.3 Examples

The command `pspax -p <process_id>` displays information about a specific process, identified by its PID. It is unlikely that you happen to know or remember the PID of a process, so it is easier to refer to them by name. The below example uses the `pidof` command to find the PID of a process which it then passes on to `pspax` :

```
# pidof inetd | xargs pspax -p
USER     PID    PAX    MAPS ETYPE      NAME            CAPS_ATTR
root     1741   peMRS  w^x  ET_EXEC    inetd           =ep cap_setpcap-ep
```

17 http://pax.grsecurity.net/docs/emutramp.txt
18 http://pax.grsecurity.net/docs/mprotect.txt
19 http://pax.grsecurity.net/docs/pageexec.txt
20 http://pax.grsecurity.net/docs/randmmap.txt
21 http://pax.grsecurity.net/docs/segmexec.txt

3.6 Managing the Executable Stack of Binaries (execstack)

`Execstack` is a tool to set, clear or query executable stack flag of ELF[22] binaries and shared libraries. It is part of the `prelink`[23] program, but your Linux distribution may provide it as a separate package.

3.6.1 Installation

You are very likely to find the `prelink` and/or `execstack` packages using your distribution's package management system. At least Gentoo, Debian, Red Hat and distributions based on them provide a `prelink` and/or `execstack` packages.

3.6.2 Usage

To display all available command-line switches, run `execstack --help` . Read the man page for more detailed information. Online version of the man page can be found at `http://linux.die.net/man/8/execstack`.

```
# execstack --help
Usage: execstack [OPTION...]
execstack -- program to query or set executable stack flag

  -c, --clear-execstack      Clear executable stack flag bit
  -q, --query                Query executable stack flag bit
  -s, --set-execstack        Set executable stack flag bit
  -?, --help                 Give this help list
      --usage                Give a short usage message
  -V, --version              Print program version

Report bugs to <jakub@redhat.com>.
```

3.6.3 Examples

To check if a library has executable stack enabled, run:

```
# execstack -q /usr/lib/libcrypto.so.0.9.8
- /usr/lib/libcrypto.so.0.9.8
```

The dash means libcrypto does not require an executable stack. If it did, the line would start with a capital "X" instead of a dash.

To query the status of all libraries in your system, run:

```
# find /lib /usr/lib -name '*.so.*.*.*' | xargs execstack
```

22 http://en.wikipedia.org/wiki/Executable_and_Linkable_Format
23 http://en.wikipedia.org/wiki/Prelink

3.7 The sysctl Interface

The sysctl[24] command provides an interface for modifying kernel parameters at runtime. There is an option in the grsecurity kernel configuration to enable support for this interface (see Configuring grsecurity[25]). In Linux, `sysctl` is simply a wrapper around filesystem routines that read and write contents of files in the */proc* directory. This means that you can also set parameters by echoing values to files in */proc* . See the Appendix[26] for a list of all available sysctl options for grsecurity.

3.7.1 Usage

The `sysctl` command takes a list of *variables* or *variable* =*value* pairs and sets or reads their value. Variable is a path to a file in */proc/sys* separated by periods or forward slashes. The value depends on the parameter in question. Most of grsecurity's options are either 1 (enabled) or 0 (disabled).

`Sysctl` 's man page is available online at `http://linux.die.net/man/8/sysctl.`

3.7.2 Examples

If you want to know every available runtime option for grsecurity, list the contents of */proc/sys/kernel/grsecurity* .

To enable mount auditing and disable chdir auditing in a single `sysctl` command, run:

```
# sysctl kernel.grsecurity.audit_mount=1 kernel.grsecurity.audit_chdir=0
kernel.grsecurity.audit_mount = 1
kernel.grsecurity.audit_chdir = 0
```

You can achieve the same result by echoing:

```
# echo 1 > /proc/sys/kernel/grsecurity/audit_mount
# echo 0 > /proc/sys/kernel/grsecurity/audit_chdir
```

24 `http://en.wikipedia.org/wiki/Sysctl`
25 Chapter 2.6.1 on page 13
26 Chapter 8.18 on page 107

4 Policy Configuration

4.1 What Is an RBAC System?

A role-based access control[1] (RBAC) system is an approach to restricting system access to authorized users. You need an RBAC system if you want to restrict access to files, capabilities, resources, or sockets to *all* users, including root[2]. This is similar to a Mandatory Access Control[3] (MAC) model. The other features of grsecurity are only effective at fending off attackers trying to gain root, so the RBAC system is used to fill in this gap. Least privilege can be granted to processes, which, in turn, forces attackers to reevaluate their methods of attack, since gaining access to the root account no longer means that they have full access to the system. Access can be explicitly granted to processes that need it, in such a way that root acts as any other user. Though grsecurity and its RBAC system are in no means perfect security, they greatly increase the difficulty of successfully compromising the system.

In grsecurity, the RBAC system is managed through a **policy** file which is essentially a system-wide set of rules. When the RBAC system is activated with `gradm` , the policy file is parsed and checked for security holes, such as granting the default role access to certain sensitive devices and files like the policy file itself. If a security hole is found, `gradm` will refuse to enable the RBAC system, and will give the user a list of things that need to be fixed. The policy file is protected when the RBAC system is active, and only the admin role may access it during that time. To make it easier to create a secure policy, `gradm` has the ability to learn how the system functions, and build a least-privilege policy based on the collected data (see Learning Mode[4]).

4.2 Limitations of Any Access Control System

So as not to contribute further to the false sense of security many have regarding access control systems (whether they be grsecurity's RBAC, SELinux[5], RSBAC[6], SMACK[7], TOMOYO[8], AppArmor[9], etc.) it's important first to describe the limitations of any access control system.

---

1 `http://en.wikipedia.org/wiki/Role-based_access_control`
2 `http://en.wikipedia.org/wiki/Superuser`
3 `http://en.wikipedia.org/wiki/Mandatory%20Access%20Control`
4 Chapter 3.3 on page 17
5 `http://en.wikipedia.org/wiki/Selinux`
6 `http://en.wikipedia.org/wiki/Rsbac`
7 `http://en.wikipedia.org/wiki/Simplified_Mandatory_Access_Control_Kernel`
8 `http://en.wikipedia.org/wiki/TOMOYO_Linux`
9 `http://en.wikipedia.org/wiki/AppArmor`

There is a fundamental architectural limitation to the kind of guarantees an access control system can provide when the policy decision-making code resides alongside the Operating System's kernel. A compromise of the Operating System can easily result in compromise of the access control system, and it is common practice for exploits which compromise the kernel to disable any active security systems.

Grsecurity is in no way immune to this fundamental limitation, though it does contain several features to help prevent exploitation of the kernel in the first place and furthermore to make the kernel a more hostile environment to an attacker if they do manage to exploit certain types of bugs. The project will continue to make adding similar protections one of its main goals.

Specifically, the following features are involved in kernel self-protection and increasing the difficulty of kernel exploitation:

```
GRKERNSEC_MODHARDEN
GRKERNSEC_HIDESYM
GRKERNSEC_RANDSTRUCT
GRKERNSEC_KSTACKOVERFLOW
PAX_MEMORY_SANITIZE
PAX_MEMORY_UDEREF
PAX_MEMORY_STACKLEAK
PAX_MEMORY_STRUCTLEAK
PAX_CONSTIFY_PLUGIN
PAX_SIZE_OVERFLOW
PAX_KERNEXEC
PAX_RANDKSTACK
PAX_USERCOPY
PAX_REFCOUNT
```

There also exist some features of grsecurity which are always active (and thus have no configure-time option) which aid in the above goals. These include the read-only and non-executable vsyscall page (and its shadow page) on amd64, hardening of the BPF interpreter buffers, and many more.

Though these features have been successful at preventing previous vulnerabilities from being exploited (and surely will continue to do so) there have still been many vulnerabilities it did nothing to prevent exploitation of, and there are entire classes of vulnerabilities (such as missing capability checks, some race conditions, etc.) that it can likely never do anything to prevent exploitation of.

It's partially due to this fundamental limitation of any access control system that grsecurity's RBAC system was designed as it was: to be as automated as possible, to provide a sufficient level of access control, to have easily editable human-readable configurations, and to enforce secure base policies to eliminate some administrator error.

Neither grsecurity's RBAC system nor any other access control system should be used to separate classified information from unclassified information on the same machine. There is no virtual replacement for a physical air-gap.

4.3 Policy Structure

The policy is made up of roles, subjects and objects. **Role** is an abstraction that encompasses traditional users and groups that exist in Linux distributions and special roles, that are specific to grsecurity. **Subjects** are processes or directories, and **objects** are files, capabilities, resources, PaX flags, and IP ACL[10]s. The location of the main policy file is */etc/grsec/policy* .

4.3.1 Policy Structure in a Nutshell

To see a small example policy, look at the default */etc/grsec/policy* file that is installed with `gradm` . In a nutshell, RBAC policies have the following structure:

```
role <role1> <rolemode>
<role attributes>
subject / <subject mode>
<subject attributes>
    / <object mode>
    <extra objects>
    <capability rules>
    <IP ACLs>
    <resource restrictions>
subject <extra subject> <subject mode>
<subject attributes>
    / <object mode>
    <extra objects>
    ...
role <role2> <rolemode>
...
```

Using the default policy as an example:

```
role admin sA
subject / rvka
       / rwcdmlxi

role default G
role_transitions admin
subject /
       /               r
       /opt            rx
       /home           rwxcd
       /mnt            rw
       /dev
       /dev/grsec      h
...
```

10 http://en.wikipedia.org/wiki/Access_control_list

4.4 Rules for Policies

4.4.1 Policy generalization

There exist some features of the RBAC system to aid in simplification and generalization of policies. One of these is the recently added ”replace” rule. The replace rule allows you to assign a string to a variable, and then use that variable within any subject or object pathname to have it replaced with the string. The syntax of replace rules are:

```
replace <variable name> <replace string>
```

So for example:

```
replace CVSROOT /home/cvs
```

The defined variable can then be used as follows:

```
replace CVSROOT /home/cvs
replace PUBHTML public_html

subject $(CVSROOT)/bin/test o
      $(CVSROOT)/grsecurity r
      /home/spender/$(PUBHTML) r
      ...
```

The variables defined with replace rules can be reassigned at any location in the policy. All rules in the policy until another redefinition of the variable will use that new assigned value for the variable. For example:

```
replace CVSROOT /home/cvs
$(CVSROOT)/grsecurity r
replace CVSROOT /var/cvs
$(CVSROOT)/test r
```

would cause the following object rules to be created:

```
/home/cvs/grsecurity r
/var/cvs/test r
```

4.4.2 Special Cases

There are some special cases you should know about when writing policies for the RBAC system.

There exist some unique accesses to filesystem objects that require specific object modes. For instance, a process that connects to a unix domain socket (*/dev/log* for example) will need ”rw” set as the object mode for that socket.

Adding the setgid or setuid flag to a path requires the ”m” object mode.

Creating a hard-link requires at minimum a "cl" object mode. The remaining object flags must match on the target and the source. So for instance, if a process is creating a hard-link from /bin/bash to /bin/bash2, example rules would be:

```
/bin/bash rx
/bin/bash2 rxcl
```

Creating a symlink requires the "wc" object mode.

4.4.3 Wildcarded Objects

One very useful feature of the RBAC system is the support of wildcards in objects. The "*" character matches zero or more characters, "?" matches exactly one character, and "[]" can be used to specify an inclusive or exclusive list or range of characters to match. Depending on how these wildcard characters are used, they have different effects. Here are four examples of the use of wildcards:

```
/dev/tty*      rw
/home/*/bin    rwx
/dev/tty[0-9]  rw
/dev/tty?      rw
```

The first example would match */dev/ttya* , */dev/tty0* , */dev/ttyS0* , etc. Since a '*' at the end of a path can match the '/' character as well, if a '/dev/tty/somefile' path existed, the first example would match it also.

The second example would match */home/user1/bin* , */home/user2/bin* , etc. Note that this rule would not match the path */home/user1/test/bin* as the wildcard characters will not match '/' unless it appears at the end of a path. To use the particular wildcarded object for this example, a */home* object must exist as an "anchor" for the wildcarded object. If you forget to add one, **gradm** will remind you.

The third example would match */dev/tty0* , */dev/tty1* , ... , */dev/tty9* and nothing else.

The fourth example would match */dev/ttya* and */dev/tty0* just like the first example, but would not match */dev/ttyS0* since only one character can match the '?' wildcard.

Wildcards are evaluated at run-time, providing a powerful way of specifying and simplifying policy. Since wildcard matching is based off pathnames and not inode/device pairs though, they aren't intended to be used for objects which are known to be hardlinked at policy enable time.

4.5 Roles

Roles exist essentially as a container for a set of subjects, put to use in specific scenarios. There exist user roles, group roles, a default role, and special roles. See Flow of Matches[11] to see how a role gets matched with a particular process.

11 Chapter 4.12 on page 39

4.5.1 User Roles

In a simplified form, user roles are roles that are automatically applied when a process either is executed by a user of a particular UID or the process changes to that particular UID. In the RBAC system, the name of a user role must match up with the name of an actual user on the system.

A user role looks like:

```
role user1 u
```

4.5.2 Group Roles

As with user roles, group roles pertain to a particular GID. The name of the group role must match up with the name of an actual group on the system. Note that this is tied only to the GID of a process, not to any supplemental groups a process may have. Group roles are applied for a given process only if a user role does not match the process' UID.

A group role looks like:

```
role group1 g
```

4.5.3 Default Role

If neither a user or group role match a given process, then it is assigned the default role. The default role should ideally be a role with nearly no access to the system. It is configured in such a way if full system learning is used.

A default role looks like:

```
role default
```

4.5.4 Special Roles

Special roles are to be used for granting extra privilege to normal user accounts. Some example uses of special roles are to provide an "admin" role that can restart services and edit system configuration files. Special roles can also be provided for regular users to keep their accounts more secure. If they have their own *public_html* directory, the user role for the user could keep this directory read-only, while a special role to which the user is allowed to transition could allow modification of the files in the directory.

Special roles come in two flavors, ones that require authentication, and ones that do not. On the side of special roles that require authentication, the RBAC system supports a flag

that allows PAM authentication to be used for the special role. See Role Modes[12] for a list of all these flags.

Special roles by themselves won't do anything unless there exist non-special (user, group, or default) roles that can transition to them. This transitioning is defined by the **role_transitions** rule, described in the Role Attributes[13] page.

To authenticate to a special role, use `gradm -a <rolename>` . To authenticate with PAM to a special role, use `gradm -p <rolename>` . To transition to a special role that requires no authentication, use `gradm -n <rolename>` .

Special roles look like:

```
role specialauth s

role specialnoauth sN

role specialpamauth sP
```

4.6 Domains

With domains you can combine users that don't share a common group ID as well as groups so that they share a single policy. Domains work just like roles, with the only exception being that the line starting with "role" is replaced with one of the following:

```
domain somedomainname u user1 user2 user3 user4 ... usern
domain somedomainname g group1 group2 group3 group4 ... groupn
```

Example:

```
domain somedomain u daemon bin www-data
subject /
    /    h
```

As it is with user and group roles, all domain members must exist, and if they're not, an error is raised.

4.7 Subjects

Subjects can describe directories, binaries or scripts. Regular expressions are currently not permitted for subjects. The ability to place a subject on a script is unique, as it permits one to grant privilege to a specific script instead of generally to the associated script's interpreter. For this to function properly, make sure the script's interpreter directive does not use *#!/usr/bin/env* but rather the full path to the interpreter.

12 Chapter 8.10 on page 94
13 Chapter 8.10 on page 94

4.8 Capability Restrictions

When no capability restriction rules are used for a given subject, all capabilities that the system grants normally to processes within that subject are allowed to be used. An exception to this is if the subject involved uses policy inheritance. In that case, the capability restrictions would come from the subject(s) being inherited from. Capability rules have the form +CAP_NAME or -CAP_NAME. CAP_ALL is a pseudo-capability meant to describe the entire list of capabilities. It's mainly used to remove all capability usage for a subject, or in conjunction with a small number of rules granting the ability to use individual capabilities. Provided below are some example scenarios of capability restriction usage, along with an explanation of how the policy is interpreted.

Scenario #1: In this scenario, we're removing all capabilities from `su` but CAP_SETUID and CAP_SETGID.

```
...
subject /bin/su o
  ...
  -CAP_ALL
  +CAP_SETUID
  +CAP_SETGID
```

Scenario #2: In this scenario, we're making use of policy inheritance. Note that the default subject allows CAP_NET_BIND_SERVICE and CAP_NET_RAW. In our `ping` subject, we're removing CAP_NET_BIND_SERVICE, but since we're inheriting from the default subject (note the lack of the *o* subject mode on the `ping` subject), we are still allowed CAP_NET_RAW. Granting important capabilities to default subjects is not something allowed by the RBAC system, so this is just an example.

```
...
subject /
  ...
  -CAP_ALL
  +CAP_NET_RAW
  +CAP_NET_BIND_SERVICE
subject /bin/ping
  ...
  -CAP_NET_BIND_SERVICE
```

Auditing and Suppression: Auditing of attempted capability use and suppression of denied capability usage is possible as well. Capability auditing and suppression supports the same policy inheritance rules as normal capability rules. The below example demonstrates auditing the use of CAP_NET_RAW and the suppression of CAP_NET_BIND_SERVICE denials:

```
...
subject /
  ...
  -CAP_ALL
  -CAP_NET_BIND_SERVICE suppress
  +CAP_NET_RAW audit
```

For a full listing of the capabilities available, see: Capability Names and Descriptions[14]. Note that not all of the capabilities listed may be supported by your particular version of the Linux kernel.

4.9 Resource Restrictions

One of the features of grsecurity's ACL system is process–based resource restrictions. Using this feature allows you to restrict things like how much memory a process can take up, how much CPU time, how many files it can open, and how many processes it can execute. Also in this section, we will discuss a "fake" resource implemented in grsecurity's ACL system called "RES_CRASH" that helps guard against bruteforce exploit attempts, which is necessary if you're using PaX.

A single resource rule follows the following syntax:

```
<resource name> <soft limit> <hard limit>
```

An example of this syntax would be:

```
RES_NOFILE 3 3
```

This would allow the process to open a maximum of 3 files (all processes have 3 open file descriptors at some point: stdin (standard input), stdout (standard output), and stderr (standard error output)).

To clarify what the soft limit and hard limit are, the soft limit is the limit assigned to the process when it is run. The hard limit is the maximum point to which a process can raise the limit via `setrlimit(2)` , unless they have CAP_SYS_RESOURCE. In the case of RES_CPU, when the soft limit is overstepped, a special signal is sent to the process continuously. When the hard limit is overstepped, the process is killed.

A person who is less familiar with Linux should stick to setting limits on the number of files, the address space limit, and number of processes. Of course, you can always use the learning mode[15] of grsecurity to set the resource limits for you. The RES_CPU resource is the only one that accepts time as limits. The time defaults to units of milliseconds. You can also append a case sensitive unit to your limit.

Some examples would be:

- 100s – 100 seconds
- 25m – 25 minutes
- 65h – 65 hours
- 2d – 2 days

The other resources either operate on a number itself or on a size, in bytes. For these you can use the following units: K, M, and G, like:

14 Chapter 8.16 on page 99
15 Chapter 3.3 on page 17

- 2G – 2 billion
- 25M – 25 million
- 100K – 100 thousand

If you don't want any restriction for the soft or hard limit for a resource, you can use "unlimited" as the limit. Here are some more examples to help you understand how this works:

```
subject /bin/bash
       /               r
       /opt            rx
       /home           rwxcd
       /mnt            rw
       /dev
       /dev/grsec      h

       RES_CPU 25m 30m
       RLIMIT_AS 5M 5M
       RLIMIT_NPROC 2 2
       RLIMIT_FSIZE 5K 10K
...
```

For a list of accepted resource names and units, see System Resources[16].

4.9.1 RES_CRASH

This "fake" resource limit is expressed by using the name "RES_CRASH" and has the following syntax:

```
RES_CRASH <number of crashes> <amt. of time>
```

For example, if you wanted to allow the program to crash once every 30 minutes, you would use the following:

```
RES_CRASH 1 30m
```

What happens when this threshold is reached? Well, the only way to ensure that the process won't crash again is to keep it from being executed. If the process is a suid/sgid binary run by a regular user, we kill all processes of that regular user and keep them from logging in for the amount of time, specified as the second parameter to the RES_CRASH resource. So for the above example, the user would be locked out of the system for 30 minutes. If the process is not a suid/sguid binary, we simply keep the binary from being run again for the amount of time specified as the second parameter to the RES_CRASH resource, after killing all processes of that binary.

16 Chapter 8.16 on page 105

4.10 Socket Policies

The RBAC system supports policies on what local IP addresses and ports can be reserved on the machine, as well as what remote hosts and ports can be communicated with. These two different accesses are abstracted to *bind* and *connect* rules, respectively. The syntax for the rules is:

```
 connect <IP/host>/<netmask>:<port/portrange> <socket type 1> ... <socket type
n> <proto 1> ... <proto n>
 bind <IP/host>/<netmask>:<port/portrange> <socket type 1> ... <socket type n>
<proto 1> ... <proto n>

 or:

 connect disabled
 bind disabled
```

"proto" can be any of the protocol names listed in */etc/protocol* or "any_proto" to denote any protocol. "socket type" is most commonly "ip", "dgram", or "stream", but can also be "raw_sock", "rdm", or "any_sock" to denote any socket type. Most of the parameters for these rules are optional, particularly the netmask and port or port range. If a port is supplied, then at least an IP address of 0.0.0.0/0 needs to be supplied.

As with capability restrictions, resource restrictions, and many other RBAC features, if the socket policies are omitted for a given subject, then the subject is allowed to bind or connect to anything normally allowed by the system. Note though that if a connect rule is given, then at least one bind rule must also be specified. Older versions of `gradm` (before the 9/16/09 2.1.14 release) will treat the unspecified rule as a "disabled" rule, whereas new versions will generate an error on such policies.

> ⚠ **Warning**
>
> Unlike with file objects and capabilities, policy inheritance has not been implemented for socket policies. Therefore, the socket policies for a given subject are solely determined by that subject alone.

Here are some example rules:

```
subject /usr/bin/ssh o
...
connect 192.168.0.0/24:22 stream tcp
connect ourdnsserver.com:53 dgram udp
```

In this example, `ssh` is allowed to connect to ssh servers anywhere on the class C 192.168.0.X network. It is also allowed to do DNS lookups through the host specified. The hostname is resolved at the time the RBAC system is enabled.

```
subject /usr/bin/nc o
...
bind 0.0.0.0/0:1024-65535 stream tcp
connect 22.22.22.22:5190 stream tcp
```

In this example, `netcat` is allowed to listen on ports 1024 through 65535 on any local interface for TCP connections. It is also able to connect to TCP port 5190 of the 22.22.22.22 host.

```
subject /bin/strange o
...
bind disabled
connect 192.168.1.5:6000-6006 stream tcp
```

This example illustrates how you can have bind disabled but still specify connect rules, or conversely, have connect disabled and only specify bind rules.

As you can see from the examples above, you can have as many socket policies as you wish for a given subject, and as you'll read below there are some powerful extensions to the socket policies.

4.10.1 Per-interface Socket Policies

Rules such as:

```
bind eth1:80 stream tcp
bind eth0#1:22 stream tcp
```

are allowed, giving you the ability to tie specific socket rules to a single interface (or by using the inverted rules mentioned below, all but one interface). Virtual interfaces are specified by the <ifname>#<vindex> syntax. If an interface is specified, no IP/netmask or host may be specified for the rule.

4.10.2 Inverted Socket Policies

Rules such as:

```
connect ! www.google.com:80 stream tcp
```

are allowed, which allows you to specify that a process can connect to anything except to port 80 of www.google.com with a stream TCP socket. The inverted socket matching also works on bind rules.

4.11 PaX Flags

In more recent versions of the RBAC system, PaX flags have been changed from single-letter subject modes to more closely resemble how capabilities are handled within the policy. Therefore, PaX flags can now be fully controlled on or off for any given subject by adding

+PAX_<feature> or -PAX_<feature> within the scope of a subject. For a full listing of the PaX flags available, see: PaX Flags[17].

4.12 Flow of Matches

Each process on the system has a role and a subject attached to it. This section describes how a process is matched to a role and subject, and how matches are calculated against the objects and capabilities they use. Understanding the flow of matches is necessary for manually creating policies.

4.12.1 Role Hierarchy

When determining a role for a process, the RBAC system matches based on the following role hierarchy, from most specific to least specific:

```
user -> group -> default
```

Both user and group roles are permitted to have the `role_allow_ip` attributes. When checking the UID or GID against the user or group role, respectively, the `role_allow_ip` attributes come into play. Imagine the following policy:

```
role user1 u
role_allow_ip 192.168.1.5
...
```

If someone attempted to log in to the machine as user1 from any IP address other than 192.168.1.5, they would not be assigned the user1 role. The matching system would then fall back on trying to find an acceptable group role, or if one could not be found, fall back to the default role.

4.12.2 Subject/Object Hierarchy

Hierarchy for subjects and objects involves matching a most specific pathname over a less specific pathname. So, if a */bin* object exists, and a */bin/ping* object exists, and a process is attempting to read */bin/ping* , the */bin/ping* object would be the one matching. If */bin/su* were being accessed instead, then */bin* would match.

The path from most specific to least specific pathname isn't linear however, particularly in the case of subjects using policy inheritance. Imagine the following policy:

```
role user1 u
  subject /
    / r
    /tmp rwcd
    /usr/bin rx
```

17 Chapter 8.16 on page 99

```
    /root r
    /root/test/blah r
    ...
  subject /usr/bin/specialbin
    /root/test rw
    ...
```

If */root/test/blah* was being accessed by */usr/bin/specialbin* , it would not be able to write to it. The reason for this is that when going from most specific to least specific for a given path (which involves stripping off each trailing path component and attempting a match for the resulting pathname), the matching algorithm will look (in order from most specific to least specific) in each of the subjects the current subject inherits from. In this case, the algorithm saw that no object existed for */root/test/blah* in the */usr/bin/specialbin* subject, so upon checking the subject for */* it found a */root/test/blah* object, thus resulting in the read-only permission.

When going from most specific to least specific, a globbed object such as */home/** is treated as less specific than */home/blah* (if the requested access is for */home/blah*). Globbed objects are matched in the order in which they're listed in the RBAC policy. So in the following example:

```
role user1 u
  subject /
     / r
     /home r
     /home/* r
     /home/test* rw
     ...
```

If a process were accessing */home/testing/somefile* it would only be allowed to read it, since the */home/** rule was listed first. It was likely that the policy writer didn't intend this behavior (because the */home/test** rule would never match) so the */home/test** object should be swapped to the line the */home/** object is on.

4.12.3 Capability Hierarchy

When determining whether a capability is granted or not, the RBAC system works from most specific subject to least specific (in the case of policy inheritance). The first subject along that path that mentions the capability in question is the one that matches. To illustrate:

```
role user1 u
  subject /
    ...
    -CAP_ALL
    +CAP_NET_BIND_SERVICE
  subject /bin
    ...
    -CAP_NET_BIND_SERVICE
  subject /bin/su
    ...
    +CAP_SETUID
    +CAP_SETGID
```

In this example, */bin/su* is able to use only CAP_SETUID and CAP_SETGID. A lookup on CAP_NET_BIND_SERVICE would fall back to the */bin* subject, since */bin/su* inherits from it and did not explicitly list a rule for CAP_NET_BIND_SERVICE. The */bin* subject specifies that CAP_NET_BIND_SERVICE be disallowed. Matching against another capability, CAP_SYS_ADMIN for instance, would end up falling back to the / subject, where it would match -CAP_ALL and be denied.

4.13 Policy Recommendations

Try to remove as many capabilities from default subjects as possible. The more you remove, the closer root comes to acting as a regular user. The more capabilities you remove, however, the more subjects you will have to create for programs that need those capabilities. The RBAC system will enforce that a minimum level of capabilities be removed from all default subjects.

Use full system learning. It will generate a better policy than you would have generated by hand. Make sure you're making full use of the */etc/grsec/learn_config* file to specify the files and directories particular to your system that you want protected. `gradm` will do all the heavy lifting of creating privilege boundaries for processes that access or modify important data.

Administrative programs, such as shutdown or reboot, should require authentication instead of giving everyone the capabilities to run them.

Always inspect your kernel logs. The RBAC system provides a great amount of human-readable information in every kernel log. Of particular importance is what role and subject were assigned to the process causing an alert. If you think that the alert doesn't match up with what you expect from your policy, make sure that the role and subject actually match. If they don't, then you may have issues with a *role_allow_ip* rule that's preventing the proper role from being applied.

Familiarize yourself with Linux's capabilities and what they cover. A full listing of them is available here: Capability Names and Descriptions[18].

Avoid using policy inheritance until you understand fully how it forms the policy for a given subject. Even then, use it sparingly, reserving it generally for cases where a default subject is configured least privilege, with no readable/writable/executable objects and no capabilities.

Wherever possible, avoid granting both write and execute permission to objects. This gives a potential attacker the ability to execute arbitrary code. Similar to how PaX prevents arbitrary code execution within a given process' address space, one of your goals in creating policies is to prevent this on the file system as well.

Be careful using the suppression ('s') object flag, especially when applying it to / to ignore accesses a program does not really need to operate correctly. A change in glibc or another library the subject uses could cause the application to fail in a way that will be difficult to debug (unless your first step is to remove the suppression flag).

18 Chapter 8.16 on page 99

4.14 Sample Policies

Below is the sample policy provided with a `gradm` installation:

```
role admin sA
subject / rvka
        / rwcdmlxi

role default G
role_transitions admin
subject /
        /               r
        /opt            rx
        /home           rwxcd
        /mnt            rw
        /dev
        /dev/grsec      h
        /dev/urandom    r
        /dev/random     r
        /dev/zero       rw
        /dev/input      rw
        /dev/psaux      rw
        /dev/null       rw
        /dev/tty?       rw
        /dev/console    rw
        /dev/tty        rw
        /dev/pts        rw
        /dev/ptmx       rw
        /dev/dsp        rw
        /dev/mixer      rw
        /dev/initctl    rw
        /dev/fd0        r
        /dev/cdrom      r
        /dev/mem        h
        /dev/kmem       h
        /dev/port       h
        /bin            rx
        /sbin           rx
        /lib            rx
        /usr            rx
# compilation of kernel code should be done within the admin role
        /usr/src        h
        /etc            rx
        /proc           rwx
        /proc/slabinfo  h
        /proc/kcore     h
        /proc/modules   h
        /proc/sys       r
        /root           r
        /tmp            rwcd
        /var            rwxcd
        /var/tmp        rwcd
        /var/log        r
# hide the kernel images and modules
        /boot           h
        /lib/modules    h
        /etc/grsec      h
        /etc/ssh        h

# if sshd needs to be restarted, it can be done through the admin role
# restarting sshd should be followed immediately by a gradm -u
        /usr/sbin/sshd

        -CAP_KILL
        -CAP_SYS_TTY_CONFIG
        -CAP_LINUX_IMMUTABLE
```

```
        -CAP_NET_RAW
        -CAP_MKNOD
        -CAP_SYS_ADMIN
        -CAP_SYS_RAWIO
        -CAP_SYS_MODULE
        -CAP_SYS_PTRACE
        -CAP_NET_ADMIN
        -CAP_NET_BIND_SERVICE
        -CAP_NET_RAW
        -CAP_SYS_CHROOT
        -CAP_SYS_BOOT

#       RES_AS 100M 100M

#       connect 192.168.1.0/24:22 stream tcp
#       bind    0.0.0.0 stream dgram tcp udp

# the d flag protects /proc fd and mem entries for sshd
# all daemons should have 'p' in their subject mode to prevent
# an attacker from killing the service (and restarting it with trojaned
# config file or taking the port it reserved to run a trojaned service)

subject /usr/sbin/sshd dpo
        /               h
        /bin/bash       x
        /dev            h
        /dev/log        rw
        /dev/random     r
        /dev/urandom    r
        /dev/null       rw
        /dev/ptmx       rw
        /dev/pts        rw
        /dev/tty        rw
        /dev/tty?       rw
        /etc            r
        /etc/grsec      h
        /home
        /lib            rx
        /root
        /proc           r
        /proc/kcore     h
        /proc/sys       h
        /usr/lib        rx
        /usr/share/zoneinfo r
        /var/log
        /var/mail
        /var/log/lastlog        rw
        /var/log/wtmp           w
        /var/run/sshd
        /var/run/utmp           rw

        -CAP_ALL
        +CAP_CHOWN
        +CAP_SETGID
        +CAP_SETUID
        +CAP_SYS_CHROOT
        +CAP_SYS_RESOURCE
        +CAP_SYS_TTY_CONFIG

subject /usr/X11R6/bin/XFree86
        /dev/mem        rw

        +CAP_SYS_ADMIN
        +CAP_SYS_TTY_CONFIG
        +CAP_SYS_RAWIO

        -PAX_SEGMEXEC
        -PAX_PAGEEXEC
```

```
        -PAX_MPROTECT

subject /usr/bin/ssh
        /etc/ssh/ssh_config r

subject /sbin/klogd
        +CAP_SYS_ADMIN

subject /sbin/syslog-ng
        +CAP_SYS_ADMIN

subject /usr/sbin/cron
        /dev/log rw

subject /bin/login
        /dev/log rw
        /var/log/wtmp w
        /var/log/faillog rwcd

subject /sbin/getty
        /var/log/wtmp w

subject /sbin/init
        /var/log/wtmp w
```

Below is a full user role policy that covers the behavior of `cvs-pserver` when run as the non-root cvs user, providing anonymous read-only CVS repository access.

```
role cvs u
   subject /
      /     h
      -CAP_ALL
      connect   disabled
      bind      disabled

   subject /usr/bin/cvs
      /
      /etc/fstab         r
      /etc/mtab          r
      /etc/passwd        r
      /proc/meminfo      r
      /dev/urandom       r
      /dev/log           rw
      /dev/null          rw
      /home/cvs          r
      /home/cvs/CVSROOT/val-tags  rw
      /home/cvs/CVSROOT/history   ra
      /tmp               rwcd
      /var/lock/cvs      rwcd
      /var/run/.nscd_socket       rw
      /proc/sys/kernel   r
      /var/run
```

Here's all that's needed for an unprivileged sshd account:

```
role sshd u
   subject /
      /  h
      /var/run/sshd r
      -CAP_ALL
      bind disabled
      connect disabled
```

5 Application-specific Settings

This page lists applications that need specific settings to work with grsecurity and PaX. If you wish to add an application to the list, you are most welcome to do so. Please keep the list in alphabetical order and remember to update the table of contents on the front page[1].

5.1 ATI Catalyst (fglrx) graphics driver

When using Xorg and the proprietary ATI Catalyst graphics driver, CONFIG_PAX_USERCOPY must not be set as PAX_USERCOPY prevents a real overflow from occurring in the ATI driver that is still unfixed. This is in addition to what's shown in the section on Xorg below.

As of 11.8, CONFIG_PAX_MEMORY_UDEREF must also be disabled.

5.2 cPanel jailshell

Because cPanel's jailshell needs to mount filesystems (including bind mounts) after chrooting, both chroot_caps (due to needing CAP_SYS_ADMIN) and chroot_deny_mount will need to be disabled. To do this, either disable the respective options in your kernel configuration (CONFIG_GRKERNSEC_CHROOT_CAPS and CONFIG_GRKERNSEC_CHROOT_MOUNT) or disable them in an init script if GRKERNSEC_SYSCTL is enabled. Use the following commands:

```
echo 0 > /proc/sys/kernel/grsecurity/chroot_caps
echo 0 > /proc/sys/kernel/grsecurity/chroot_deny_mount
```

We will be working with cPanel developers to see if the need for this workaround can be avoided in future jailshell versions.

5.3 Firefox (or Iceweasel in Debian)

Mozilla Firefox and possibly all, if not some(?) of, the lib.so files in the folder (/usr/lib/firefox) with the Firefox binary (called /usr/lib/firefox/firefox) need `mprotect`

1 http://en.wikibooks.org/wiki/Grsecurity%23Application-specific_Settings

disabled for flash[2] to function. Without the Firefox binary having disabled `mprotect` Firefox will enter an infinite loop at startup or take minutes to load. Without the lib.so files having `mprotect` disabled any page encountered with Flash will surely run an infinite loop and the Firefox process will have to be killed.

The option must be disabled for just-in-time compilation of certain scripts for both *xulrunner-stub* and *xulrunner-bin* . See Grsecurity forums for more details. `http://forums.grsecurity.net/viewtopic.php?f=3&t=2083` The safest option would of course be denying `mprotect` and boycot sites that use just-in-time (JIT) flash scripts. You may disable JIT compilation in the browser by initiating the address about:config, search for "jit" in the page's integrated search bar, and double-click the options "javascript.options.methodjit.chrome" and "javascript.options.methodjit.content" to set them to "false".

Firefox >= 3.5 may need `RANDMMAP` to be disabled (), if not it will enter in an infinite loop during startup. To disable, execute `paxctl -r /firefox_binary` . Usually the binary is somewhere in */usr/lib64/*firefox** . See `http://bugs.gentoo.org/show_bug.cgi?id=278698` for more details. As of at least Firefox 13 on Ubuntu-based distros you can enable `RANDMMAP` .

5.4 Google Chrome 15.0.874.106

On Google Chrome:

```
  $ paxctl -v /opt/google/chrome/chrome PaX control v0.5 Copyright 2004,2005,2006,2007 PaX Team
<pageexec@freemail.hu> - PaX flags: P----m-x-eR- [/opt/google/chrome/chrome] PAGEEXEC is enabled
MPROTECT is disabled RANDEXEC is disabled EMUTRAMP is disabled RANDMMAP is enabled

$ paxctl -v /opt/google/chrome/nacl_helper PaX control v0.5 Copyright 2004,2005,2006,2007 PaX Team
<pageexec@freemail.hu> - PaX flags: -p---m-x-e-- [/opt/google/chrome/nacl_helper] PAGEEXEC is dis-
abled MPROTECT is disabled RANDEXEC is disabled EMUTRAMP is disabled

$ paxctl -v /opt/google/chrome/chrome-sandbox PaX control v0.5 Copyright 2004,2005,2006,2007 PaX
Team <pageexec@freemail.hu> - PaX flags: -----m-x-e-- [/opt/google/chrome/chrome-sandbox] MPROTECT
is disabled RANDEXEC is disabled EMUTRAMP is disabled
```

These PaX flags work well on my system with flash. Chrome's nacl does throw this however:

```
[1:1:14105440733:ERROR:nacl_fork_delegate_linux.cc(78)] Bad NaCl helper startup ack (0 bytes)
```

5.5 Grub

Grub uses nested functions and thus needs either PAX_EMUTRAMP enabled in the kernel and EMUTRAMP enabled on affected binaries, or if PAX_EMUTRAMP is not enabled in the kernel, needs MPROTECT disabled on affected binaries. Depending on the version of grub in use, some of the following files may not exist, but you should mark

2 `http://en.wikipedia.org/wiki/Adobe_Flash_Player`

all those that exist. To add EMUTRAMP, use the '-CE' argument to paxctl. To remove MPROTECT, use '-Cm'.

```
/usr/bin/grub-script-check
/usr/sbin/grub-probe
/usr/sbin/grub-mkdevicemap
```

5.6 GUFW/UFW firewalls or Update Manager

GUFW is an optional graphical application interface for the Ubuntu firewall (UFW), both of which use Python. Update Manager is a Gnome application for updating packages that also depends on Python. Really, any application that uses Python try enabling EMUTRAMP for the version of Python that is the dependency of your affected program (GUFW or Update Manager). (Example: # paxctl -E /usr/bin/Python2.7).

5.7 IOQuake3

Ioquake3[3] requires disabling mprotect restrictions to run correctly.

5.8 ISC DHCP Server

NOTE: grsecurity patches released as of May 4th, 2014 do not require the below modifications

On some systems, after upgrading to a grsecurity-enabled kernel with GRKERNSEC_PROC_USERGROUP enabled, the kernel log may be spammed with:

```
init: isc-dhcp-server main process ended, respawning
init: isc-dhcp-server main process (pid) terminated with status           1
```

This may be due to unprivileged users not having access to /proc/net/dev as this dhcpd requires. You can confirm by running *dhcpd -f* from the command-line, which should display the following error:

```
Error opening '/proc/net/dev' to list interfaces
```

To fix this, grep your kernel config for CONFIG_GRKERNSEC_PROC_GID, then add a group for that gid to */etc/group* if it doesn't already exist. Then add dhcpd to that group. The added line will look similar to:

3 http://en.wikipedia.org/wiki/Ioquake3

```
procview:x:1001:dhcpd
```

As the DHCP server is continually attempting to respawn, upon making this change you should find it running properly.

5.9 Java

With problems with an epoll stack trace lookup `http://forums.grsecurity.net/viewtopic.php?f=3&t=1983&p=8168&hilit=java#p8168e`. Also there is a problem with just-in-time compilation. Disable `mprotect` for */usr/lib/jvm/java-6-sun-1.6.0.10/jre/bin/java* and */usr/lib/jvm/java-6-sun-1.6.0.10/jre/bin/javaws* .

5.10 Nagios

Nagios needs to be able to view all processes on the system in order to accurately portray service status and performance statistics. It must therefore be run with the group of the CONFIG_GRKERNSEC_PROC_GID you configured, or as set with the `grsec_proc_gid` kernel command-line option.

5.11 Node.js

Node.js needs to execute arbitrary code at runtime. To permit this, `mprotect` needs to be disabled. On most systems, this can be accomplished with the command:

```
paxctl -Cm /usr/bin/nodejs
```

5.12 Openoffice.org

Openoffice.org[4] uses two binaries which need custom settings to work. Both */usr/lib/openoffice/program/soffice.bin* and */usr/lib/openoffice/program/unopkg.bin* need to have unrestricted `mprotect` . `http://forums.grsecurity.net/viewtopic.php?f=3&t=1817`

5.13 libreoffice.org

the same as openoffice.org, but libreoffice.org need to have unrestricted `mprotect` for:

4 `http://en.wikipedia.org/wiki/Openoffice`

/usr/lib/jvm/java-6-openjdk-amd64/jre/bin/java to work if you use libreoffice-base: Database.

5.14 PHP and other applications that set their own resource limits

While Apache/PHP run very well with a grsec/PaX enabled kernel, you could feel like there are possible memory leaks or strange OOM (out of memory) errors with PHP using a PaX enabled kernel with the SEGMEXEC flag enabled. There's no memory leak, and the OOM errors are normal, particularly if you didn't set high enough resource limits.

Concerning "abnormal" memory usage with PHP and SEGMEXEC flag enabled, see spender's answers on `http://bugs.php.net/bug.php?id=49501` comments:

```
"Due to VMA mirroring, the SEGMEXEC option causes accounted vm usage to double.
 So you weren't
experiencing a memory leak -- you were just being accounted for twice as much
memory as you
thought you were using. The solution would be to double the resource limit or,
if your system
is NX-capable and PAE is enabled, use PAGEEXEC."
```

5.15 X.org

X.org might need some specific kernel settings during configuration (depending on the hardware and the drivers used X won't run with non-executable pages (PAX_NOEXEC)). The problem manifested especially in XFree4. Although, recent versions of X.org are known to work with non-executable pages enabled. If you run into problems with X watch your non-executeble settings.

Some users experience mouse freezes when the system load is high. Typically the mouse pointer is reset, but stays in the upper left corner of the screen. This behaviour was found to occur with certain pre-emption settings `http://forums.grsecurity.net/viewtopic.php?f=3&t=2114http://home.coming.dk/index.php/2008/06/24/p815#comments`. It seems to be an interaction between forced-preemption and KERNEXEC. You should be able to re-enable KERNEXEC as long as you disable preemption or use voluntary preemption.

According to the Pax-Team KERNEXEC should work as is, since the changes should be only basic functions like open/close functions. If you should experience problems switch to voluntary or none pre-emption.

6 Reporting Bugs

6.1 Contacts

Submitting bug reports to the proper developer will help get your bug resolved quicker. Though the developers of PaX and grsecurity will forward bug reports to each other, doing so may delay the resolution of your problem.

For bugs within grsecurity features, submit bug reports to: **spender@grsecurity.net** . For bugs within PaX, submit bug reports to **pageexec@freemail.hu** .

Bug reports can also be submitted to the gsecurity forums[1]. (this is the preferred method). The developers monitor RSS feeds of the forums to be able to respond to bug reports quickly.

If possible, avoid submitting bug reports to the grsecurity mailing list, as it is mainly intended for announcements or other important topics.

6.2 Requirements

To be able to reproduce the problem you're experiencing or properly debug it, information will be requested of you depending on the type of bug you are reporting. For any large files that are requested, such as the kernel's *vmlinux* file, please attempt to make these available via a website (you can use a free file uploading service) as they will likely be rejected by the developers' mail servers. Additional information may be requested for debugging purposes (particularly if the problem cannot be reproduced by the developers), but below is specified the minimum requested information.

For any bug you report, please specify the name of the patch you have applied to the kernel. Please also note that the developers only support the latest test patches, as a bug reported in an older patch may have already been fixed in the latest test patch.

A properly submitted bug report that includes the requested information below up-front greatly improves turnaround time for getting your problem solved.

6.2.1 Compilation Errors

A copy of your kernel *.config*

1 `http://forums.grsecurity.net`

6.2.2 Build/Linking Errors

A copy of your kernel *.config*

Your binutils version (`ld --version`)

6.2.3 RBAC Problems

A copy of your kernel *.config*

A copy of your policy file

A listing of the steps performed to produce the problem

6.2.4 Kernel Crashes/Hangs

A copy of your kernel *.config*

Your binutils version (`ld --version`)

A copy of your *vmlinux* file (from the kernel source tree)

A copy of your *bzImage* file (from the */boot* directory)

A copy of your *System.map* file (from the */boot* directory)

The OOPS report, if one exists (take a photo of the screen if you are unable to capture it on disk). Note: we previously required that GRKERNSEC_HIDESYM be disabled for bug reports. This is no longer the case. Any recent grsecurity patch doesn't require GRKERNSEC_HIDESYM to be disabled for symbols to be displayed in OOPs messages.

A description of the machine's hardware (particularly any non-standard hardware)

Information about your Virtual Machine setup (if applicable): preferred execution mode and kernel paravirtualization

Steps required to reproduce the crash (if not before init starts)

7 Appendix

7.1 Appendix Lists

7.2 Introduction

This is a list of all grsecurity and PaX[1] configuration options in the kernel. You can access this same information using the kernel configuration's built-in help. This page contains only the configuration options present in the latest stable grsecurity release[2]. The grsecurity options are available under Security options » Grsecurity.

Each option contains the corresponding kernel configuration symbol (e.g. `GRKERNSEC_FIFO`), all related sysctl[3] variable names if the option is configurable through sysctl, and description of the option.

> This listing was generated February 15, 2014 from the Kconfig file of grsecurity 3.0-3.13.3-201402132113.patch using a script. Manual updates will be lost the next time the content is regenerated.

1 `http://en.wikipedia.org/wiki/PaX`
2 `http://grsecurity.net/download_stable.php`
3 Chapter 3.6.3 on page 24

8 Grsecurity (top level menu)

8.1 Grsecurity

GRKERNSEC

If you say Y here, you will be able to configure many features
that will enhance the security of your system. It is highly
recommended that you say Y here and read through the help
for each option so that you fully understand the features and
can evaluate their usefulness for your machine.

8.2 Configuration Method

8.2.1 Automatic

GRKERNSEC_CONFIG_AUTO

If you choose this configuration method, you'll be able to answer a small
number of simple questions about how you plan to use this kernel.
The settings of grsecurity and PaX will be automatically configured for
the highest commonly-used settings within the provided constraints.

If you require additional configuration, custom changes can still be made
from the "custom configuration" menu.

8.2.2 Custom

GRKERNSEC_CONFIG_CUSTOM

If you choose this configuration method, you'll be able to configure all
grsecurity and PaX settings manually. Via this method, no options are
automatically enabled.

8.3 Usage Type

8.3.1 Server

GRKERNSEC_CONFIG_SERVER

Choose this option if you plan to use this kernel on a server.

8.3.2 Desktop

GRKERNSEC_CONFIG_DESKTOP

Choose this option if you plan to use this kernel on a desktop.

8.4 Virtualization Type

8.4.1 None

GRKERNSEC_CONFIG_VIRT_NONE

Choose this option if this kernel will be run on bare metal.

8.4.2 Guest

GRKERNSEC_CONFIG_VIRT_GUEST

Choose this option if this kernel will be run as a VM guest.

8.4.3 Host

GRKERNSEC_CONFIG_VIRT_HOST

Choose this option if this kernel will be run as a VM host.

8.5 Virtualization Hardware

8.5.1 EPT/RVI Processor Support

GRKERNSEC_CONFIG_VIRT_EPT

Choose this option if your CPU supports the EPT or RVI features of 2nd-gen
hardware virtualization. This allows for additional kernel hardening
 protections
to operate without additional performance impact.

To see if your Intel processor supports EPT, see:
http://ark.intel.com/Products/VirtualizationTechnology
(Most Core i3/5/7 support EPT)

To see if your AMD processor supports RVI, see:
http://support.amd.com/us/kbarticles/Pages/GPU120AMDRVICPUsHyperVWin8.aspx

8.5.2 First-gen/No Hardware Virtualization

GRKERNSEC_CONFIG_VIRT_SOFT

Choose this option if you use an Atom/Pentium/Core 2 processor that either
 doesn't
support hardware virtualization or doesn't support the EPT/RVI extensions.

8.6 Virtualization Software

8.6.1 Xen

GRKERNSEC_CONFIG_VIRT_XEN

Choose this option if this kernel is running as a Xen guest or host.

8.6.2 VMWare

GRKERNSEC_CONFIG_VIRT_VMWARE

Choose this option if this kernel is running as a VMWare guest or host.

8.6.3 KVM

GRKERNSEC_CONFIG_VIRT_KVM

Choose this option if this kernel is running as a KVM guest or host.

8.6.4 VirtualBox

GRKERNSEC_CONFIG_VIRT_VIRTUALBOX

Choose this option if this kernel is running as a VirtualBox guest or host.

8.7 Required Priorities

8.7.1 Performance

GRKERNSEC_CONFIG_PRIORITY_PERF

Choose this option if performance is of highest priority for this deployment
of grsecurity. Features like UDEREF on a 64bit kernel, kernel stack clearing,

```
clearing of structures intended for userland, and freed memory sanitizing will
be disabled.
```

8.7.2 Security

GRKERNSEC_CONFIG_PRIORITY_SECURITY

```
Choose this option if security is of highest priority for this deployment of
grsecurity.  UDEREF, kernel stack clearing, clearing of structures intended
for userland, and freed memory sanitizing will be enabled for this kernel.
In a worst-case scenario, these features can introduce a 20% performance hit
(UDEREF on x64 contributing half of this hit).
```

8.8 Default Special Groups

8.8.1 GID exempted from /proc restrictions

GRKERNSEC_PROC_GID

```
Setting this GID determines which group will be exempted from
grsecurity's /proc restrictions, allowing users of the specified
group  to view network statistics and the existence of other users'
processes on the system.  This GID may also be chosen at boot time
via "grsec_proc_gid=" on the kernel commandline.
```

8.8.2 GID for TPE-untrusted users

GRKERNSEC_TPE_UNTRUSTED_GID

```
Setting this GID determines what group TPE restrictions will be
*enabled* for.  If the sysctl option is enabled, a sysctl option
with name "tpe_gid" is created.
```

8.8.3 GID for TPE-trusted users

GRKERNSEC_TPE_TRUSTED_GID

```
Setting this GID determines what group TPE restrictions will be
*disabled* for.  If the sysctl option is enabled, a sysctl option
with name "tpe_gid" is created.
```

8.8.4 GID for users with kernel-enforced SymlinksIfOwnerMatch

GRKERNSEC_SYMLINKOWN_GID

```
Setting this GID determines what group kernel-enforced
SymlinksIfOwnerMatch will be enabled for.  If the sysctl option
is enabled, a sysctl option with name "symlinkown_gid" is created.
```

8.9 Customize Configuration

8.9.1 PaX

Enable various PaX features

PAX

```
This allows you to enable various PaX features.  PaX adds
intrusion prevention mechanisms to the kernel that reduce
the risks posed by exploitable memory corruption bugs.
```

PaX Control

Support soft mode
PAX_SOFTMODE

```
Enabling this option will allow you to run PaX in soft mode, that
is, PaX features will not be enforced by default, only on executables
marked explicitly.  You must also enable PT_PAX_FLAGS or XATTR_PAX_FLAGS
support as they are the only way to mark executables for soft mode use.

Soft mode can be activated by using the "pax_softmode=1" kernel command
line option on boot.  Furthermore you can control various PaX features
at runtime via the entries in /proc/sys/kernel/pax.
```

Use legacy ELF header marking
PAX_EI_PAX

```
Enabling this option will allow you to control PaX features on
a per executable basis via the 'chpax' utility available at
http://pax.grsecurity.net/.  The control flags will be read from
an otherwise reserved part of the ELF header.  This marking has
numerous drawbacks (no support for soft-mode, toolchain does not
know about the non-standard use of the ELF header) therefore it
has been deprecated in favour of PT_PAX_FLAGS and XATTR_PAX_FLAGS
support.

Note that if you enable PT_PAX_FLAGS or XATTR_PAX_FLAGS marking
support as well, they will override the legacy EI_PAX marks.

If you enable none of the marking options then all applications
will run with PaX enabled on them by default.
```

Use ELF program header marking
PAX_PT_PAX_FLAGS

```
Enabling this option will allow you to control PaX features on
a per executable basis via the 'paxctl' utility available at
http://pax.grsecurity.net/.  The control flags will be read from
a PaX specific ELF program header (PT_PAX_FLAGS).  This marking
has the benefits of supporting both soft mode and being fully
integrated into the toolchain (the binutils patch is available
from http://pax.grsecurity.net).

Note that if you enable the legacy EI_PAX marking support as well,
the EI_PAX marks will be overridden by the PT_PAX_FLAGS marks.

If you enable both PT_PAX_FLAGS and XATTR_PAX_FLAGS support then you
must make sure that the marks are the same if a binary has both marks.

If you enable none of the marking options then all applications
will run with PaX enabled on them by default.
```

Use filesystem extended attributes marking
PAX_XATTR_PAX_FLAGS

```
Enabling this option will allow you to control PaX features on
a per executable basis via the 'setfattr' utility.  The control
flags will be read from the user.pax.flags extended attribute of
the file.  This marking has the benefit of supporting binary-only
applications that self-check themselves (e.g., skype) and would
not tolerate chpax/paxctl changes.  The main drawback is that
extended attributes are not supported by some filesystems (e.g.,
isofs, udf, vfat) so copying files through such filesystems will
lose the extended attributes and these PaX markings.

Note that if you enable the legacy EI_PAX marking support as well,
the EI_PAX marks will be overridden by the XATTR_PAX_FLAGS marks.

If you enable both PT_PAX_FLAGS and XATTR_PAX_FLAGS support then you
must make sure that the marks are the same if a binary has both marks.

If you enable none of the marking options then all applications
will run with PaX enabled on them by default.
```

MAC system integration

```
Mandatory Access Control systems have the option of controlling
PaX flags on a per executable basis, choose the method supported
by your particular system.

- "none": if your MAC system does not interact with PaX,
- "direct": if your MAC system defines pax_set_initial_flags() itself,
- "hook": if your MAC system uses the pax_set_initial_flags_func callback.

NOTE: this option is for developers/integrators only.
```

none
PAX_NO_ACL_FLAGS

direct
PAX_HAVE_ACL_FLAGS

hook
PAX_HOOK_ACL_FLAGS

Non-executable pages

Enforce non-executable pages
PAX_NOEXEC

```
By design some architectures do not allow for protecting memory
pages against execution or even if they do, Linux does not make
use of this feature.  In practice this means that if a page is
readable (such as the stack or heap) it is also executable.

There is a well known exploit technique that makes use of this
fact and a common programming mistake where an attacker can
introduce code of his choice somewhere in the attacked program's
memory (typically the stack or the heap) and then execute it.

If the attacked program was running with different (typically
higher) privileges than that of the attacker, then he can elevate
his own privilege level (e.g. get a root shell, write to files for
which he does not have write access to, etc).

Enabling this option will let you choose from various features
that prevent the injection and execution of 'foreign' code in
a program.

This will also break programs that rely on the old behaviour and
expect that dynamically allocated memory via the malloc() family
of functions is executable (which it is not).  Notable examples
are the XFree86 4.x server, the java runtime and wine.
```

Paging based non-executable pages
PAX_PAGEEXEC

```
This implementation is based on the paging feature of the CPU.
On i386 without hardware non-executable bit support there is a
variable but usually low performance impact, however on Intel's
P4 core based CPUs it is very high so you should not enable this
for kernels meant to be used on such CPUs.

On alpha, avr32, ia64, parisc, sparc, sparc64, x86_64 and i386
with hardware non-executable bit support there is no performance
impact, on ppc the impact is negligible.

Note that several architectures require various emulations due to
badly designed userland ABIs, this will cause a performance impact
but will disappear as soon as userland is fixed. For example, ppc
userland MUST have been built with secure-plt by a recent toolchain.
```

Segmentation based non-executable pages
PAX_SEGMEXEC

```
This implementation is based on the segmentation feature of the
CPU and has a very small performance impact, however applications
will be limited to a 1.5 GB address space instead of the normal
3 GB.
```

Emulate trampolines
PAX_EMUTRAMP

```
There are some programs and libraries that for one reason or
another attempt to execute special small code snippets from
non-executable memory pages.  Most notable examples are the
signal handler return code generated by the kernel itself and
the GCC trampolines.

If you enabled CONFIG_PAX_PAGEEXEC or CONFIG_PAX_SEGMEXEC then
such programs will no longer work under your kernel.

As a remedy you can say Y here and use the 'chpax' or 'paxctl'
utilities to enable trampoline emulation for the affected programs
yet still have the protection provided by the non-executable pages.

On parisc you MUST enable this option and EMUSIGRT as well, otherwise
your system will not even boot.

Alternatively you can say N here and use the 'chpax' or 'paxctl'
utilities to disable CONFIG_PAX_PAGEEXEC and CONFIG_PAX_SEGMEXEC
for the affected files.

NOTE: enabling this feature *may* open up a loophole in the
protection provided by non-executable pages that an attacker
could abuse.  Therefore the best solution is to not have any
files on your system that would require this option.  This can
be achieved by not using libc5 (which relies on the kernel
signal handler return code) and not using or rewriting programs
that make use of the nested function implementation of GCC.
Skilled users can just fix GCC itself so that it implements
nested function calls in a way that does not interfere with PaX.
```

Automatically emulate sigreturn trampolines
PAX_EMUSIGRT

```
Enabling this option will have the kernel automatically detect
and emulate signal return trampolines executing on the stack
that would otherwise lead to task termination.

This solution is intended as a temporary one for users with
legacy versions of libc (libc5, glibc 2.0, uClibc before 0.9.17,
Modula-3 runtime, etc) or executables linked to such, basically
everything that does not specify its own SA_RESTORER function in
normal executable memory like glibc 2.1+ does.

On parisc you MUST enable this option, otherwise your system will
not even boot.

NOTE: this feature cannot be disabled on a per executable basis
and since it *does* open up a loophole in the protection provided
```

```
by non-executable pages, the best solution is to not have any
files on your system that would require this option.
```

Restrict mprotect()
PAX_MPROTECT

```
Enabling this option will prevent programs from
 - changing the executable status of memory pages that were
   not originally created as executable,
 - making read-only executable pages writable again,
 - creating executable pages from anonymous memory,
 - making read-only-after-relocations (RELRO) data pages writable again.

You should say Y here to complete the protection provided by
the enforcement of non-executable pages.

NOTE: you can use the 'chpax' or 'paxctl' utilities to control
this feature on a per file basis.
```

Use legacy/compat protection demoting (read help)
PAX_MPROTECT_COMPAT

```
The current implementation of PAX_MPROTECT denies RWX allocations/mprotects
by sending the proper error code to the application.  For some broken
userland, this can cause problems with Python or other applications.  The
current implementation however allows for applications like clamav to
detect if JIT compilation/execution is allowed and to fall back gracefully
to an interpreter-based mode if it does not.  While we encourage everyone
to use the current implementation as-is and push upstream to fix broken
userland (note that the RWX logging option can assist with this), in some
environments this may not be possible.  Having to disable MPROTECT
completely on certain binaries reduces the security benefit of PaX,
so this option is provided for those environments to revert to the old
behavior.
```

Allow ELF text relocations (read help)
PAX_ELFRELOCS

```
Non-executable pages and mprotect() restrictions are effective
in preventing the introduction of new executable code into an
attacked task's address space.  There remain only two venues
for this kind of attack: if the attacker can execute already
existing code in the attacked task then he can either have it
create and mmap() a file containing his code or have it mmap()
an already existing ELF library that does not have position
independent code in it and use mprotect() on it to make it
writable and copy his code there.  While protecting against
the former approach is beyond PaX, the latter can be prevented
by having only PIC ELF libraries on one's system (which do not
need to relocate their code).  If you are sure this is your case,
as is the case with all modern Linux distributions, then leave
this option disabled.  You should say 'n' here.
```

Allow ELF ET_EXEC text relocations
PAX_ETEXECRELOCS

On some architectures there are incorrectly created applications
that require text relocations and would not work without enabling
this option. If you are an alpha, ia64 or parisc user, you should
enable this option and disable it once you have made sure that
none of your applications need it.

Automatically emulate ELF PLT
PAX_EMUPLT

Enabling this option will have the kernel automatically detect
and emulate the Procedure Linkage Table entries in ELF files.
On some architectures such entries are in writable memory, and
become non-executable leading to task termination. Therefore
it is mandatory that you enable this option on alpha, parisc,
sparc and sparc64, otherwise your system would not even boot.

NOTE: this feature *does* open up a loophole in the protection
provided by the non-executable pages, therefore the proper
solution is to modify the toolchain to produce a PLT that does
not need to be writable.

Emulate old glibc resolver stub
PAX_DLRESOLVE

This option is needed if userland has an old glibc (before 2.4)
that puts a 'save' instruction into the runtime generated resolver
stub that needs special emulation.

Enforce non-executable kernel pages
PAX_KERNEXEC

This is the kernel land equivalent of PAGEEXEC and MPROTECT,
that is, enabling this option will make it harder to inject
and execute 'foreign' code in kernel memory itself.

Return Address Instrumentation Method

Select the method used to instrument function pointer dereferences.
Note that binary modules cannot be instrumented by this approach.

Note that the implementation requires a gcc with plugin support,
i.e., gcc 4.5 or newer. You may need to install the supporting
headers explicitly in addition to the normal gcc package.

bts
PAX_KERNEXEC_PLUGIN_METHOD_BTS

This method is compatible with binary only modules but has
a higher runtime overhead.

bts
PAX_KERNEXEC_PLUGIN_METHOD

Minimum amount of memory reserved for module code
PAX_KERNEXEC_MODULE_TEXT

```
Due to implementation details the kernel must reserve a fixed
amount of memory for runtime allocated code (such as modules)
at compile time that cannot be changed at runtime.  Here you
can specify the minimum amount in MB that will be reserved.
Due to the same implementation details this size will always
be rounded up to the next 2/4 MB boundary (depends on PAE) so
the actually available memory for runtime allocated code will
usually be more than this minimum.

The default 4 MB should be enough for most users but if you have
an excessive number of modules (e.g., most distribution configs
compile many drivers as modules) or use huge modules such as
nvidia's kernel driver, you will need to adjust this amount.
A good rule of thumb is to look at your currently loaded kernel
modules and add up their sizes.
```

Address Space Layout Randomization

Address Space Layout Randomization
PAX_ASLR

```
Many if not most exploit techniques rely on the knowledge of
certain addresses in the attacked program.  The following options
will allow the kernel to apply a certain amount of randomization
to specific parts of the program thereby forcing an attacker to
guess them in most cases.  Any failed guess will most likely crash
the attacked program which allows the kernel to detect such attempts
and react on them.  PaX itself provides no reaction mechanisms,
instead it is strongly encouraged that you make use of Nergal's
segvguard (ftp://ftp.pl.openwall.com/misc/segvguard/) or grsecurity's
(http://www.grsecurity.net/) built-in crash detection features or
develop one yourself.

By saying Y here you can choose to randomize the following areas:
 - top of the task's kernel stack
 - top of the task's userland stack
 - base address for mmap() requests that do not specify one
   (this includes all libraries)
 - base address of the main executable

It is strongly recommended to say Y here as address space layout
randomization has negligible impact on performance yet it provides
a very effective protection.

NOTE: you can use the 'chpax' or 'paxctl' utilities to control
this feature on a per file basis.
```

Randomize kernel stack base
PAX_RANDKSTACK

By saying Y here the kernel will randomize every task's kernel
stack on every system call. This will not only force an attacker
to guess it but also prevent him from making use of possible
leaked information about it.

Since the kernel stack is a rather scarce resource, randomization
may cause unexpected stack overflows, therefore you should very
carefully test your system. Note that once enabled in the kernel
configuration, this feature cannot be disabled on a per file basis.

Randomize user stack base
PAX_RANDUSTACK

By saying Y here the kernel will randomize every task's userland
stack. The randomization is done in two steps where the second
one may apply a big amount of shift to the top of the stack and
cause problems for programs that want to use lots of memory (more
than 2.5 GB if SEGMEXEC is not active, or 1.25 GB when it is).
For this reason the second step can be controlled by 'chpax' or
'paxctl' on a per file basis.

Randomize mmap() base
PAX_RANDMMAP

By saying Y here the kernel will use a randomized base address for
mmap() requests that do not specify one themselves. As a result
all dynamically loaded libraries will appear at random addresses
and therefore be harder to exploit by a technique where an attacker
attempts to execute library code for his purposes (e.g. spawn a
shell from an exploited program that is running at an elevated
privilege level).

Furthermore, if a program is relinked as a dynamic ELF file, its
base address will be randomized as well, completing the full
randomization of the address space layout. Attacking such programs
becomes a guess game. You can find an example of doing this at
http://pax.grsecurity.net/et_dyn.tar.gz and practical samples at
http://www.grsecurity.net/grsec-gcc-specs.tar.gz .

NOTE: you can use the 'chpax' or 'paxctl' utilities to control this
feature on a per file basis.

Miscellaneous hardening features

Sanitize all freed memory
PAX_MEMORY_SANITIZE

By saying Y here the kernel will erase memory pages and slab objects
as soon as they are freed. This in turn reduces the lifetime of data
stored in them, making it less likely that sensitive information such
as passwords, cryptographic secrets, etc stay in memory for too long.

This is especially useful for programs whose runtime is short, long
lived processes and the kernel itself benefit from this as long as
they ensure timely freeing of memory that may hold sensitive
information.

A nice side effect of the sanitization of slab objects is the
reduction of possible info leaks caused by padding bytes within the
leaky structures. Use-after-free bugs for structures containing
pointers can also be detected as dereferencing the sanitized pointer
will generate an access violation.

The tradeoff is performance impact, on a single CPU system kernel
compilation sees a 3% slowdown, other systems and workloads may vary
and you are advised to test this feature on your expected workload
before deploying it.

To reduce the performance penalty by sanitizing pages only, albeit
limiting the effectiveness of this feature at the same time, slab
sanitization can be disabled with the kernel commandline parameter
"pax_sanitize_slab=0".

Note that this feature does not protect data stored in live pages,
e.g., process memory swapped to disk may stay there for a long time.

Sanitize kernel stack
PAX_MEMORY_STACKLEAK

By saying Y here the kernel will erase the kernel stack before it
returns from a system call. This in turn reduces the information
that a kernel stack leak bug can reveal.

Note that such a bug can still leak information that was put on
the stack by the current system call (the one eventually triggering
the bug) but traces of earlier system calls on the kernel stack
cannot leak anymore.

The tradeoff is performance impact: on a single CPU system kernel
compilation sees a 1% slowdown, other systems and workloads may vary
and you are advised to test this feature on your expected workload
before deploying it.

Note that the full feature requires a gcc with plugin support,
i.e., gcc 4.5 or newer. You may need to install the supporting
headers explicitly in addition to the normal gcc package. Using
older gcc versions means that functions with large enough stack
frames may leave uninitialized memory behind that may be exposed
to a later syscall leaking the stack.

Forcibly initialize local variables copied to userland
PAX_MEMORY_STRUCTLEAK

By saying Y here the kernel will zero initialize some local
variables that are going to be copied to userland. This in
turn prevents unintended information leakage from the kernel
stack should later code forget to explicitly set all parts of
the copied variable.

The tradeoff is less performance impact than PAX_MEMORY_STACKLEAK
at a much smaller coverage.

Note that the implementation requires a gcc with plugin support,
i.e., gcc 4.5 or newer. You may need to install the supporting
headers explicitly in addition to the normal gcc package.

Prevent invalid userland pointer dereference
PAX_MEMORY_UDEREF

```
By saying Y here the kernel will be prevented from dereferencing
userland pointers in contexts where the kernel expects only kernel
pointers.  This is both a useful runtime debugging feature and a
security measure that prevents exploiting a class of kernel bugs.

The tradeoff is that some virtualization solutions may experience
a huge slowdown and therefore you should not enable this feature
for kernels meant to run in such environments.  Whether a given VM
solution is affected or not is best determined by simply trying it
out, the performance impact will be obvious right on boot as this
mechanism engages from very early on.  A good rule of thumb is that
VMs running on CPUs without hardware virtualization support (i.e.,
the majority of IA-32 CPUs) will likely experience the slowdown.

On X86_64 the kernel will make use of PCID support when available
(Intel's Westmere, Sandy Bridge, etc) for better security (default)
or performance impact.  Pass pax_weakuderef on the kernel command
line to choose the latter.
```

Prevent various kernel object reference counter overflows
PAX_REFCOUNT

```
By saying Y here the kernel will detect and prevent overflowing
various (but not all) kinds of object reference counters.  Such
overflows can normally occur due to bugs only and are often, if
not always, exploitable.

The tradeoff is that data structures protected by an overflowed
refcount will never be freed and therefore will leak memory.  Note
that this leak also happens even without this protection but in
that case the overflow can eventually trigger the freeing of the
data structure while it is still being used elsewhere, resulting
in the exploitable situation that this feature prevents.

Since this has a negligible performance impact, you should enable
this feature.
```

Automatically constify eligible structures
PAX_CONSTIFY_PLUGIN

```
By saying Y here the compiler will automatically constify a class
of types that contain only function pointers.  This reduces the
kernel's attack surface and also produces a better memory layout.

Note that the implementation requires a gcc with plugin support,
i.e., gcc 4.5 or newer.  You may need to install the supporting
headers explicitly in addition to the normal gcc package.

Note that if some code really has to modify constified variables
then the source code will have to be patched to allow it.  Examples
can be found in PaX itself (the no_const attribute) and for some
out-of-tree modules at http://www.grsecurity.net/~paxguy1/ .
```

Harden heap object copies between kernel and userland
PAX_USERCOPY

```
By saying Y here the kernel will enforce the size of heap objects
when they are copied in either direction between the kernel and
userland, even if only a part of the heap object is copied.

Specifically, this checking prevents information leaking from the
kernel heap during kernel to userland copies (if the kernel heap
object is otherwise fully initialized) and prevents kernel heap
overflows during userland to kernel copies.

Note that the current implementation provides the strictest bounds
checks for the SLUB allocator.

Enabling this option also enables per-slab cache protection against
data in a given cache being copied into/out of via userland
accessors.  Though the whitelist of regions will be reduced over
time, it notably protects important data structures like task structs.

If frame pointers are enabled on x86, this option will also restrict
copies into and out of the kernel stack to local variables within a
single frame.

Since this has a negligible performance impact, you should enable
this feature.
```

Prevent various integer overflows in function size parameters
PAX_SIZE_OVERFLOW

```
By saying Y here the kernel recomputes expressions of function
arguments marked by a size_overflow attribute with double integer
precision (DImode/TImode for 32/64 bit integer types).

The recomputed argument is checked against TYPE_MAX and an event
is logged on overflow and the triggering process is killed.

Homepage: http://www.grsecurity.net/~ephox/overflow_plugin/

Note that the implementation requires a gcc with plugin support,
i.e., gcc 4.5 or newer.  You may need to install the supporting
headers explicitly in addition to the normal gcc package.
```

Generate some entropy during boot and runtime
PAX_LATENT_ENTROPY

```
By saying Y here the kernel will instrument some kernel code to
extract some entropy from both original and artificially created
program state.  This will help especially embedded systems where
there is little 'natural' source of entropy normally.  The cost
is some slowdown of the boot process and fork and irq processing.

When pax_extra_latent_entropy is passed on the kernel command line,
entropy will be extracted from up to the first 4GB of RAM while the
runtime memory allocator is being initialized.  This costs even more
slowdown of the boot process.

Note that the implementation requires a gcc with plugin support,
i.e., gcc 4.5 or newer.  You may need to install the supporting
headers explicitly in addition to the normal gcc package.
```

```
Note that entropy extracted this way is not cryptographically
secure!
```

8.9.2 Memory Protections

Deny reading/writing to /dev/kmem, /dev/mem, and /dev/port

GRKERNSEC_KMEM

```
If you say Y here, /dev/kmem and /dev/mem won't be allowed to
be written to or read from to modify or leak the contents of the running
kernel.  /dev/port will also not be allowed to be opened, and support
for /dev/cpu/*/msr and kexec will be removed.  If you have module
support disabled, enabling this will close up six ways that are
currently used to insert malicious code into the running kernel.

Even with this feature enabled, we still highly recommend that
you use the RBAC system, as it is still possible for an attacker to
modify the running kernel through other more obscure methods.

Enabling this feature will prevent the "cpupower" and "powertop" tools
from working.

It is highly recommended that you say Y here if you meet all the
conditions above.
```

Restrict VM86 mode

GRKERNSEC_VM86

```
If you say Y here, only processes with CAP_SYS_RAWIO will be able to
make use of a special execution mode on 32bit x86 processors called
Virtual 8086 (VM86) mode.  XFree86 may need vm86 mode for certain
video cards and will still work with this option enabled.  The purpose
of the option is to prevent exploitation of emulation errors in
virtualization of vm86 mode like the one discovered in VMWare in 2009.
Nearly all users should be able to enable this option.
```

Disable privileged I/O

GRKERNSEC_IO
Related sysctl variables:

```
 kernel.grsecurity.disable_priv_io
```

```
If you say Y here, all ioperm and iopl calls will return an error.
Ioperm and iopl can be used to modify the running kernel.
Unfortunately, some programs need this access to operate properly,
the most notable of which are XFree86 and hwclock.  hwclock can be
remedied by having RTC support in the kernel, so real-time
clock support is enabled if this option is enabled, to ensure
that hwclock operates correctly.

If you're using XFree86 or a version of Xorg from 2012 or earlier,
```

```
you may not be able to boot into a graphical environment with this
option enabled.  In this case, you should use the RBAC system instead.
```

Harden BPF JIT against spray attacks

GRKERNSEC_JIT_HARDEN

```
If you say Y here, the native code generated by the kernel's Berkeley
Packet Filter (BPF) JIT engine will be hardened against JIT-spraying
attacks that attempt to fit attacker-beneficial instructions in
32bit immediate fields of JIT-generated native instructions.  The
attacker will generally aim to cause an unintended instruction sequence
of JIT-generated native code to execute by jumping into the middle of
a generated instruction.  This feature effectively randomizes the 32bit
immediate constants present in the generated code to thwart such attacks.

If you're using KERNEXEC, it's recommended that you enable this option
to supplement the hardening of the kernel.
```

Disable unprivileged PERF_EVENTS usage by default

GRKERNSEC_PERF_HARDEN

```
If you say Y here, the range of acceptable values for the
/proc/sys/kernel/perf_event_paranoid sysctl will be expanded to allow and
default to a new value: 3.  When the sysctl is set to this value, no
unprivileged use of the PERF_EVENTS syscall interface will be permitted.

Though PERF_EVENTS can be used legitimately for performance monitoring
and low-level application profiling, it is forced on regardless of
configuration, has been at fault for several vulnerabilities, and
creates new opportunities for side channels and other information leaks.

This feature puts PERF_EVENTS into a secure default state and permits
the administrator to change out of it temporarily if unprivileged
application profiling is needed.
```

Insert random gaps between thread stacks

GRKERNSEC_RAND_THREADSTACK

```
If you say Y here, a random-sized gap will be enforced between allocated
thread stacks.  Glibc's NPTL and other threading libraries that
pass MAP_STACK to the kernel for thread stack allocation are supported.
The implementation currently provides 8 bits of entropy for the gap.

Many distributions do not compile threaded remote services with the
-fstack-check argument to GCC, causing the variable-sized stack-based
allocator, alloca(), to not probe the stack on allocation.  This
permits an unbounded alloca() to skip over any guard page and potentially
modify another thread's stack reliably.  An enforced random gap
reduces the reliability of such an attack and increases the chance
that such a read/write to another thread's stack instead lands in
an unmapped area, causing a crash and triggering grsecurity's
anti-bruteforcing logic.
```

Harden ASLR against information leaks and entropy reduction

GRKERNSEC_PROC_MEMMAP

```
If you say Y here, the /proc/<pid>/maps and /proc/<pid>/stat files will
give no information about the addresses of its mappings if
PaX features that rely on random addresses are enabled on the task.
In addition to sanitizing this information and disabling other
dangerous sources of information, this option causes reads of sensitive
/proc/<pid> entries where the file descriptor was opened in a different
task than the one performing the read.  Such attempts are logged.
This option also limits argv/env strings for suid/sgid binaries
to 512KB to prevent a complete exhaustion of the stack entropy provided
by ASLR.  Finally, it places an 8MB stack resource limit on suid/sgid
binaries to prevent alternative mmap layouts from being abused.

If you use PaX it is essential that you say Y here as it closes up
several holes that make full ASLR useless locally.
```

Prevent kernel stack overflows

GRKERNSEC_KSTACKOVERFLOW

```
If you say Y here, the kernel's process stacks will be allocated
with vmalloc instead of the kernel's default allocator.  This
introduces guard pages that in combination with the alloca checking
of the STACKLEAK feature prevents all forms of kernel process stack
overflow abuse.  Note that this is different from kernel stack
buffer overflows.
```

Deter exploit bruteforcing

GRKERNSEC_BRUTE
Related sysctl variables:

```
 kernel.grsecurity.deter_bruteforce
```

```
If you say Y here, attempts to bruteforce exploits against forking
daemons such as apache or sshd, as well as against suid/sgid binaries
will be deterred.  When a child of a forking daemon is killed by PaX
or crashes due to an illegal instruction or other suspicious signal,
the parent process will be delayed 30 seconds upon every subsequent
fork until the administrator is able to assess the situation and
restart the daemon.
In the suid/sgid case, the attempt is logged, the user has all their
existing instances of the suid/sgid binary terminated and will
be unable to execute any suid/sgid binaries for 15 minutes.

It is recommended that you also enable signal logging in the auditing
section so that logs are generated when a process triggers a suspicious
signal.
If the sysctl option is enabled, a sysctl option with name
"deter_bruteforce" is created.
```

Harden module auto-loading

GRKERNSEC_MODHARDEN

```
If you say Y here, module auto-loading in response to use of some
feature implemented by an unloaded module will be restricted to
root users.  Enabling this option helps defend against attacks
by unprivileged users who abuse the auto-loading behavior to
cause a vulnerable module to load that is then exploited.

If this option prevents a legitimate use of auto-loading for a
non-root user, the administrator can execute modprobe manually
with the exact name of the module mentioned in the alert log.
Alternatively, the administrator can add the module to the list
of modules loaded at boot by modifying init scripts.

Modification of init scripts will most likely be needed on
Ubuntu servers with encrypted home directory support enabled,
as the first non-root user logging in will cause the ecb(aes),
ecb(aes)-all, cbc(aes), and cbc(aes)-all  modules to be loaded.
```

Hide kernel symbols

GRKERNSEC_HIDESYM

```
If you say Y here, getting information on loaded modules, and
displaying all kernel symbols through a syscall will be restricted
to users with CAP_SYS_MODULE.  For software compatibility reasons,
/proc/kallsyms will be restricted to the root user.  The RBAC
system can hide that entry even from root.

This option also prevents leaking of kernel addresses through
several /proc entries.

Note that this option is only effective provided the following
conditions are met:
1) The kernel using grsecurity is not precompiled by some distribution
2) You have also enabled GRKERNSEC_DMESG
3) You are using the RBAC system and hiding other files such as your
   kernel image and System.map.  Alternatively, enabling this option
   causes the permissions on /boot, /lib/modules, and the kernel
   source directory to change at compile time to prevent
   reading by non-root users.
If the above conditions are met, this option will aid in providing a
useful protection against local kernel exploitation of overflows
and arbitrary read/write vulnerabilities.

It is highly recommended that you enable GRKERNSEC_PERF_HARDEN
in addition to this feature.
```

Randomize layout of sensitive kernel structures

GRKERNSEC_RANDSTRUCT

```
If you say Y here, the layouts of a number of sensitive kernel
structures (task, fs, cred, etc) and all structures composed entirely
of function pointers (aka "ops" structs) will be randomized at compile-time.
This can introduce the requirement of an additional infoleak
```

```
vulnerability for exploits targeting these structure types.

Enabling this feature will introduce some performance impact, slightly
increase memory usage, and prevent the use of forensic tools like
Volatility against the system (unless the kernel source tree isn't
cleaned after kernel installation).

The seed used for compilation is located at tools/gcc/randomize_layout_seed.h.
It remains after a make clean to allow for external modules to be compiled
with the existing seed and will be removed by a make mrproper or
make distclean.
```

Use cacheline-aware structure randomization

GRKERNSEC_RANDSTRUCT_PERFORMANCE

```
If you say Y here, the RANDSTRUCT randomization will make a best effort
at restricting randomization to cacheline-sized groups of elements.  It
will further not randomize bitfields in structures.  This reduces the
performance hit of RANDSTRUCT at the cost of weakened randomization.
```

Active kernel exploit response

GRKERNSEC_KERN_LOCKOUT

```
If you say Y here, when a PaX alert is triggered due to suspicious
activity in the kernel (from KERNEXEC/UDEREF/USERCOPY)
or an OOPS occurs due to bad memory accesses, instead of just
terminating the offending process (and potentially allowing
a subsequent exploit from the same user), we will take one of two
actions:
 If the user was root, we will panic the system
 If the user was non-root, we will log the attempt, terminate
 all processes owned by the user, then prevent them from creating
 any new processes until the system is restarted
This deters repeated kernel exploitation/bruteforcing attempts
and is useful for later forensics.
```

Old ARM userland compatibility

GRKERNSEC_OLD_ARM_USERLAND

```
If you say Y here, stubs of executable code to perform such operations
as "compare-exchange" will be placed at fixed locations in the ARM vector
table.  This is unfortunately needed for old ARM userland meant to run
across a wide range of processors.  Without this option enabled,
the get_tls and data memory barrier stubs will be emulated by the kernel,
which is enough for Linaro userlands or other userlands designed for v6
and newer ARM CPUs.  It's recommended that you try without this option enabled
first, and only enable it if your userland does not boot (it will likely fail
at init time).
```

8.9.3 Role Based Access Control Options

Disable RBAC system

GRKERNSEC_NO_RBAC

```
If you say Y here, the /dev/grsec device will be removed from the kernel,
preventing the RBAC system from being enabled.  You should only say Y
here if you have no intention of using the RBAC system, so as to prevent
an attacker with root access from misusing the RBAC system to hide files
and processes when loadable module support and /dev/[k]mem have been
locked down.
```

Hide kernel processes

GRKERNSEC_ACL_HIDEKERN

```
If you say Y here, all kernel threads will be hidden to all
processes but those whose subject has the "view hidden processes"
flag.
```

Maximum tries before password lockout

GRKERNSEC_ACL_MAXTRIES

```
This option enforces the maximum number of times a user can attempt
to authorize themselves with the grsecurity RBAC system before being
denied the ability to attempt authorization again for a specified time.
The lower the number, the harder it will be to brute-force a password.
```

Time to wait after max password tries, in seconds

GRKERNSEC_ACL_TIMEOUT

```
This option specifies the time the user must wait after attempting to
authorize to the RBAC system with the maximum number of invalid
passwords.  The higher the number, the harder it will be to brute-force
a password.
```

8.9.4 Filesystem Protections

Proc restrictions

GRKERNSEC_PROC

```
If you say Y here, the permissions of the /proc filesystem
will be altered to enhance system security and privacy.  You MUST
choose either a user only restriction or a user and group restriction.
Depending upon the option you choose, you can either restrict users to
```

```
see only the processes they themselves run, or choose a group that can
view all processes and files normally restricted to root if you choose
the "restrict to user only" option.  NOTE: If you're running identd or
ntpd as a non-root user, you will have to run it as the group you
specify here.
```

Restrict /proc to user only

GRKERNSEC_PROC_USER

```
If you say Y here, non-root users will only be able to view their own
processes, and restricts them from viewing network-related information,
and viewing kernel symbol and module information.
```

Allow special group

GRKERNSEC_PROC_USERGROUP

```
If you say Y here, you will be able to select a group that will be
able to view all processes and network-related information.  If you've
enabled GRKERNSEC_HIDESYM, kernel and symbol information may still
remain hidden.  This option is useful if you want to run identd as
a non-root user.  The group you select may also be chosen at boot time
via "grsec_proc_gid=" on the kernel commandline.
```

GID exempted from /proc restrictions

GRKERNSEC_PROC_GID

```
Setting this GID determines which group will be exempted from
grsecurity's /proc restrictions, allowing users of the specified
group  to view network statistics and the existence of other users'
processes on the system.  This GID may also be chosen at boot time
via "grsec_proc_gid=" on the kernel commandline.
```

Additional restrictions

GRKERNSEC_PROC_ADD

```
If you say Y here, additional restrictions will be placed on
/proc that keep normal users from viewing device information and
slabinfo information that could be useful for exploits.
```

Linking restrictions

GRKERNSEC_LINK
Related sysctl variables:

kernel.grsecurity.linking_restrictions

```
If you say Y here, /tmp race exploits will be prevented, since users
will no longer be able to follow symlinks owned by other users in
world-writable +t directories (e.g. /tmp), unless the owner of the
symlink is the owner of the directory. users will also not be
able to hardlink to files they do not own.  If the sysctl option is
enabled, a sysctl option with name "linking_restrictions" is created.
```

Kernel-enforced SymlinksIfOwnerMatch

GRKERNSEC_SYMLINKOWN
Related sysctl variables:

kernel.grsecurity.enforce_symlinksifowner

kernel.grsecurity.symlinkown_gid

```
Apache's SymlinksIfOwnerMatch option has an inherent race condition
that prevents it from being used as a security feature.  As Apache
verifies the symlink by performing a stat() against the target of
the symlink before it is followed, an attacker can setup a symlink
to point to a same-owned file, then replace the symlink with one
that targets another user's file just after Apache "validates" the
symlink -- a classic TOCTOU race.  If you say Y here, a complete,
race-free replacement for Apache's "SymlinksIfOwnerMatch" option
will be in place for the group you specify. If the sysctl option
is enabled, a sysctl option with name "enforce_symlinksifowner" is
created.
```

GID for users with kernel-enforced SymlinksIfOwnerMatch

GRKERNSEC_SYMLINKOWN_GID

```
Setting this GID determines what group kernel-enforced
SymlinksIfOwnerMatch will be enabled for.  If the sysctl option
is enabled, a sysctl option with name "symlinkown_gid" is created.
```

FIFO restrictions

GRKERNSEC_FIFO
Related sysctl variables:

kernel.grsecurity.fifo_restrictions

```
If you say Y here, users will not be able to write to FIFOs they don't
own in world-writable +t directories (e.g. /tmp), unless the owner of
the FIFO is the same owner of the directory it's held in.  If the sysctl
option is enabled, a sysctl option with name "fifo_restrictions" is
created.
```

Sysfs/debugfs restriction

GRKERNSEC_SYSFS_RESTRICT

```
If you say Y here, sysfs (the pseudo-filesystem mounted at /sys) and
any filesystem normally mounted under it (e.g. debugfs) will be
mostly accessible only by root.  These filesystems generally provide access
to hardware and debug information that isn't appropriate for unprivileged
users of the system.  Sysfs and debugfs have also become a large source
of new vulnerabilities, ranging from infoleaks to local compromise.
There has been very little oversight with an eye toward security involved
in adding new exporters of information to these filesystems, so their
use is discouraged.
For reasons of compatibility, a few directories have been whitelisted
for access by non-root users:
/sys/fs/selinux
/sys/fs/fuse
/sys/devices/system/cpu
```

Runtime read-only mount protection

GRKERNSEC_ROFS
Related sysctl variables:

```
 kernel.grsecurity.romount_protect
```

```
If you say Y here, a sysctl option with name "romount_protect" will
be created.  By setting this option to 1 at runtime, filesystems
will be protected in the following ways:
* No new writable mounts will be allowed
* Existing read-only mounts won't be able to be remounted read/write
* Write operations will be denied on all block devices
This option acts independently of grsec_lock: once it is set to 1,
it cannot be turned off.  Therefore, please be mindful of the resulting
behavior if this option is enabled in an init script on a read-only
filesystem.
Also be aware that as with other root-focused features, GRKERNSEC_KMEM
and GRKERNSEC_IO should be enabled and module loading disabled via
config or at runtime.
This feature is mainly intended for secure embedded systems.
```

Eliminate stat/notify-based device sidechannels

GRKERNSEC_DEVICE_SIDECHANNEL

```
If you say Y here, timing analyses on block or character
devices like /dev/ptmx using stat or inotify/dnotify/fanotify
will be thwarted for unprivileged users.  If a process without
CAP_MKNOD stats such a device, the last access and last modify times
will match the device's create time.  No access or modify events
will be triggered through inotify/dnotify/fanotify for such devices.
This feature will prevent attacks that may at a minimum
allow an attacker to determine the administrator's password length.
```

Chroot jail restrictions

GRKERNSEC_CHROOT

```
If you say Y here, you will be able to choose several options that will
make breaking out of a chrooted jail much more difficult.  If you
encounter no software incompatibilities with the following options, it
is recommended that you enable each one.
```

Deny mounts

GRKERNSEC_CHROOT_MOUNT
Related sysctl variables:

```
kernel.grsecurity.chroot_deny_mount
```

```
If you say Y here, processes inside a chroot will not be able to
mount or remount filesystems.  If the sysctl option is enabled, a
sysctl option with name "chroot_deny_mount" is created.
```

Deny double-chroots

GRKERNSEC_CHROOT_DOUBLE
Related sysctl variables:

```
kernel.grsecurity.chroot_deny_chroot
```

```
If you say Y here, processes inside a chroot will not be able to chroot
again outside the chroot.  This is a widely used method of breaking
out of a chroot jail and should not be allowed.  If the sysctl
option is enabled, a sysctl option with name
"chroot_deny_chroot" is created.
```

Deny pivot_root in chroot

GRKERNSEC_CHROOT_PIVOT
Related sysctl variables:

```
kernel.grsecurity.chroot_deny_pivot
```

```
If you say Y here, processes inside a chroot will not be able to use
a function called pivot_root() that was introduced in Linux 2.3.41.  It
works similar to chroot in that it changes the root filesystem.  This
function could be misused in a chrooted process to attempt to break out
of the chroot, and therefore should not be allowed.  If the sysctl
option is enabled, a sysctl option with name "chroot_deny_pivot" is
created.
```

Enforce chdir("/")

GRKERNSEC_CHROOT_CHDIR
Related sysctl variables:

kernel.grsecurity.chroot_enforce_chdir

```
If you say Y here, the current working directory of all newly-chrooted
applications will be set to the the root directory of the chroot.
The man page on chroot(2) states:
Note that usually chhroot does not change  the  current  working
directory,  so  that `.' can be outside the tree rooted at
`/'.  In particular, the  super-user  can  escape  from  a
`chroot jail' by doing `mkdir foo; chroot foo; cd ..'.

It is recommended that you say Y here, since it's not known to break
any software.  If the sysctl option is enabled, a sysctl option with
name "chroot_enforce_chdir" is created.
```

Deny (f)chmod +s

GRKERNSEC_CHROOT_CHMOD
Related sysctl variables:

kernel.grsecurity.chroot_deny_chmod

```
If you say Y here, processes inside a chroot will not be able to chmod
or fchmod files to make them have suid or sgid bits.  This protects
against another published method of breaking a chroot.  If the sysctl
option is enabled, a sysctl option with name "chroot_deny_chmod" is
created.
```

Deny fchdir out of chroot

GRKERNSEC_CHROOT_FCHDIR
Related sysctl variables:

kernel.grsecurity.chroot_deny_fchdir

```
If you say Y here, a well-known method of breaking chroots by fchdir'ing
to a file descriptor of the chrooting process that points to a directory
outside the filesystem will be stopped.  If the sysctl option
is enabled, a sysctl option with name "chroot_deny_fchdir" is created.
```

Deny mknod

GRKERNSEC_CHROOT_MKNOD
Related sysctl variables:

kernel.grsecurity.chroot_deny_mknod

```
If you say Y here, processes inside a chroot will not be allowed to
mknod.  The problem with using mknod inside a chroot is that it
would allow an attacker to create a device entry that is the same
as one on the physical root of your system, which could range from
anything from the console device to a device for your harddrive (which
they could then use to wipe the drive or steal data).  It is recommended
that you say Y here, unless you run into software incompatibilities.
If the sysctl option is enabled, a sysctl option with name
"chroot_deny_mknod" is created.
```

Deny shmat() out of chroot

GRKERNSEC_CHROOT_SHMAT
Related sysctl variables:

kernel.grsecurity.chroot_deny_shmat

```
If you say Y here, processes inside a chroot will not be able to attach
to shared memory segments that were created outside of the chroot jail.
It is recommended that you say Y here.  If the sysctl option is enabled,
a sysctl option with name "chroot_deny_shmat" is created.
```

Deny access to abstract AF_UNIX sockets out of chroot

GRKERNSEC_CHROOT_UNIX
Related sysctl variables:

kernel.grsecurity.chroot_deny_unix

```
If you say Y here, processes inside a chroot will not be able to
connect to abstract (meaning not belonging to a filesystem) Unix
domain sockets that were bound outside of a chroot.  It is recommended
that you say Y here.  If the sysctl option is enabled, a sysctl option
with name "chroot_deny_unix" is created.
```

Protect outside processes

GRKERNSEC_CHROOT_FINDTASK
Related sysctl variables:

kernel.grsecurity.chroot_findtask

```
If you say Y here, processes inside a chroot will not be able to
kill, send signals with fcntl, ptrace, capget, getpgid, setpgid,
getsid, or view any process outside of the chroot.  If the sysctl
option is enabled, a sysctl option with name "chroot_findtask" is
created.
```

Restrict priority changes

GRKERNSEC_CHROOT_NICE
Related sysctl variables:

kernel.grsecurity.chroot_restrict_nice

If you say Y here, processes inside a chroot will not be able to raise the priority of processes in the chroot, or alter the priority of processes outside the chroot. This provides more security than simply removing CAP_SYS_NICE from the process' capability set. If the sysctl option is enabled, a sysctl option with name "chroot_restrict_nice" is created.

Deny sysctl writes

GRKERNSEC_CHROOT_SYSCTL
Related sysctl variables:

kernel.grsecurity.chroot_deny_sysctl

If you say Y here, an attacker in a chroot will not be able to write to sysctl entries, either by sysctl(2) or through a /proc interface. It is strongly recommended that you say Y here. If the sysctl option is enabled, a sysctl option with name "chroot_deny_sysctl" is created.

Capability restrictions

GRKERNSEC_CHROOT_CAPS
Related sysctl variables:

kernel.grsecurity.chroot_caps

If you say Y here, the capabilities on all processes within a chroot jail will be lowered to stop module insertion, raw i/o, system and net admin tasks, rebooting the system, modifying immutable files, modifying IPC owned by another, and changing the system time. This is left an option because it can break some apps. Disable this if your chrooted apps are having problems performing those kinds of tasks. If the sysctl option is enabled, a sysctl option with name "chroot_caps" is created.

Exempt initrd tasks from restrictions

GRKERNSEC_CHROOT_INITRD

If you say Y here, tasks started prior to init will be exempted from grsecurity's chroot restrictions. This option is mainly meant to resolve Plymouth's performing privileged operations unnecessarily in a chroot.

8.9.5 Kernel Auditing

Single group for auditing

GRKERNSEC_AUDIT_GROUP
Related sysctl variables:

```
kernel.grsecurity.audit_gid
```

```
kernel.grsecurity.audit_group
```

```
If you say Y here, the exec and chdir logging features will only operate
on a group you specify.  This option is recommended if you only want to
watch certain users instead of having a large amount of logs from the
entire system.  If the sysctl option is enabled, a sysctl option with
name "audit_group" is created.
```

GID for auditing

GRKERNSEC_AUDIT_GID

Exec logging

GRKERNSEC_EXECLOG
Related sysctl variables:

```
kernel.grsecurity.exec_logging
```

```
If you say Y here, all execve() calls will be logged (since the
other exec*() calls are frontends to execve(), all execution
will be logged).  Useful for shell-servers that like to keep track
of their users.  If the sysctl option is enabled, a sysctl option with
name "exec_logging" is created.
WARNING: This option when enabled will produce a LOT of logs, especially
on an active system.
```

Resource logging

GRKERNSEC_RESLOG
Related sysctl variables:

```
kernel.grsecurity.resource_logging
```

```
If you say Y here, all attempts to overstep resource limits will
be logged with the resource name, the requested size, and the current
limit.  It is highly recommended that you say Y here.  If the sysctl
option is enabled, a sysctl option with name "resource_logging" is
created.  If the RBAC system is enabled, the sysctl value is ignored.
```

Log execs within chroot

GRKERNSEC_CHROOT_EXECLOG
Related sysctl variables:

```
kernel.grsecurity.chroot_execlog
```

```
If you say Y here, all executions inside a chroot jail will be logged
to syslog.  This can cause a large amount of logs if certain
applications (eg. djb's daemontools) are installed on the system, and
is therefore left as an option.  If the sysctl option is enabled, a
sysctl option with name "chroot_execlog" is created.
```

Ptrace logging

GRKERNSEC_AUDIT_PTRACE
Related sysctl variables:

```
kernel.grsecurity.audit_ptrace
```

```
If you say Y here, all attempts to attach to a process via ptrace
will be logged.  If the sysctl option is enabled, a sysctl option
with name "audit_ptrace" is created.
```

Chdir logging

GRKERNSEC_AUDIT_CHDIR
Related sysctl variables:

```
kernel.grsecurity.audit_chdir
```

```
If you say Y here, all chdir() calls will be logged.  If the sysctl
option is enabled, a sysctl option with name "audit_chdir" is created.
```

(Un)Mount logging

GRKERNSEC_AUDIT_MOUNT
Related sysctl variables:

```
kernel.grsecurity.audit_mount
```

```
If you say Y here, all mounts and unmounts will be logged.  If the
sysctl option is enabled, a sysctl option with name "audit_mount" is
created.
```

Signal logging

GRKERNSEC_SIGNAL
Related sysctl variables:

kernel.grsecurity.signal_logging

```
If you say Y here, certain important signals will be logged, such as
SIGSEGV, which will as a result inform you of when a error in a program
occurred, which in some cases could mean a possible exploit attempt.
If the sysctl option is enabled, a sysctl option with name
"signal_logging" is created.
```

Fork failure logging

GRKERNSEC_FORKFAIL
Related sysctl variables:

kernel.grsecurity.forkfail_logging

```
If you say Y here, all failed fork() attempts will be logged.
This could suggest a fork bomb, or someone attempting to overstep
their process limit.  If the sysctl option is enabled, a sysctl option
with name "forkfail_logging" is created.
```

Time change logging

GRKERNSEC_TIME
Related sysctl variables:

kernel.grsecurity.timechange_logging

```
If you say Y here, any changes of the system clock will be logged.
If the sysctl option is enabled, a sysctl option with name
"timechange_logging" is created.
```

/proc/<pid>/ipaddr support

GRKERNSEC_PROC_IPADDR

```
If you say Y here, a new entry will be added to each /proc/<pid>
directory that contains the IP address of the person using the task.
The IP is carried across local TCP and AF_UNIX stream sockets.
This information can be useful for IDS/IPSes to perform remote response
to a local attack.  The entry is readable by only the owner of the
process (and root if he has CAP_DAC_OVERRIDE, which can be removed via
the RBAC system), and thus does not create privacy concerns.
```

Denied RWX mmap/mprotect logging

GRKERNSEC_RWXMAP_LOG
Related sysctl variables:

kernel.grsecurity.rwxmap_logging

```
If you say Y here, calls to mmap() and mprotect() with explicit
usage of PROT_WRITE and PROT_EXEC together will be logged when
denied by the PAX_MPROTECT feature.  This feature will also
log other problematic scenarios that can occur when PAX_MPROTECT
is enabled on a binary, like textrels and PT_GNU_STACK.  If the
sysctl option is enabled, a sysctl option with name "rwxmap_logging"
is created.
```

8.9.6 Executable Protections

Dmesg(8) restriction

GRKERNSEC_DMESG
Related sysctl variables:

kernel.grsecurity.dmesg

```
If you say Y here, non-root users will not be able to use dmesg(8)
to view the contents of the kernel's circular log buffer.
The kernel's log buffer often contains kernel addresses and other
identifying information useful to an attacker in fingerprinting a
system for a targeted exploit.
If the sysctl option is enabled, a sysctl option with name "dmesg" is
created.
```

Deter ptrace-based process snooping

GRKERNSEC_HARDEN_PTRACE
Related sysctl variables:

kernel.grsecurity.harden_ptrace

```
If you say Y here, TTY sniffers and other malicious monitoring
programs implemented through ptrace will be defeated.  If you
have been using the RBAC system, this option has already been
enabled for several years for all users, with the ability to make
fine-grained exceptions.

This option only affects the ability of non-root users to ptrace
processes that are not a descendent of the ptracing process.
This means that strace ./binary and gdb ./binary will still work,
but attaching to arbitrary processes will not.  If the sysctl
option is enabled, a sysctl option with name "harden_ptrace" is
created.
```

Require read access to ptrace sensitive binaries

GRKERNSEC_PTRACE_READEXEC
Related sysctl variables:

kernel.grsecurity.ptrace_readexec

```
If you say Y here, unprivileged users will not be able to ptrace unreadable
binaries.  This option is useful in environments that
remove the read bits (e.g. file mode 4711) from suid binaries to
prevent infoleaking of their contents.  This option adds
consistency to the use of that file mode, as the binary could normally
be read out when run without privileges while ptracing.

If the sysctl option is enabled, a sysctl option with name "ptrace_readexec"
is created.
```

Enforce consistent multithreaded privileges

GRKERNSEC_SETXID
Related sysctl variables:

kernel.grsecurity.consistent_setxid

```
If you say Y here, a change from a root uid to a non-root uid
in a multithreaded application will cause the resulting uids,
gids, supplementary groups, and capabilities in that thread
to be propagated to the other threads of the process.  In most
cases this is unnecessary, as glibc will emulate this behavior
on behalf of the application.  Other libcs do not act in the
same way, allowing the other threads of the process to continue
running with root privileges.  If the sysctl option is enabled,
a sysctl option with name "consistent_setxid" is created.
```

Disallow access to overly-permissive IPC objects

GRKERNSEC_HARDEN_IPC
Related sysctl variables:

kernel.grsecurity.harden_ipc

```
If you say Y here, access to overly-permissive IPC objects (shared
memory, message queues, and semaphores) will be denied for processes
given the following criteria beyond normal permission checks:
1) If the IPC object is world-accessible and the euid doesn't match
   that of the creator or current uid for the IPC object
2) If the IPC object is group-accessible and the egid doesn't
   match that of the creator or current gid for the IPC object
It's a common error to grant too much permission to these objects,
with impact ranging from denial of service and information leaking to
privilege escalation.  This feature was developed in response to
research by Tim Brown:
http://labs.
portcullis.co.uk/whitepapers/memory-squatting-attacks-on-system-v-shared-memory/
who found hundreds of such insecure usages.  Processes with
CAP_IPC_OWNER are still permitted to access these IPC objects.
```

```
If the sysctl option is enabled, a sysctl option with name
"harden_ipc" is created.
```

Trusted Path Execution (TPE)

GRKERNSEC_TPE
Related sysctl variables:

```
kernel.grsecurity.tpe
```

```
kernel.grsecurity.tpe_gid
```

```
If you say Y here, you will be able to choose a gid to add to the
supplementary groups of users you want to mark as "untrusted."
These users will not be able to execute any files that are not in
root-owned directories writable only by root.  If the sysctl option
is enabled, a sysctl option with name "tpe" is created.
```

Partially restrict all non-root users

GRKERNSEC_TPE_ALL
Related sysctl variables:

```
kernel.grsecurity.tpe_restrict_all
```

```
If you say Y here, all non-root users will be covered under
a weaker TPE restriction.  This is separate from, and in addition to,
the main TPE options that you have selected elsewhere.  Thus, if a
"trusted" GID is chosen, this restriction applies to even that GID.
Under this restriction, all non-root users will only be allowed to
execute files in directories they own that are not group or
world-writable, or in directories owned by root and writable only by
root.  If the sysctl option is enabled, a sysctl option with name
"tpe_restrict_all" is created.
```

Invert GID option

GRKERNSEC_TPE_INVERT
Related sysctl variables:

```
kernel.grsecurity.tpe_invert
```

```
If you say Y here, the group you specify in the TPE configuration will
decide what group TPE restrictions will be *disabled* for.  This
option is useful if you want TPE restrictions to be applied to most
users on the system.  If the sysctl option is enabled, a sysctl option
with name "tpe_invert" is created.  Unlike other sysctl options, this
entry will default to on for backward-compatibility.
```

GID for TPE-untrusted users

GRKERNSEC_TPE_UNTRUSTED_GID

```
Setting this GID determines what group TPE restrictions will be
*enabled* for.  If the sysctl option is enabled, a sysctl option
with name "tpe_gid" is created.
```

GID for TPE-trusted users

GRKERNSEC_TPE_TRUSTED_GID

```
Setting this GID determines what group TPE restrictions will be
*disabled* for.  If the sysctl option is enabled, a sysctl option
with name "tpe_gid" is created.
```

8.9.7 Network Protections

Larger entropy pools

GRKERNSEC_RANDNET

```
If you say Y here, the entropy pools used for many features of Linux
and grsecurity will be doubled in size.  Since several grsecurity
features use additional randomness, it is recommended that you say Y
here.  Saying Y here has a similar effect as modifying
/proc/sys/kernel/random/poolsize.
```

TCP/UDP blackhole and LAST_ACK DoS prevention

GRKERNSEC_BLACKHOLE
Related sysctl variables:

```
kernel.grsecurity.ip_blackhole
kernel.grsecurity.lastack_retries
```

```
If you say Y here, neither TCP resets nor ICMP
destination-unreachable packets will be sent in response to packets
sent to ports for which no associated listening process exists.
This feature supports both IPV4 and IPV6 and exempts the
loopback interface from blackholing.  Enabling this feature
makes a host more resilient to DoS attacks and reduces network
visibility against scanners.

The blackhole feature as-implemented is equivalent to the FreeBSD
blackhole feature, as it prevents RST responses to all packets, not
just SYNs.  Under most application behavior this causes no
problems, but applications (like haproxy) may not close certain
connections in a way that cleanly terminates them on the remote
end, leaving the remote host in LAST_ACK state.  Because of this
side-effect and to prevent intentional LAST_ACK DoSes, this
feature also adds automatic mitigation against such attacks.
```

The mitigation drastically reduces the amount of time a socket
can spend in LAST_ACK state. If you're using haproxy and not
all servers it connects to have this option enabled, consider
disabling this feature on the haproxy host.

If the sysctl option is enabled, two sysctl options with names
"ip_blackhole" and "lastack_retries" will be created.
While "ip_blackhole" takes the standard zero/non-zero on/off
toggle, "lastack_retries" uses the same kinds of values as
"tcp_retries1" and "tcp_retries2". The default value of 4
prevents a socket from lasting more than 45 seconds in LAST_ACK
state.

Disable TCP Simultaneous Connect

GRKERNSEC_NO_SIMULT_CONNECT

If you say Y here, a feature by Willy Tarreau will be enabled that
removes a weakness in Linux's strict implementation of TCP that
allows two clients to connect to each other without either entering
a listening state. The weakness allows an attacker to easily prevent
a client from connecting to a known server provided the source port
for the connection is guessed correctly.

As the weakness could be used to prevent an antivirus or IPS from
fetching updates, or prevent an SSL gateway from fetching a CRL,
it should be eliminated by enabling this option. Though Linux is
one of few operating systems supporting simultaneous connect, it
has no legitimate use in practice and is rarely supported by firewalls.

Socket restrictions

GRKERNSEC_SOCKET

If you say Y here, you will be able to choose from several options.
If you assign a GID on your system and add it to the supplementary
groups of users you want to restrict socket access to, this patch
will perform up to three things, based on the option(s) you choose.

Deny any sockets to group

GRKERNSEC_SOCKET_ALL
Related sysctl variables:

kernel.grsecurity.socket_all

kernel.grsecurity.socket_all_gid

If you say Y here, you will be able to choose a GID of whose users will
be unable to connect to other hosts from your machine or run server
applications from your machine. If the sysctl option is enabled, a
sysctl option with name "socket_all" is created.

GID to deny all sockets for

GRKERNSEC_SOCKET_ALL_GID

Here you can choose the GID to disable socket access for. Remember to add the users you want socket access disabled for to the GID specified here. If the sysctl option is enabled, a sysctl option with name "socket_all_gid" is created.

Deny client sockets to group

GRKERNSEC_SOCKET_CLIENT
Related sysctl variables:

kernel.grsecurity.socket_client

kernel.grsecurity.socket_client_gid

If you say Y here, you will be able to choose a GID of whose users will be unable to connect to other hosts from your machine, but will be able to run servers. If this option is enabled, all users in the group you specify will have to use passive mode when initiating ftp transfers from the shell on your machine. If the sysctl option is enabled, a sysctl option with name "socket_client" is created.

GID to deny client sockets for

GRKERNSEC_SOCKET_CLIENT_GID

Here you can choose the GID to disable client socket access for. Remember to add the users you want client socket access disabled for to the GID specified here. If the sysctl option is enabled, a sysctl option with name "socket_client_gid" is created.

Deny server sockets to group

GRKERNSEC_SOCKET_SERVER
Related sysctl variables:

kernel.grsecurity.socket_server

kernel.grsecurity.socket_server_gid

If you say Y here, you will be able to choose a GID of whose users will be unable to run server applications from your machine. If the sysctl option is enabled, a sysctl option with name "socket_server" is created.

GID to deny server sockets for

GRKERNSEC_SOCKET_SERVER_GID

```
Here you can choose the GID to disable server socket access for.
Remember to add the users you want server socket access disabled for to
the GID specified here.  If the sysctl option is enabled, a sysctl
option with name "socket_server_gid" is created.
```

8.9.8 Physical Protections

Deny new USB connections after toggle

GRKERNSEC_DENYUSB
Related sysctl variables:

```
 kernel.grsecurity.deny_new_usb
```

```
If you say Y here, a new sysctl option with name "deny_new_usb"
will be created.  Setting its value to 1 will prevent any new
USB devices from being recognized by the OS.  Any attempted USB
device insertion will be logged.  This option is intended to be
used against custom USB devices designed to exploit vulnerabilities
in various USB device drivers.

For greatest effectiveness, this sysctl should be set after any
relevant init scripts.  This option is safe to enable in distros
as each user can choose whether or not to toggle the sysctl.
```

Reject all USB devices not connected at boot

GRKERNSEC_DENYUSB_FORCE

```
If you say Y here, a variant of GRKERNSEC_DENYUSB will be enabled
that doesn't involve a sysctl entry.  This option should only be
enabled if you're sure you want to deny all new USB connections
at runtime and don't want to modify init scripts.  This should not
be enabled by distros.  It forces the core USB code to be built
into the kernel image so that all devices connected at boot time
can be recognized and new USB device connections can be prevented
prior to init running.
```

8.9.9 Sysctl Support

Sysctl support

GRKERNSEC_SYSCTL

```
If you say Y here, you will be able to change the options that
grsecurity runs with at bootup, without having to recompile your
kernel.  You can echo values to files in /proc/sys/kernel/grsecurity
to enable (1) or disable (0) various features.  All the sysctl entries
```

```
are mutable until the "grsec_lock" entry is set to a non-zero value.
All features enabled in the kernel configuration are disabled at boot
if you do not say Y to the "Turn on features by default" option.
All options should be set at startup, and the grsec_lock entry should
be set to a non-zero value after all the options are set.
*THIS IS EXTREMELY IMPORTANT*
```

Extra sysctl support for distro makers (READ HELP)

GRKERNSEC_SYSCTL_DISTRO

```
If you say Y here, additional sysctl options will be created
for features that affect processes running as root.  Therefore,
it is critical when using this option that the grsec_lock entry be
enabled after boot.  Only distros with prebuilt kernel packages
with this option enabled that can ensure grsec_lock is enabled
after boot should use this option.
*Failure to set grsec_lock after boot makes all grsec features
this option covers useless*

Currently this option creates the following sysctl entries:
"Disable Privileged I/O": "disable_priv_io"
```

Turn on features by default

GRKERNSEC_SYSCTL_ON

```
If you say Y here, instead of having all features enabled in the
kernel configuration disabled at boot time, the features will be
enabled at boot time.  It is recommended you say Y here unless
there is some reason you would want all sysctl-tunable features to
be disabled by default.  As mentioned elsewhere, it is important
to enable the grsec_lock entry once you have finished modifying
the sysctl entries.
```

8.9.10 Logging Options

Seconds in between log messages (minimum)

GRKERNSEC_FLOODTIME

```
This option allows you to enforce the number of seconds between
grsecurity log messages.  The default should be suitable for most
people, however, if you choose to change it, choose a value small enough
to allow informative logs to be produced, but large enough to
prevent flooding.

Setting both this value and GRKERNSEC_FLOODBURST to 0 will disable
any rate limiting on grsecurity log messages.
```

Number of messages in a burst (maximum)

```
GRKERNSEC_FLOODBURST
```

```
This option allows you to choose the maximum number of messages allowed
within the flood time interval you chose in a separate option.  The
default should be suitable for most people, however if you find that
many of your logs are being interpreted as flooding, you may want to
raise this value.

Setting both this value and GRKERNSEC_FLOODTIME to 0 will disable
any rate limiting on grsecurity log messages.
```

8.10 Appendix Tables

Mode	Meaning
u	This role is a user role. That is, the role name must be an existing user on the system.
g	This role is a group role. That is, the role name must be an existing group on the system.
s	This role is a special role, meaning it does not belong to a user or group and does not require an enforced secure policy base to be included in the ruleset.
l	Lowercase L. This role has learning enabled.
A	This role is an administrative role, thus it has special privileges that normal roles do not have. In particular, this role bypasses the additional ptrace and library loading restrictions.
G	This role can use `gradm` to authenticate to the kernel. A policy for `gradm` will automatically be added to the role.
N	This role does not require authentication. To access this role, use '`gradm -n <rolename>` '.
P	This role uses Pluggable Authentication Modules[1] (PAM) for authentication.
T	This role has Trusted Path Execution (TPE) enabled.
R	The role is persistent. When the shell/session in which authorization was done is terminated, spawned processes won't be dropped to non-special role. Do **NOT** use this flag with any role that does anything but shut the system down.

8.11 role_transitions

Role transitions specify which special roles a given role is allowed to authenticate to. This applies to special roles that do not require password authentication as well. If a user tries to authenticate to a role that is not within his transition table, he will receive a permission denied error. A common mistake when creating a new special role is forgetting to create a `role_transitions` rule for the role that will transition to the special role, which a user

1 `http://en.wikipedia.org/wiki/Pluggable_Authentication_Modules`

confuses with having entered an incorrect password. The `role_transitions` rule is added below the declaration of a role, but before any subject declaration.

Usage:

```
role_transitions <special role 1> <special role 2> ... <special role n>
```

Example:

```
role person u
role_transitions www_admin dns_admin
subject /
...
```

8.12 role_allow_ip

This rule restricts the use of a role to a list of IPs. If a user is on the system who would normally get the rule does not belong to the specified list of IPs, the system falls back through its method of determining a role for the user (checking for an applicable group role then falling back to the default role). This rule can be specified multiple times for a role. Like `role_transitions` , it should be added below the declaration of a role, but before any subject declaration.

Usage:

```
role_allow_ip <IP>/<optional netmask>
```

Example:

```
role person u
role_allow_ip 192.168.1.0/24
subject /
...
```

A netmask of `0.0.0.0/32` permits use of the role only by local processes that haven't been used by remote clients `http://permalink.gmane.org/gmane.linux.kernel.grsecurity/74`.

8.13 role_umask

This rule ensures that a user cannot accidentally create a file that others can read (a confidentiality issue). Like previous role attributes, it should be added below the declaration of a role, but before any subject declaration.

Usage:

```
role_umask <mask>
```

Example:

```
role person u
role_umask 077
subject /
...
```

Mode	Meaning
a	Allow this process to talk to the */dev/grsec* device.
b	Enable process accounting for processes in this subject.
d	Protect the */proc/<pid>/fd* , */proc/<pid>/mem* , */proc/<pid>/cmdline* , and */proc/<pid>/environ* entries for processes in this subject,
h	This process is hidden and only viewable by processes with the *v* mode.
i	Enable inheritance-based learning, causing all accesses of this subject and anything it executes to be logged as originating from this subject. The policy generated from this learning will have the inheritance flag added to every file executed from this subject.
k	This process can kill protected processes.
l	Enables learning mode for this process.
o	Override ACL inheritance for this process.
p	This process is protected; it can only be killed by processes with the *k* mode, or by processes within the same subject.
r	Relax `ptrace` restrictions (allows `ptracing` of processes other than one's own children).
s	(In v2.2.1 and above): Enable `AT_SECURE` when entering this subject. This enables the same environment sanitization that occurs in glibc upon execution of a suid binary.
t	Allow `ptracing` of any process (do not use unless necessary, allows `ptrace` to cross subject boundaries). This flag also allows a process to use `CLONE_FS` and execute a binary that causes a subject change.
v	This process can view hidden processes.
x	Allows executable anonymous shared memory for this subject.
A	Protect the shared memory of this subject. No other processes but processes contained within this subject may access the shared memory of this subject.
C	Auto-kill all processes belonging to the attacker's IP address upon violation of security policy.
K	When processes belonging to this subject generate an alert, kill the process.
O	Allow loading of writable libraries.
T	Deny execution of binaries or scripts that are writable by any other subject in the policy. This flag is evaluated at policy enable time. All binaries with execute permission that are writable by another subject (ignoring special roles) will be reported and the RBAC system will not allow itself to be enabled until the changes are made.

8.14 user/group transitions

You may specify what users and groups a given subject can transition to. This can be done on an inclusive or exclusive basis. Omitting these rules allows a subject with proper privilege granted by capabilities to transition to any user/group.

Usage:

```
 user_transition_allow <user 1> <user 2> ... <user n>
 user_transition_deny <protected user 1> <protected user 2> ... <protected user
n>

 group_transition_allow <group 1> <group 2> ... <group n>
 group_transition_deny <protected group 1> <protected group 2> ... <protected
group n>
```

Example:

```
 role person u
 subject /bin/su
 user_transition_allow root spender
 group_transition_allow root spender
 ...

 role person u
 subject /bin/su
 user_transition_deny specialuser
 user_transition_deny specialgroup
 ...
```

8.15 ip_override

It is possible to force a given subject to bind to a particular IP address on the machine. This can be useful for some sandboxed environments, to ensure the source IP used from the sandbox is one determined by RBAC policy. To restrict what other source IP addresses a subject can bind to, use the normal IP ACL support of the RBAC system. This option is solely used to override an application's use of INADDR_ANY when connecting out or binding to a local port.

Usage:

```
ip_override <IP>
```

Example:

```
role person u
subject /
ip_override 192.168.0.1
...
```

8.16 Socket policy (`bind` /`connect` /`sock_allow_family`)

`bind` /`connect` are described under The RBAC System[2].

When connect/bind rules are used, additional rules will be required to unlock the use of additional socket families (outside of the common unix family). Multiple families can be specified per line.

To enable use of IPv6, add the line:

```
sock_allow_family ipv6
```

To enable use of netlink, add the line:

```
sock_allow_family netlink
```

To enable all other families, add the line:

```
sock_allow_family all
```

Mode	Meaning
Object Permission Modes	
none	Lack of any of the below modes implies ”find” access to the object. The object can be listed and have its ownership, size, etc. information obtained, but cannot be read or modified.
a	This object can be opened for appending.
c	Allow creation of the file/directory.
d	Allow deletion of the file/directory.
f	Needed to mark the pipe used for communication with init to transfer the privilege of the persistent role; only valid within a persistent role. Transfer only occurs when the file is opened for writing.
h	This object is hidden.
i	This mode only applies to binaries. When the object is executed, it inherits the ACL of the subject in which it was contained.
l	Lowercase L. Allow a hardlink at this path. Hardlinking requires a minimum of c and l modes, and the target link cannot have any greater permission than the source file.
m	Allow creation of setuid/setgid files/directories and modification of files/directories to be setuid/setgid.
p	Reject all ptraces to this object.
r	This object can be opened for reading.
t	This object can be ptraced, but cannot modify the running task. This is referred to as a ’read-only ptrace’.
w	This object can be opened for writing or appending.
x	This object can be executed (or mmap’d with PROT_EXEC into a task).

2 Chapter 4.10 on page 37

Mode	Meaning
Object Auditing Flags	
A	Audit successful appends to this object.
C	Audit the creation of the file/directory.
D	Audit the deletion of the file/directory.
F	Audit successful finds of this object.
I	Audit successful ACL inherits of this object.
L	Audit link creation.
M	Audit the setuid/setgid creation/modification.
R	Audit successful reads to this object.
W	Audit successful writes to this object.
X	Audit successful execs of this object.

Mode	Meaning
Other Object Flags	
s	Logs will be suppressed for denied access to this object.

This table lists all PaX[3] flags that can be forced on or off in the policy, regardless of the flags on the binary, by using + or - before the flag name.

Flag	Description	Details
PAX_EMUTRAMP	PaX Trampoline emulation[4]	http://pax.grsecurity.net/docs/emutramp.txt
PAX_MPROTECT	PaX MPROTECT[5]	http://pax.grsecurity.net/docs/mprotect.txt
PAX_PAGEEXEC	PaX PAGEEXEC[6]	http://pax.grsecurity.net/docs/pageexec.txt
PAX_RANDMMAP	PaX ASLR[7]	http://pax.grsecurity.net/docs/randmmap.txt
PAX_SEGMEXEC	PaX SEGMEXEC[8]	http://pax.grsecurity.net/docs/segmexec.txt

This table lists all standard Linux[9] capabilities and one special capability related to grsecurity. With capabilities, the system is divided into logical groups that may be individually granted to, or removed from, different processes. See Capability Restrictions[10] for more information.

3 http://en.wikipedia.org/wiki/PaX
4 http://en.wikipedia.org/wiki/PaX%23Trampoline_emulation
5 http://en.wikipedia.org/wiki/PaX%23Restricted_mprotect.28.29
6 http://en.wikipedia.org/wiki/PaX%23PAGEEXEC
7 http://en.wikipedia.org/wiki/PaX%23Address_space_layout_randomization
8 http://en.wikipedia.org/wiki/PaX%23SEGMEXEC
9 http://en.wikipedia.org/wiki/Linux
10 Chapter 4.8 on page 34

Capability Name	Meaning
CAP_ALL	CAP_ALL is not a real capability, but was coded into `gradm` to represent all capabilities. Therefore to denote dropping of all capabilities, but CAP_SETUID, -CAP_ALL and +CAP_SETUID would be used.
CAP_CHOWN	In a system with the [_POSIX_CHOWN_RESTRICTED] option defined, this overrides the restriction of changing file ownership and group ownership.
CAP_DAC_OVERRIDE	Override all DAC access, including ACL execute access if [_POSIX_ACL] is defined. Excluding DAC access covered by CAP_LINUX_IMMUTABLE.
CAP_DAC_READ_SEARCH	Overrides all DAC restrictions, regarding read and search on files and directories, including ACL restrictions, if [_POSIX_ACL] is defined. Excluding DAC access covered by CAP_LINUX_IMMUTABLE.
CAP_FOWNER	Overrides all restrictions about allowed operations on files, where file owner ID must be equal to the user ID, except where CAP_FSETID is applicable. It doesn't override MAC and DAC restrictions.
CAP_FSETID	Overrides the following restrictions, that the effective user ID shall match the file owner ID, when setting the S_ISUID and S_ISGID bits on that file; that the effective group ID (or one of the supplementary group IDs) shall match the file owner ID when setting the S_ISGID bit on that file; that the S_ISUID and S_ISGID bits are cleared on successful return from `chown(2)` (not implemented).
CAP_KILL	Overrides the restriction, that the real or effective user ID of a process, sending a signal, must match the real or effective user ID of the process receiving the signal.
CAP_SETGID	• Allows `setgid(2)` manipulation. • Allows `setgroups(2)` . • Allows forged gids on socket credentials passing.
CAP_SETUID	• Allows `set*uid(2)` manipulation (including fsuid). • Allows forged pids on socket credentials passing.

Capability Name	Meaning
CAP_SETPCAP	Without VFS support for capabilities: • Transfer any capability in your permitted set to any pid, remove any capability in your permitted set from any pid. With VFS support for capabilities (neither of above, but) • Add any capability from current's capability bounding set to the current process' inheritable set • Allow taking bits out of capability bounding set. • Allow modification of the securebits for a process.
CAP_LINUX_IMMUTABLE	Allow modification of S_IMMUTABLE and S_APPEND file attributes.
CAP_NET_BIND_SERVICE	• Allows binding to TCP/UDP sockets below 1024. • Allows binding to ATM VCIs below 32.
CAP_NET_BROADCAST	Allow broadcasting, listen to multicast.
CAP_NET_ADMIN	• Allow interface configuration. • Allow administration of IP firewall, masquerading and accounting. • Allow setting debug option on sockets. • Allow modification of routing tables. • Allow setting arbitrary process / process group ownership on sockets. • Allow binding to any address for transparent proxying. • Allow setting TOS (type of service). • Allow setting promiscuous mode. • Allow clearing driver statistics. • Allow multicasting. • Allow read/write of device–specific registers. • Allow activation of ATM control sockets.
CAP_NET_RAW	• Allow use of RAW sockets. • Allow use of PACKET sockets.
CAP_IPC_LOCK	• Allow locking of shared memory segments. • Allow `mlock` and `mlockall` (which doesn't really have anything to do with IPC).
CAP_IPC_OWNER	Override IPC ownership checks.
CAP_SYS_MODULE	Insert and remove kernel modules – modify kernel without limit.

Capability Name	**Meaning**
CAP_SYS_RAWIO	• Allow ioperm/iopl access • Allow sending USB messages to any device via */proc/bus/usb*
CAP_SYS_CHROOT	Allow use of `chroot()` .
CAP_SYS_PTRACE	Allow `ptrace()` of any process.
CAP_SYS_PACCT	Allow configuration of process accounting.

Capability Name	**Meaning**
CAP_SYS_ADMIN	• Allow configuration of the secure attention key. • Allow administration of the random device. • Allow examination and configuration of disk quotas. • Allow configuring the kernel's syslog (printk behaviour). • Allow setting the domainname. • Allow setting the hostname. • Allow calling `bdflush()` . • Allow `mount()` and umount(), setting up new smb connection. • Allow some autofs root ioctls. • Allow nfsservctl. • Allow VM86_REQUEST_IRQ. • Allow to read/write pci config on alpha. • Allow irix_prctl on mips (setstacksize). • Allow flushing all cache on m68k (sys_cacheflush). • Allow removing semaphores. Used instead of CAP_CHOWN to "chown" IPC message queues, semaphores and shared memory. • Allow locking/unlocking of shared memory segment. • Allow turning swap on/off. • Allow forged pids on socket credentials passing. • Allow setting readahead and flushing buffers on block devices. • Allow setting geometry in floppy driver. • Allow turning DMA on/off in xd driver. • Allow administration of md devices (mostly the above, but some extra ioctls). • Allow tuning the ide driver. • Allow access to the nvram device. • Allow administration of apm_bios, serial and bttv (TV) device. • Allow manufacturer commands in isdn CAPI support driver. • Allow reading non–standardized portions of pci configuration space. • Allow DDI debug ioctl on sbpcd driver. • Allow setting up serial ports. • Allow sending raw qic–117 commands. • Allow enabling/disabling tagged queuing on SCSI controllers and sending arbitrary SCSI commands. • Allow setting encryption key on loopback filesystem. • Allow setting zone reclaim policy.

Capability Name	Meaning
CAP_SYS_BOOT	• Allow use of `reboot()` • Allow use of `kexec()` syscall
CAP_SYS_NICE	• Allow raising priority and setting priority on other (different UID) processes. • Allow use of FIFO and round–robin (realtime) scheduling on own processes and setting the scheduling algorithm used by another process. • Allow setting cpu affinity on other processes.
CAP_SYS_RESOURCE	• Override resource limits. Set resource limits. • Override quota limits • Override reserved space on ext2 filesystem • Modify data journaling mode on ext3 filesystem (uses journaling resources). NOTE: ext2 honors fsuid when checking for resource overrides, so you can override using fsuid too. • Override size restrictions on IPC message queues. • Allow more than 64Hz interrupts from the real–time clock. • Override max number of consoles on console allocation. • Override max number of keymaps.
CAP_SYS_TIME	• Allow manipulation of system clock. • Allow `irix_stime` on mips. • Allow setting the real–time clock.
CAP_SYS_TTY_CONFIG	• Allow configuration of tty devices. • Allow `vhangup()` of tty.
CAP_MKNOD	Allow the privileged aspects of `mknod()` .
CAP_LEASE	Allow taking of leases on files.
CAP_AUDIT_WRITE	Allow emitting auditing messages.
CAP_AUDIT_CONTROL	Allow administration of the kernel's auditing system.
CAP_SETFCAP	Allow the setting of file capabilities.
CAP_MAC_OVERRIDE	Override MAC access. The base kernel enforces no MAC policy. An LSM may enforce a MAC policy and if it does and it chooses to implement capability based overrides of that policy, this is the capability it should use to do so.

Capability Name	Meaning
CAP_MAC_ADMIN	Allow MAC configuration or state changes. The base kernel requires no MAC configuration. An LSM may enforce a MAC policy, and if it does and it chooses to implement capability based checks on modifications to that policy or the data required to maintain it, this is the capability it should use to do so.
CAP_SYSLOG	Allow configuring the kernel's syslog (printk behaviour).
CAP_WAKE_ALARM	Allow triggering something that will wake the system.

8.17 Introduction

This table lists all system resources that can be restricted by grsecurity. Grsecurity supports all the resources Linux[11] supports, but uses slightly different names for them: The `RLIMIT` prefix has been replaced with `RES` . For example, the Linux[12] resource `RLIMIT_CPU` is called `RES_CPU` in grsecurity.

For detailed information about resources in Linux, see the man page of getrlimit[13].

8.18 Syntax and Examples

A single resource rule follows the following syntax:

```
<resource name> <soft limit> <hard limit>
```

An example of this syntax would be:

```
RES_FSIZE 5K 5K
```

This would prevent the process from creating files that are bigger than 5 Kilobytes[14].

Using `unlimited` is valid for both the soft limit and the hard limit, to denote an unlimited resource. Note that by omitting a resource restriction, the system's default limits are used (as set by PAM or the application itself). If a resource is specified within the policy, the specific limits override the system's default limits for the given subject.

A number of suffixes are allowed when specifying resource limits. They are described below.

Suffix	Meaning
s	Amount of time in seconds.
m	Amount of time in minutes.

11 http://en.wikipedia.org/wiki/Linux
12 http://en.wikipedia.org/wiki/Linux
13 http://www.kernel.org/doc/man-pages/online/pages/man2/getrlimit.2.html
14 http://en.wikipedia.org/wiki/Kilobyte

Suffix	Meaning
h	Amount of time in hours.
d	Amount of time in days.
K	Size in kilobytes.
M	Size in megabytes.
G	Size in gigabytes.

A full list of supported resources is supplied below.

Resource Name	Meaning
RES_AS	The maximum size of the process's virtual memory (address space) in bytes.
RES_CORE	Maximum size of core file, in bytes. When 0 no core dump files are created. When non-zero, larger dumps are truncated to this size.
RES_CPU	CPU time limit in seconds.
RES_DATA	The maximum size of the process's data segment, in bytes (initialized data, uninitialized data, and heap).
RES_FSIZE	The maximum size of files, in bytes, that the process may create.
RES_LOCKS	A limit on the combined number of flock() locks and fcntl() leases that this process may establish.
RES_MEMLOCK	The maximum number of bytes of memory that may be locked into RAM. In effect this limit is rounded down to the nearest multiple of the system page size.
RES_MSGQUEUE	Specifies the limit on the number of bytes that can be allocated for POSIX message queues for the real user ID of the calling process
RES_NICE	Specifies a ceiling to which the process's nice value can be raised using setpriority(2) or nice(2).
RES_NOFILE	Specifies a value one greater than the maximum file descriptor number that can be opened by this process.
RES_NPROC	The maximum number of threads that can be created for the real user ID of the calling process.
RES_RSS	Specifies the limit (in pages) of the process's resident set (the number of virtual pages resident in RAM). This limit only has effect in Linux 2.4.x, x $<$ 30.
RES_RTPRIO	Specifies a ceiling on the real-time priority that may be set for this process using sched_setscheduler(2) and sched_setparam(2).
RES_SIGPENDING	Specifies the limit on the number of signals that may be queued for the real user ID of the calling process. Both standard and real-time signals are counted for the purpose of checking this limit.
RES_STACK	The maximum size of the process stack, in bytes.

Resource Name	Meaning
RES_RTTIME	Specifies a limit on the amount of CPU time that a process scheduled under a real-time scheduling policy may consume without making a blocking system call. for the purpose of this limit, each time a process makes a blocking system call, the count of its consumed CPU time is reset to zero. The CPU time count is not reset if the process continues trying to use the CPU but is preempted, if its time slice expires, or if it calls sched_yield(2). Upon reaching the soft limit, the process is sent a SIGXCPU signal. If the process catches or ignores this signal and continues consuming CPU time, then SIGXCPU will be generated once each second until the hard limit is reached, at which point the process is sent a SIGKILL signal. The intended use of this limit is to stop a runaway real-time process from locking up the system.
RES_CRASH	This is a pseudo-resource interpreted only by the RBAC system. The meaning of soft and hard limits are overridden for this resource. The intent of the resource is to be able to rate-limit bruteforced exploit attempts for a given subject. The soft limit for this resource is the number of times the subject is allowed to crash in ways that suggest an exploitation attempt. The hard limit specifies the amount of time those crashes are allowed to occur in. With a rule like `RES_CRASH 3 30m` one can limit a privileged binary to three exploit attempts every 10 minutes, deterring bruteforcing attempts. The RBAC system actively responds to bruteforcing attempts that overstep this limit. If the target is a setuid binary, the attacker has all of his/her processes killed and will be unable to log in for the remainder of the configured time. If the target is a forking network daemon, that daemon will be unable to fork additional copies of itself for the remainder of the configured time.

Below is a table of every available option in grsecurity that can be changed at runtime. The options can be changed using the sysctl[15] interface, or by echoing values to files in */proc/sys/kernel/grsecurity/* . Available options vary depending on how grsecurity was configured. See Configuring and Installing grsecurity[16] and Runtime configuration[17] for more information.

To find out what options are available in your system, list the contents of */proc/sys/kernel/grsecurity* . If you use the `sysctl` interface, all of grsecurity's option are prefixed with kernel.grsecurity (e.g. kernel.grsecurity.audit_chdir).

Clicking an option will take you to or at least close to its description in another appendix page.

15 `http://en.wikipedia.org/wiki/Sysctl`
16 Chapter 2.6.1 on page 13
17 Chapter 3.7 on page 25

This list of sysctl options was generated February 15, 2014 from the grsec_sysctl.c file of grsecurity 3.0-3.13.3-201402132113.patch using a script. Manual updates will be lost the next time the content is regenerated.

Auditing **Chroot restrictions** **Network-based features** **Misc. options**

All sysctl variables

Auditing

- audit_chdir[18]
- audit_gid[19]
- audit_group[20]
- audit_mount[21]
- audit_ptrace[22]
- chroot_execlog[23]
- exec_logging[24]
- forkfail_logging[25]
- resource_logging[26]
- rwxmap_logging[27]
- signal_logging[28]
- timechange_logging[29]

Chroot restrictions

- chroot_caps[30]
- chroot_deny_chmod[31]
- chroot_deny_chroot[32]
- chroot_deny_fchdir[33]
- chroot_deny_mknod[34]
- chroot_deny_mount[35]
- chroot_deny_pivot[36]
- chroot_deny_shmat[37]
- chroot_deny_sysctl[38]
- chroot_deny_unix[39]
- chroot_enforce_chdir[40]
- chroot_findtask[41]
- chroot_restrict_nice[42]

Network-based features

- harden_ipc[43]
- ip_blackhole[44]
- socket_all[45]
- socket_all_gid[46]
- socket_client[47]
- socket_all_gid[48]
- socket_client[49]
- socket_client_gid[50]
- socket_server[51]
- socket_server_gid[52]

Misc. options

- consistent_setxid[53]
- deny_new_usb[54]
- deter_bruteforce[55]
- disable_priv_io[56]
- dmesg[57]
- enforce_symlinksifowner[58]
- fifo_restrictions[59]
- harden_ptrace[60]
- lastack_retries[61]
- linking_restrictions[62]
- ptrace_readexec[63]
- romount_protect[64]
- symlinkown_gid[65]
- tpe[66]
- tpe_gid[67]
- tpe_invert[68]
- tpe_restrict_all[69]

18 Chapter 8.9.5 on page 84
19 Chapter 8.9.5 on page 83
20 Chapter 8.9.5 on page 83
21 Chapter 8.9.5 on page 84
22 Chapter 8.9.5 on page 84
23 Chapter 8.9.5 on page 84
24 Chapter 8.9.5 on page 83
25 Chapter 8.9.5 on page 85
26 Chapter 8.9.5 on page 83
27 Chapter 8.9.5 on page 86
28 Chapter 8.9.5 on page 85
29 Chapter 8.9.5 on page 85
30 Chapter 8.9.4 on page 82
31 Chapter 8.9.4 on page 80
32 Chapter 8.9.4 on page 79
33 Chapter 8.9.4 on page 80
34 Chapter 8.9.4 on page 80
35 Chapter 8.9.4 on page 79
36 Chapter 8.9.4 on page 79
37 Chapter 8.9.4 on page 81
38 Chapter 8.9.4 on page 82
39 Chapter 8.9.4 on page 81
40 Chapter 8.9.1 on page 60
41 Chapter 8.9.4 on page 81
42 Chapter 8.9.4 on page 82
43 Chapter 8.9.6 on page 87
44 Chapter 8.9.7 on page 89
45 Chapter 8.9.7 on page 90
46 Chapter 8.9.7 on page 90
47 Chapter 8.9.7 on page 91
48 Chapter 8.9.7 on page 90
49 Chapter 8.9.7 on page 91
50 Chapter 8.9.7 on page 91
51 Chapter 8.9.7 on page 91
52 Chapter 8.9.7 on page 91
53 Chapter 8.9.6 on page 87
54 Chapter 8.9.8 on page 92
55 Chapter 8.9.2 on page 72
56 Chapter 8.9.2 on page 70
57 Chapter 8.9.6 on page 86
58 Chapter 8.9.4 on page 77
59 Chapter 8.9.4 on page 77
60 Chapter 8.9.6 on page 86
61 Chapter 8.9.7 on page 89
62 Chapter 8.9.4 on page 76
63 Chapter 8.9.6 on page 87
64 Chapter 8.9.4 on page 78
65 Chapter 8.9.4 on page 77
66 Chapter 8.9.6 on page 88
67 Chapter 8.9.6 on page 88
68 Chapter 8.9.6 on page 88
69 Chapter 8.9.6 on page 88

9 Credits and Permissions

10 Introduction

On this page you will find documentation regarding permissions to use material written by others before this Wikibook was started.

10.1 The Original grsecurity Documentation

The original documentation for grsecurity was written by Brad Spengler, the author of grsecurity. This includes the ACL documentation[1] and the grsecurity Quick-Start Guide[2] (PDF).

10.2 Permission to Use the Official Documentation

Below is the correspondence between myself (Meev0[3] (talk[4])) and Brad Spengler regarding the use of his works in this Wikibook.

> **Sent at:** Mon Apr 20, 2009 5:56 pm
>
> You may publish my answer to the original request (and this request too). You may copy/republish any and all parts of the grsecurity documentation. I don't think I put an explicit license on the documentation, but I consider it to be essentially public domain.
>
> -Brad
>
> ---- **Sent at:** Mon Apr 20, 2009 5:27 pm
>
> Thanks!
>
> I'm making a separate page for the book that will include credits, links to the original documents and a copy of your message where you grant this permission.

1 http://grsecurity.net/gracldoc.htm
2 http://grsecurity.net/quickstart.pdf
3 http://en.wikibooks.org/wiki/User%3AMeev0
4 http://en.wikibooks.org/wiki/User%20talk%3AMeev0

Just so that there is no misunderstanding

1) May I publish your answer to my original request?

2) In my request I mentioned wanting to copy ”parts” which is very vague. Basically what's needed (IMO) is you clearly stating what parts of the grsecurity documentation can be published under the GNU Free Documentation License. I'm not a copyright lawyer, but I think the clearer the situation the better.

I'll try to limit the amount of text I need to copy, as I like writing documentation, but most of technical notes are better left as they are.

- Ville

---- **Sent at:** Sat Apr 18, 2009 7:59 pm

Of course, that's fine with me. Thanks again for your work, and hope things get better for you personally.

-Brad

---- **Sent at:** Sat Apr 18, 2009 5:41 pm

Hi Brad,

I wanted to ask about using the Grsecurity QuickStart guides (the quickstart.pdf and the newGradmDoc.pdf) in the Wikibook. As you are the copyright holder of both documents, I need your permission to copy parts from those files. Mainly I would like to copy the ACL documentation, as it would be silly for me to start writing it from scratch. Naturally I would credit you and include a link to the original documents.

Below is a link to the copyright policy of Wikibooks. `http://en.wikibooks.org/wiki/Wikibooks:Copyrights`

You can reach me by replying to this PM or by email at ***.

- Ville

11 External Links

- Official grsecurity website[1]
- Official website of the PaX project[2]

1 http://grsecurity.net/
2 http://pax.grsecurity.net/

12 Contributors

Edits	User
1	16@r[1]
14	Adrignola[2]
7	Avicennasis[3]
1	Bluefoxicy[4]
1	Gronky[5]
1	Marudubshinki[6]
2	Piotrkarbowski[7]
1	Platonides[8]
1	Pratyeka[9]
5	QuiteUnusual[10]
1	Samsara[11]
1	Sandahl[12]
1	Simetrical[13]
112	Spender2001[14]
1	Thumperward[15]
2	Xania[16]

1 https://en.wikibooks.org/wiki/User:16@r
2 https://en.wikibooks.org/wiki/User:Adrignola
3 https://en.wikibooks.org/wiki/User:Avicennasis
4 https://en.wikibooks.org/wiki/User:Bluefoxicy
5 https://en.wikibooks.org/wiki/User:Gronky
6 https://en.wikibooks.org/wiki/User:Marudubshinki
7 https://en.wikibooks.org/wiki/User:Piotrkarbowski
8 https://en.wikibooks.org/wiki/User:Platonides
9 https://en.wikibooks.org/wiki/User:Pratyeka
10 https://en.wikibooks.org/wiki/User:QuiteUnusual
11 https://en.wikibooks.org/wiki/User:Samsara
12 https://en.wikibooks.org/wiki/User:Sandahl
13 https://en.wikibooks.org/wiki/User:Simetrical
14 https://en.wikibooks.org/wiki/User:Spender2001
15 https://en.wikibooks.org/wiki/User:Thumperward
16 https://en.wikibooks.org/wiki/User:Xania

List of Figures

- GFDL: Gnu Free Documentation License. http://www.gnu.org/licenses/fdl.html
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. http://creativecommons.org/licenses/by-sa/3.0/
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. http://creativecommons.org/licenses/by-sa/2.5/
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. http://creativecommons.org/licenses/by-sa/1.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/deed.en
- cc-by-2.5: Creative Commons Attribution 2.5 License. http://creativecommons.org/licenses/by/2.5/deed.en
- cc-by-3.0: Creative Commons Attribution 3.0 License. http://creativecommons.org/licenses/by/3.0/deed.en
- GPL: GNU General Public License. http://www.gnu.org/licenses/gpl-2.0.txt
- LGPL: GNU Lesser General Public License. http://www.gnu.org/licenses/lgpl.html
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. http://artlibre.org/licence/lal/de
- CFR: Copyright free use.

- EPL: Eclipse Public License. `http://www.eclipse.org/org/documents/epl-v10.php`

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses[17]. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrower.

17 Chapter 13 on page 123

13 Licenses

13.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10. 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy

both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

13.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

13.3 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user’s computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.