Hardened kernels

for everyone

Yves-Alexis Perez

Kernel Recipes 2015

Who am I?

Yves-Alexis Perez

- Head of hardware and software security lab, ANSSI
  - platform security (x86 mainly, also ARM)
  - focus on operating systems and underlying layers (chipset/SoC, PCI Express bus, devices)
- Debian developer
  - maintainer for Xfce desktop environment and strongSwan IKE daemon
  - security team member
  - maintainer of unofficial linux-grsec package

Context

This talk is **not**

- a detailed list of grsecurity benefits

This talk **is** about

- kernel security
- kernel hardening
- Debian distribution (and derivatives)

And especially integration of all these in something suitable for end-users

The need for kernel security

- *kernel knows everything* (Michaël Kerrisk)
- kernel runs CPU in ring0 (supervisor) mode
- kernel handles security for userspace:

DAC	*Discretionary access control*
MAC	*Mandatory access control*
Namespace	process separation
IPsec	network traffic encryption
dm-crypt	disk encryption
...	

Kernel security often limited to fixing security bugs

Usually:

kernel security bug ⟹ local privilege escalation

- *most of the time* (see *Android rooting*)
- not always the case

Fixing security bugs

- preferably *before* they're exploited in the wild
- sometimes after

This is not enough!

but still mandatory, obviously

kernel hardening

Active protection against attackers

- new marketing term: *kernel self protection*
- protect the kernel from outside (user processes)
- reduce kernel attack surface

Userland security features:

Kernels already supports ”security features”

- DAC
- LSM
- Namespaces

They’re mostly targeted at userland:

- process isolation
- user isolation
- resource isolation
- policy enforcement

grsecurity

Quick reminder

- a (large, 6.2M) patch against Linux kernel
- security oriented (obviously)
- started more than 14 years ago, pioneered multiple techniques
- includes multiple parts:

PaX	protection against memory corruption bugs sometimes called an HIPS[1]
RBAC	Role-based access control (not implemented as LSM)
Other	Generic hardening (memory, filesystem, network protections)

[1] Host Intrusion Prevention System

PaX

Major features

NOEXEC segmentation or pagination-based implementation of NX bit (before it was available on CPUs)

MPROTECT W⊕X at the page level: forbid memory pages available both for write and execute, and completes NOEXEC

KERNEXEC kernel equivalent of NOEXEC+MPROTECT; prevents injection and execution of foreign code to the kernel

ASLR predates the Linux version (actually predates all version), and still an improvement

UDEREF prevents the kernel dereferencing userland pointers, like SMEP/SMAP or PXN/PAN on steroid

CONSTIFY constify structure containing only function pointers, using a gcc plugin

Debian distribution

Not much to say here I guess

- well known distribution
- maintained by volunteers
- community-driven
- lot of users
- lot of derivatives

Why?

- external patch → hard for end users
- not "secure by default"
- but no interest from grsecurity in upstreaming things
  - independence
  - global approach

Debian 7 *Wheezy*

- kernel based on 3.2
- Ben Hutchings (Debian kernel maintainer) maintains LTS version on kernel.org
- Bradley Spengler (grsecurity upstream) also chose 3.2 kernel for grsecurity *stable*

How

src:linux Debian package

- maintained in svn
- version 3.2, with on top:
  - stable kernel.org patches (patch-3.2.68)
  - bug fixes
  - security fixes
  - backports
  - Debian-specific integration
- supports *featuresets*: openVZ, RT

What's a featureset?

- additional set of patches
- separate options (.config, dependencies etc.)
- additional set of *binary* packages (-image, -headers etc.)

Grsecurity featureset

Why a featureset?

- easy to include new patches
- easy to set different options
- easy to add build-dependencies (gcc plugins)
- adds a new binary package, not installed by default

How to add a grsecurity featureset

- stack of *quilt* patches against src:linux svn tree
- maintained in a git repository [1]

grsec-patches repository content

02_force-hostcc-version force HOSTCC to CC
: gcc plugins have to be built using the host compiler, so we can build i386 on amd64

03_add-grsec-featureset define the *grsecurity* featureset

04_grsecurity grsecurity patch itself, ported to Debian kernel
: put into `debian/patches/features/all/grsec/`

series the quilt `series` file

README explains how to build the packages

Focus on 03_add-grsec-featureset

- defines the featureset
- adds the grsecurity specific kernel config
- adds the grsecurity patch to the featureset series

Build the kernel

Procedure:

1. get the linux source package

   ```
   apt-get source linux
   ```

2. apply the patches

   ```
   QUILT_PATCHES=../grsec-patches QUILT_PC=.pc-grsec quilt push -a
   ```

3. regenerate control files

   ```
   python debian/bin/gencontrol.py
   ```

4. build the kernel

   ```
   dpkg-buildpackage -us -uc
   ```

More details in the README

Status

bug report against src:linux package (#605090 [2])

- patches reviewed by Ben Hutchings and Bastian Blank (Debian kernel maintainers)
- basically NACK'ed by Bastian Blank
- git repository maintained (a bit sporadically) for *Wheezy*
- `amd64` and `i386` packages available from my repository [3]

Debian 8 *Jessie*

Jessie kernel

- based on 3.16
- 3.16 EOL on kernel.org
  LTS work done by Canonical Kernel Team for Ubuntu
- no grsecurity support for 3.16 (*stable2* is 3.14)
- porting to 3.16 not doable outside of grsecurity team

Need for a new plan.

Mempo's SameKernel [4]

What is it?

- part of Mempo project (main goal: *Hardened Privacy*)
- attempt to build the kernel in a *verifiable* (deterministic and reproducible) way
- also includes grsecurity by default

Unfortunately

- build system really complex
- lot of wrappers
- eventually calling the *deprecated* make-kpkg

Use kernel makefile

make *deb-pkg*

- simple target of linux Makefile
- generates binary Debian packages from the current tree
- exists for other distributions as well (*rpm-pkg*, *binrpm-pkg*, *tar-pkg* etc.)

How can we use it?

1. get the kernel sources from git
2. patch them with grsecurity
3. add a .config
4. run make deb-pkg

Easy enough to do a little shell script for that

New repository

`debian-grsec-config` [5]

- completely different work than the featureset
- only based on upstream/kernel.org work
- mostly distribution independent

repository content

`bin/`	various scripts
`configs/`	reference configs
`patches/`	additional patches to make my life easier
`README`	self-explanatory (read it!)

repository details

`bin/`

get-grsec runs from a local linux git clone

- gets the latest grsec patch
- creates a local branch
- applies patch to it

kconfig

- taken from Debian linux source package
- used to merge to KConfig files
- useful to keep grsecurity specific config separate

repository details

`configs/`

Various config files:

- Debian references
  `config-3.14-2-686-pae`, `config-4.1.0-1-amd64` ...
- generic *hardening* options
- grsecurity specific options

Merged with `kconfig` script to produce a config file for the build

repository details

`patches/`

Various patches against linux source tree, Debian specific

- only touch packaging
  - `scripts/package/builddeb`
  - `scripts/package/Makefile`
- add support for generating a source package
- add architecture name to the `.changes` filename
- included in 4.3-rc1

Targeted at enterprise local distributors

- easier integration into Debian-like infrastructures
- easy upload to local mirrors

Results

Dead easy procedure

1. get the last patch and apply it

   ```
   get-grsec.sh stable2
   ```

2. (regenerate the config file)

   ```
   kconfig.py .config ../config-3.14-2-amd64 ../hardening ../grsec
   ```

3. build the kernel

   ```
   make deb-pkg
   ```

4. (upload result to your local mirror)

Rinse, repeat

Source and binary packages

- available on my repository [3] for 3.14 and 4.1
- you don't really need them, it's so easy to build them yourself

Caveats

Still some issues

1. *Jessie* has 3.16, not 3.14

Caveats

Still some issues

1. *Jessie* has 3.16, not 3.14

I know, but:

1. I can't do much about it, but 3.14 works fine on *Jessie*

Caveats

Still some issues

1. *Jessie* has 3.16, not 3.14
2. resulting packages not suitable for the Debian archive

I know, but:

1. I can't do much about it, but 3.14 works fine on *Jessie*

Caveats

Still some issues

1. *Jessie* has 3.16, not 3.14
2. resulting packages not suitable for the Debian archive

I know, but:

1. I can't do much about it, but 3.14 works fine on *Jessie*
2. it's **really** easy to build them yourself!

Caveats

Still some issues

1. *Jessie* has 3.16, not 3.14
2. resulting packages not suitable for the Debian archive
3. grsecurity for 3.14 is not available anymore

I know, but:

1. I can't do much about it, but 3.14 works fine on *Jessie*
2. it's **really** easy to build them yourself!

Caveats

Still some issues

1. *Jessie* has 3.16, not 3.14
2. resulting packages not suitable for the Debian archive
3. grsecurity for 3.14 is not available anymore

I know, but:

1. I can't do much about it, but 3.14 works fine on *Jessie*
2. it's **really** easy to build them yourself!
3. baah...

Current status

grsecurity

- *stable* patches (`3.2` and `3.14`) are not public anymore
- only *test* patch is available (`4.1.7` at the time of writing)

Debian

(old)stable no solution right now:

- stay on current patches and try to port them to later kernels
- switch to *test* patches (and `4.1+` kernels)

stretch need for a more future-proof solution

Retrospectively maybe it's better that grsecurity was not included

Plans for the future

Debian integration

- separate source package
- only target *unstable*, not *testing* and *stable*
- maybe provide backports

Upstreaming

- more and more needed with the unavailability of *stable* grsecurity patches
- *Kernel self protection* discussions at the Linux Security Summit
- Kernel Hardening topic proposal for Kernel Summit [6]

src:linux-grsec

Why a new source package?

Featureset is not enough

- patch porting is too time consuming
- we need to minimize the differences with linux vanilla
- we want to follow the grsecurity timeline for *test* patches

Try to avoid duplicate work

- try to reuse src:linux repository (now switched to git)
- branch major versions
- drop all non-relevant stuff
- merge or cherry-pick specific changes

If that doesn't work, then we'll fork completely

Will it work this time?

We don't know

- don't hold your breathe...
- keep using `make deb-pkg` and `debian-grsec-config` [5] in the meantime

But

- seems to be some momentum on this
- some interested people pinged me recently about this
- so we'll see, stay tuned

References


Y.-A. Perez, “quilt set of patches against src:linux to enable grsec featureset.”
https://anonscm.debian.org/cgit/users/corsac/grsec/grsec-patches.git/.


Y.-A. Perez, “linux-2.6: [rfc] add a grsec featureset to debian kernels.”
https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=605090.


Y.-A. Perez, “Debian grsecurity kernel apt repository.”
http://perso.corsac.net/~corsac/debian/kernel-grsec/packages/.


Mempo, “Samekernel.”
https://wiki.debian.org/SameKernel.


Y.-A. Perez, “Tools to build a linux kernel with grsecurity patch for debian.”
https://anonscm.debian.org/cgit/users/corsac/grsec/debian-grsec-config.git/.


K. Cook, “[ksummit-discuss] [tech topic] kernel hardening.”
https://lists.linuxfoundation.org/pipermail/ksummit-discuss/2015-August/002292.html.