SELinux Troubleshooting

Milos Malik

mmalik (at) redhat (dot) com

Senior Quality Engineer

June 2018

Agenda

- Introduction
- Reasons
- Basic principles
- Troubleshooting workflow
- Documentation
- SELinux troubleshooting chart

Reality check

- Have you heard about SELinux?
- Do you use SELinux?
- Which distribution do you use?
- Do you use SELinux in enforcing mode?
- Have you reported at least 1 selinux-policy bug?
- Have you resolved at least 1 selinux-policy issue?

SELinux policy - complex system?

Entities / OS	RHEL-7.5	Fedora 28
Types	~4700	~4800
Process types	816	833
File types	~2900	~3000
Allow rules	~106000	~109000
Type transition rules	~66000	~232000
Dontaudit rules	~10000	~10000
Policy modules	408	415
Booleans	311	319

Bugs resolved

since 2012-01-01 (not counting duplicates)

	selinux-policy	user-space
Fedora	~3500	~300
RHELs	~2400	~370

User-space components: checkpolicy, libsepol, libsemanage, libselinux, policycoreutils, setools, setroubleshoot, setroubleshoot-plugins, mcstrans

Reasons

- Provide users a guidance
- Reduce the time between finding the bug and resolving it
- Teach users new skills
- Improve collaboration among reporters, developers and QE
- Make the SELinux adoption easier

Basic SELinux principles

- Everything is denied by default
- Policy defines rules which allow various accesses
- Everything has a label
- Labels are inherited
- Labels can change during creation / execution

Almost all SELinux problems

fall into one of the following categories:

- labeling problem
- confined process behaves in a different way than what default SELinux configuration expects
- bug in SELinux policy or in the application
- your machine has been compromised

Troubleshooting workflow

Problem identification

Best friends are:

- auditd
- ausearch
- dmesg
- full path in audit messages

SELinux denial

type=PROCTITLE msg=audit(05/29/2018 22:34:31.817:1776) : proctitle=smbcontrol smbd ping
type=MMAP msg=audit(05/29/2018 22:34:31.817:1776) : fd=7 flags=MAP_SHARED
type=SYSCALL msg=audit(05/29/2018 22:34:31.817:1776) : arch=x86_64 syscall=mmap success=no exit=EACCES(Permission denied) a0=0x0 a1=0x2b8 a2=PROT_READ|PROT_WRITE a3=MAP_SHARED items=0 ppid=15427 pid=21481 auid=root uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=10 comm=smbcontrol exe=/usr/bin/smbcontrol subj=unconfined_u:unconfined_r:smbcontrol_t:s0-s0:c0.c1023 key=(null)
type=AVC msg=audit(05/29/2018 22:34:31.817:1776) : avc: denied { map } for pid=21481 comm=smbcontrol path=/var/lib/samba/lock/names.tdb dev="vda3" ino=9394097 scontext=unconfined_u:unconfined_r:smbcontrol_t:s0-s0:c0.c1023 tcontext=unconfined_u:object_r:samba_var_t:s0 tclass=file permissive=0

Problem analysis

Best friends are:

- matchpathcon
- sesearch
- audit2allow
- sealert reports

Finding policy rules

SELinux is a labeling system. First thought should be "Is there a label that would make this work?"

```
# sesearch -s smbcontrol_t -t samba_var_t -c file -p map --allow
#
```

Troubleshooting workflow

Conservative solution

Best friends are:

- chcon
- semanage
- restorecon

Radical solution

Best friends are:

- audit2allow
- semodule
- sepolicy

Please read the documentation

- SELinux User's and Administrator's Guide
- Security Guide
- SELinux project page
- CentOS SELinux wiki
- Fedora SELinux wiki
- Gentoo SELinux wiki
- Debian SELinux wiki
- The SELinux Notebook 4th edition!
- SELinux Troubleshooting Chart on github.com

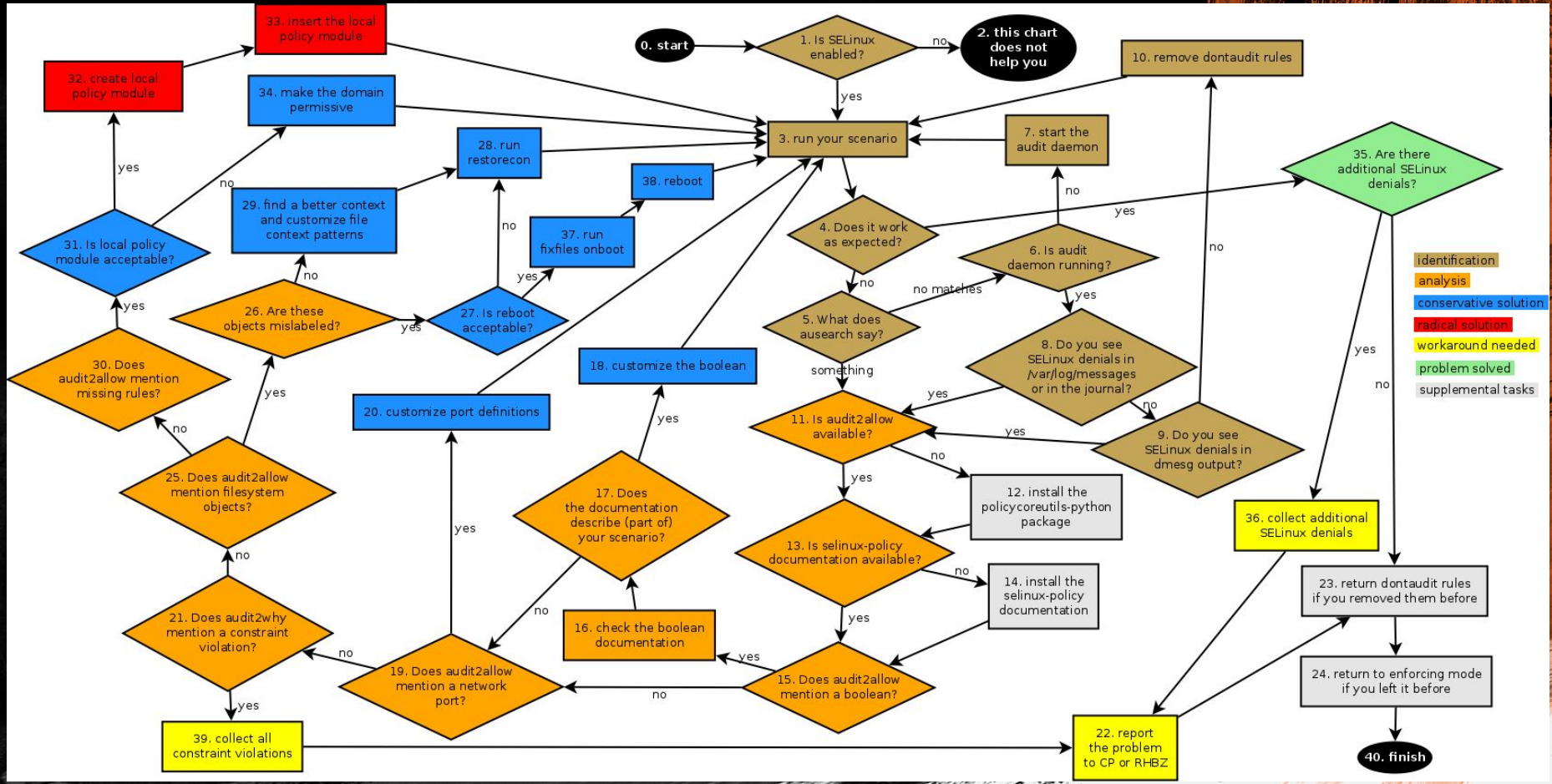
Allow everything? Rather not

Dangerous operations:

- execmod, execstack, execheap, execmem
- sys_ptrace, sys_module, sys_admin
- writing to base SELinux types (e.g. etc_t, usr_t)
- module_request
- writing to admin_home_t
- writing to modules_object_t
- mac_admin

Operations for further inspection:

- accessing unlabeled_t files, dirs, devices etc.
- dac_read_search, dac_override
- net_admin and other capabilities


0. start
1. Is SELinux enabled?
no
2. this chart does not help you
yes
3. run your scenario
4. Does it work as expected?
5. What does ausearch say?
6. Is audit daemon running?
7. start the audit daemon
8. Do you see SELinux denials in /var/log/messages or in the journal?
9. Do you see SELinux denials in dmesg output?
10. remove dontaudit rules
11. Is audit2allow available?
12. install the policycoreutils-python package
13. Is selinux-policy documentation available?
14. install the selinux-policy documentation
15. Does audit2allow mention a boolean?
16. check the boolean documentation
17. Does the documentation describe (part of) your scenario?
18. customize the boolean
19. Does audit2allow mention a network port?
20. customize port definitions
21. Does audit2why mention a constraint violation?
22. report the problem to CP or RHBZ
23. return dontaudit rules if you removed them before
24. return to enforcing mode if you left it before
25. Does audit2allow mention filesystem objects?
26. Are these objects mislabeled?
27. Is reboot acceptable?
28. run restorecon
29. find a better context and customize file context patterns
30. Does audit2allow mention missing rules?
31. Is local policy module acceptable?
32. create local policy module
33. insert the local policy module
34. make the domain permissive
35. Are there additional SELinux denials?
36. collect additional SELinux denials
37. run fixfiles onboot
38. reboot
39. collect all constraint violations
40. finish
identification
analysis
conservative solution
radical solution
workaround needed
problem solved
supplemental tasks
something
no matches

redhat.