Protect your server with SELinux on SUSE Linux Enterprise Server 11 SP

Sander van Vugt
Instructor, Consultant and Author
Sandervanvugt.nl

About Sander van Vugt

- Trainer, consultant and author
- Doing much work on SUSE Linux Enterprise as well as “the other distribution” (and started to appreciate SELinux on that distribution)
- www.sandervanvugt.{com|org|nl}
- mail@sander.fr

Why would you need kernel level security anyway?

- Some things cannot be handled with permissions and firewalls alone...
  - Just updating your server doesn't protect against badly written scripts or zero-day exploits
  - Restricting on permissions isn't always feasible (you don't want to take away other permissions on /tmp for instance)
  - Firewall protection is also limited

What are your options for kernel level security?

- SELinux or AppArmor
  - AppArmor is the current default in SLES 11
  - Large market demand for SELinux
  - More vivid community for SELinux
  - SELinux is more thorough, but more complex as well
- Solutions building their own lists of syscalls to be blocked

SELinux Status in SUSE® Linux Enterprise Server 11 SP 3

- Complete Support for the stack
- Consult your SUSE support engineer to find out about policy support
- Binary support still developing
- What it comes down to: install the refpolicy, find out that it doesn't do what you need it to do and walk the long road of having it do what you need it to do

This session helps you taking the first steps

SELinux in SLES12

- AppArmor will still be the default solution
  - Many customers are using AppArmor and want to keep using it
- SELinux will be fully supported as well and a small MLS policy will be provided to protect specific parts of a system
- Developing an all-including policy is on the roadmap

Overview

What is SELinux

- All syscalls are are denied by default, unless specifically enabled
- All objects (files, ports, processes etc.) are provided with a security label (the context)
  - User, role and type part in the context
  - Type part is the most important
- The SELinux policy contains rules where you can see which source context has access to which target context

SELinux Components

- The Policy is the key component
- It defines the security contexts for type enforcements

That means: if you use **ls -Z** on a directory, you'll find a context user, role and type and the policy knows how to handle that.

- It all comes down to a line like the following that is defined in the policy:

allow user_t bin_t : file {read execute getattr};

This states that user_t (the source) has allow to bin_t (the target) on the object class file with the permissions read execute and getattr

- The SELinux Policy has tens of thousands of rules

Which makes it extremely complex

Refpolicy: mother of all policies

- The refpolicy is a generic fully functional policy managed as a free software project
- Application developers provide code for the Refpolicy, where after peer review it can be included
- Refpolicy is a common base for distributions that only need to make modifications to it
- Because of differences that in some cases are huge, making these changes can be hard

Using Policy Features

- MLS is multi level security, every object gets a security clearance label
- MCS (Multi Category Security) is MLS but less detailed
- Targeted is the “normal” policy without MLS
- Support of features is indicted by policy version, current is 28.
- Compile policy to support different options
  - How to handle unknown permissions?
  - Do or don't support unconfined domains?

Compiling the Policy

- Monolithic: One huge policy
  - Don't use it anymore!
- Modular: Different rules are groups in modules, which makes it more readable and less complicated to manage

Managing SELinux

Managing SELinux means applying context

- You can find the context you need on default objects
- From the _selinux man page (if available)
- Using **semanage fcontext -l**
- From the policy itself (may be hard)

Managing SELinux

- Setting the Appropriate Contexts

Use **semanage fcontext -l** to find available contexts

Use **semanage fcontext -a -t your_content_t "/web(/.*)?"**

Use **restorecon -R -v / to relabel**

- Setting Booleans

Use **semanage boolean -l** to get an overview of available Booleans

Use **getsebool -a** for an overview

Use **setsebool -P your_boolean** on to switch on a boolean

Understanding SELinux Modes

- Set mode while booting

Enforcing: Active and Armored

Permissive: Active, but only logging

Use this while setting up your system!

Log messages are in /var/log/audit/audit.log

Disabled

- Are there good reasons to switch to disabled?

Applications and SELinux

- Most applications are not SELinux aware
- Some applications are and they make active SELinux calls
  - These applications behave differently than expected on a system where SELinux code is active
  - Check your application with ldd
  - Most applications on SUSE are not native SELinux

Configuring SELinux on SLES

Installing SELinux on SUSE® Linux Enterprise Server

- Install all SELinux Packages

Use YaST for easy selection

- Install a Policy

Install the refpolicy source file from OpenSUSE.org

- Enable GRUB Boot Options and reboot

security=selinux selinux=1 enforcing=0

- Compile the policy and reboot

Compiling the Policy

- Extract the tarball to /etc/selinux/refpolicy
- Edit /etc/selinux/refpolicy/build.conf

DISTRO = suse

UNK_PERMS = allow

DIRECT_INITRC = y

MONOLITHIC = n

Leave other params to their default

- Run make load from /etc/selinux/refpolicy
- Reboot
- Fix a small bug and run make relabel for file system labelling
  - semanage fcontext -a -e /dev /lib/udev/devices
  - restorecon -Rv /

At This Point: Verify the Installation

- Use **sestatus -v**

Shows current mode and process / file contexts

- Use semanage boolean -l

Lists all boolean switches to change policy settings and at the same time verifies access to the policy

- Use semanage fcontext -l

Lists all file contexts and verifies policy access

Tuning Policy Modules

About Policy Modules

- Policy modules provide the base chunks of SELinux functionality
  - **semodule -l**
- Shut off all rules for a part of the system by unloading the corresponding policy (not very useful)
  - semodule -d
- Or compile a new module that allows all you need to accomplish: use **audit2allow**

Modifying Policies manually

- Input files are in /etc/selinux/refpolicy/policy/modules/services

*.te files contain everything a module needs to have

*.if files define how other policy modules get access to this policy

*.fc files contain the labeling instructions

- After modifying the input files, run make && make install && make load to activate them

Using Audit2allow

- Audit2allow helps analyzing the log messages in /var/log/audit/audit.log

audit2allow -w -a presents the audit information in a more readable way

audit2allow -a shows the type enforcement rule that allows the denied access

audit2allow -a -M blah creates a te file and a compiled pp file that will allow the denied access

- After creating the new module, use **semodule -i modulename.pp** to execute it
- **Be careful when using audit2allow**

Disabling SELinux for a single service

- **semanage permissive -a somelabel_t**
- Get an overview of current permissive domains using **semanage permissive -l**
- Not supported on all environments

More information

- Refpolicy project: http://oss.resys.com/projects/refpolicy
- Dan Walsh blog: http://danwalsh.liveblog.com
- Sven Vermeulen, SELinux System Administration (ISBN 978-1-78328-317-0)

Questions?

Thank you.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org