BIND 9 Security

(Part 1 - SELinux on RedHat based Linux distributions)

Carsten Strotmann and the ISC Team

Welcome

Welcome to part one of our BIND 9 security webinar series

In this Webinar

- Linux security modules
- What is SELinux
- SELinux label on files and processes
- Fixing SELinux file permission issues
- Fixing SELinux port/socket permission issues
- SELinux Boolean switches
- SELinux troubleshooting
- SELinux Hands-On lab

All content © 2021 Internet Systems Consortium, Inc.

Linux security modules

LSM (Linux Security Modules)

- LSMs are extensions of the Linux kernel.
- The Linux kernel defines interfaces into which the LSMs can hook into:
  - Syscalls
  - File accesses
  - Process creation
  - Namespaces and cgroups
  - User identity (UID/GID)
  - ...

LSM (Linux Security Modules)

- A LSM module can plug into one or more of these interfaces
- If an application (such as the BIND 9 DNS server) uses a kernel function, the LSM becomes active and will allow or disallow the kernel function based on the rules in the LSM policy.
- Link: A Brief Tour of Linux Security Modules

Major LSMs

- Mandatory Access Controls - only one of these LSMs can (currently) be enabled in the Linux kernel (at any given time)
  - SELinux
  - AppArmor
  - SMACK (Simplified Mandatory Access Control Kernel)
  - TOMOYO
- This webinar does cover *SELinux*. The next webinar will cover the other security modules.

Minor LSMs

- Minor LSMs can be activated in addition to the major LSMs and other minor LSMs:
  - YAMA
  - LoadPin
  - SafeSetID
  - Lockdown
  - Landlock
  - BPF - LSM security policies can be enforced using eBPF
  - capabilities - Linux capabilities

Which LSMs are available in a Linux system?

- You can check the available Linux security modules available in your system in the file `/sys/kernel/security/lsm`:

```
# cat /sys/kernel/security/lsm
lockdown,capability,yama,tomoyo,bpf
```

What is SELinux

SELinux history

- SELinux (Security Enhanced Linux) is a Linux Security Module (LSM) that implements mandatory access control (MAC) inside a Linux system
  - SELinux was originally developed by the United States National Security Agency (NSA)
  - It was merged in the mainline Linux-Kernel 2.6.0 in 2003
  - SELinux consists of two major parts
    - The kernel code and the utilities (generic and available on many Linux systems)
    - The SELinux policies (must be adapted for each Linux distribution)

DAC and MAC

- Traditionally, Unix and Linux systems implement discretionary access control (DAC)
  - DAC is implemented via the Unix file permissions (read-write-execute for owner/group/world)

```
drwxr-xr-x  147 user  admin      4704 Sep  5 22:04 src
```

  - The owner of a file or directory can change the permissions

SELinux implements MAC

- SELinux enables Mandatory Access Control (MAC) in **addition** to the DAC available in a Linux system
  - SELinux MAC has higher preference than DAC
  - In addition to file access, SELinux does also control access to network sockets, processes, name-spaces, user- and group ids and system calls (Kernel functions)
  - Access to an object must be permitted by MAC and DAC

Benefits of SELinux

- SELinux allows to define a fine grained policy for processes
  - The processes are not able to access files or manipulate other processes outside the rules
  - The SELinux system can prevent privilege escalation through security vulnerabilities in software
    - Either that the BIND 9 process is accessing files that do not belong to a BIND 9 configuration, or any other SELinux confined software to access the BIND 9 files

SELinux Policy

- SELinux implements a modular security policy
  - This security policy is independent of the Unix/Linux file permissions
  - The Policy is enforced by the Linux Kernel
    - Applications that violate the policy will by denied access

Targeted vs. Full mode

- SELinux can be deployed in two different modes:
  - Full mode: all files, users and processes are subject to the SELinux policy
  - Targeted mode: only selected files, users and processes are under the control of the SELinux policy. All other objects are `unconfined` (normal Unix permissions apply)

Multilevel Security and RBAC

- SELinux can also used to implement
  - Multilevel Security - creating up to 1024 "security levels" for users. Users of a lower level cannot access content created by a user on a higher level
  - Role-Based-Access-Control - Users can gain extra privileges by switching *roles* in the system
- We don't discuss these SELinux features in this webinar, as they are not part of the BIND 9 SELinux policy

SELinux and Linux Distributions

- A SELinux policy ruleset needs to be adapted for each Linux distribution
  - The different Linux-Distributions have varying support for SELinux:
    - RedHat / CentOS / Fedora / Alma / Rocky - adapted targeted policy
    - Suse - basic targeted policy
    - Ubuntu - basic targeted policy
    - Debian (11) - basic targeted policy
    - Arch Linux - work in progress, based on reference policy

All content © 2021 Internet Systems Consortium, Inc.

SELinux basic administration

SELinux status

- The command `sestatus` gives information about the SELinux function of the system

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      32
```

Enable/Disable SELinux

- SELinux can be in three modes:
  - `disabled`: The SELinux modules and policies are not loaded and not enforced, SELinux labels are not created on new files
  - `permissive`: The SELinux modules and policies are loaded, but they are not enforced. Policy violations will be logged through the audit subsystem. New files and processes will get SELinux labels
  - `enforcing`: The SELinux system is fully loaded and the policy will be enforced. New files and processes will get SELinux labels

Enable/Disable SELinux

- A Linux system with SELinux switched off is missing an important security function
  - If you encounter an issue with SELinux, try to fix the issue without disabling SELinux

Enable/Disable SELinux

- On RedHat EL8 (and older), SELinux modes can be switched at run-time by an administrator
  - `setenforce 1` will set SELinux into *enforcing* mode
  - `setenforce 0` will set SELinux into *permissive* mode (turn SELinux security off)
    - Fedora Linux 34+ (and possibly the next Red Hat Enterprise Linux version) does not allow to change the SELinux mode in a running system.

Enable/Disable SELinux

- The Kernel parameter `selinux=0` will disable SELinux completely (requires reboot)
- SELinux modes can be switched in the file `/etc/selinux/config`

```
# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
[...]
```

List SELinux modules available

- SELinux is a modular system
  - Modules can be loaded or unloaded by an administrator with the correct permissions
- Red Hat based Linux systems come with a SELinux module for BIND 9
- List all SELinux modules of a running system:

```
# semodule -l
abrt
accountsd
acct
afs
aiccu
[...]
bind
[...]
```

Modules can be selectively disabled/enabled

- To disable just the BIND 9 SELinux module

```
semodule -d bind
```

- To enable the BIND 9 SELinux module

```
# semodule -ve bind
Attempting to enable module 'bind':
Ok: return value of 0.
Committing changes:
Ok: transaction number 6.
```

SELinux man pages

- SELinux modules come with (automatically generated) man pages
  - These man pages are not installed by default on an Red Hat/CentOS system
  - They can be added from the SELinux policy sources

```
dnf install -y selinux-policy-devel
sepolicy manpage -a -p /usr/share/man/man8
```

- While the SELinux module is called `bind`, the manpage is called `named_selinux`. This manpage documents the `named` process types, which besides BIND 9 is also used for the Unbound resolver:

```
man named_selinux
```

List SELinux labels on files and directories

- SELinux controls access to files through the SELinux file label
  - The file label are stored in extended attributes on the file-system
  - SELinux can only secure files that are stored in file-systems that support extended attributes

BIND Configuration files

- Files with the label `named_conf_t` are for BIND 9 configuration files and can only be read by the BIND 9 processes

```
# ls -lZ /etc/named.conf
-rw-r-----. 1 root named system_u:object_r:named_conf_t:s0 1705 May 27 20:49 /etc/named.conf
# ls -lZ /etc/named.rfc1912.zones
-rw-r-----. 1 root named system_u:object_r:named_conf_t:s0 1029 May 27 20:49 /etc/named.rfc1912.zones
```

- File of type `etc_t` (general Linux configuration files under `/etc`) can also be read

```
[root@bind9-selinux ~]# ls -lZ /etc/named.root.key
-rw-r--r--. 1 root named system_u:object_r:etc_t:s0 1070 May 27 20:49 /etc/named.root.key
```

Zone-Files

- BIND 9 zone-files are labeled as type `named_zone_t`. By default, the BIND 9 processes can read and write these files.
  - With the switch `named_write_master_zones` (see below) write access to these files can be forbidden

```
# ls -lZ /var/named/
total 16
drwxrwx---. 2 named named system_u:object_r:named_cache_t:s0   23 Sep 16 18:39 data
drwxrwx---. 2 named named system_u:object_r:named_cache_t:s0   60 Sep 16 18:40 dynamic
-rw-r-----. 1 root  named system_u:object_r:named_conf_t:s0  2253 May 27 20:49 named.ca
-rw-r-----. 1 root  named system_u:object_r:named_zone_t:s0   152 May 27 20:49 named.empty
-rw-r-----. 1 root  named system_u:object_r:named_zone_t:s0   152 May 27 20:49 named.localhost
-rw-r-----. 1 root  named system_u:object_r:named_zone_t:s0   168 May 27 20:49 named.loopback
drwxrwx---. 2 named named system_u:object_r:named_cache_t:s0    6 May 27 20:49 slaves
```

Journal and Dump-files

- Files of type `named_cache_t` are *dynamic* files and can be read and written by the BIND 9 processes

```
# ls -lZ /var/named/data/
total 4
-rw-r--r--. 1 named named system_u:object_r:named_cache_t:s0 443 Sep 16 18:39 named.run
```

- This includes dynamic zones and journal files (including automatically managed DNSSEC signed zones)

```
# ls -lZ /var/named/dynamic/
total 8
-rw-r--r--. 1 named named system_u:object_r:named_cache_t:s0 821 Sep 16 18:40 managed-keys.bind
-rw-r--r--. 1 named named system_u:object_r:named_cache_t:s0 512 Sep 16 18:40 managed-keys.bind.jnl
```

List SELinux labels on processes

- BIND 9 running with SELinux enabled, the `named` process is labeled as `named_t`:

```
# ps auxZ | grep named
system_u:system_r:named_t:s0    named      62221  0.0  7.0 265260 58248 ?
        Ssl  Sep16   0:00 /usr/sbin/named -u named -c /etc/named.conf
```

List SELinux labels on processes

- SELinux module for BIND 9 is not active, the `named` process is running as type `unlabeled_t`:

```
# ps auxZ | grep named
system_u:object_r:unlabeled_t:s0 named     62221  0.0  7.1 265260 59316 ?
        Ssl  Sep16   0:00 /usr/sbin/named -u named -c /etc/named.conf
```

List SELinux labels on processes

- A BIND 9 process running outside the default Red Hat directory structure is not secured by SELinux
  - This compiled version of BIND 9 running from `/opt/bind/sbin/named` is labeled `unconfined_t`

```
# ps auxZ | grep named
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 named 111228 0.8  2.3 183672 19736 ?
       Ssl 03:53   0:00 /opt/bind/sbin/named -u named -c /etc/named.conf
```


All content © 2021 Internet Systems Consortium, Inc.

SELinux troubleshooting

Linux Audit Subsystem

- SELinux policy violations are logged using the Linux Audit Subsystem
- The command `ausearch` can be used to list the policy violation of a specific process
  - `-m avc` list LSM policy violations
  - `-x /usr/sbin/named` filter for violations of this process
  - `-i` (interpret) print the data in human readable form

```
# ausearch -m avc -x /usr/sbin/named -i
----
type=PROCTITLE msg=audit(09/17/2021 04:06:05.836:2278) : proctitle=/usr/sbin/named -u named -c /etc/nam
type=SYSCALL msg=audit(09/17/2021 04:06:05.836:2278) : arch=x86_64 syscall=bind success=no exit=EACCES(
type=AVC msg=audit(09/17/2021 04:06:05.836:2278) : avc:  denied  { name_bind } for  pid=111615 comm=isc
```

File Type Label mismatch

- The SELinux system will deny the BIND 9 processes access to zone- or configuration files if the file labels are not correct
- Reasons for wrong or missing file label
  - The Linux system has been run with SELinux disabled
  - The files are located in a non-default directory (not in `/etc` and `/var/named`)
  - The files have been created in a non-default directory and then moved into the correct directory. The file labels are assigned during creation of a file and will not change if moved on the same file-system.

Finding the correct label

- The command `matchpathcon` (Match Path Context) will report files where the file label does not match the SELinux policy
  - It will also report the expected file label types

```
# matchpathcon -V /var/named/named.localhost
/var/named/named.localhost has context system_u:object_r:etc_t:s0, should be system_u:object_r:named_zo
```

Changing the file label

- The command `chcon` (change SELinux context) can be used to set the file label type:

```
chcon --type named_cache_t /var/named/zonefile.db
```

Applying the correct label from the policy

- The command `restorecon` will adjust the label on a file so that it matches the label expected by the SELinux policy

```
# restorecon -v /var/named/named.localhost
Relabeled /var/named/named.localhost from system_u:object_r:etc_t:s0 to system_u:object_r:named_zone_t:
```

Adjust the expected file context on a single file

- If BIND 9 configuration or zone files are stored in a non-default location, the SELinux policy should be adjusted to include the correct context label
  - The command `semanage fcontext -a` will add a file context label to the SELinux policy
    - It will not automatically relabel the files
    - Use `restrorecon` to relabel the files

```
# semanage fcontext -a -t named_zone_t /srv/bind/zones/primary/example.com.db
# restorecon -vr /srv/bind/zones
Relabeled /srv/bind/zones/primary/example.com.db from unconfined_u:object_r:var_t:s0 to unconfined_u:ob
```

Adjusting the file context recursively for all files and directories

- New SELinux file context can be added recursively
  - All new files created in the specified directories will automatically get the correct SELinux file label

```
# semanage fcontext -a -t named_zone_t --ftype f "/srv/bind/zones(/.*)?"
# semanage fcontext -a -t named_zone_t --ftype d "/srv/bind/zones(/.*)?"
# semanage fcontext -a -t named_cont_t --ftype f "/srv/bind/conf(/.*)?"
# semanage fcontext -a -t named_conf_t --ftype d "/srv/bind/conf(/.*)?"
```

Tweaking the BIND SELinux module

- SELinux modules can be tweaked with the help of Boolean switches
- All switches of all running SELinux modules can be listed with

```
# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
[...]
named_tcp_bind_http_port --> off
named_write_master_zones --> on
[...]
```

Switch named_tcp_bind_http_port

- BIND 9 can expose the statistics channel over http
  - Configuration of the statistics channel in `named.conf`

```
statistics-channels {
 inet 192.0.2.0 port 8053 allow { localnets; };
};
```

BIND 9 HTTP Port

- SELinux will deny BIND 9 to listen on port 8053

```
# ausearch -m avc -ts recent  -i
----
type=PROCTITLE msg=audit(09/17/2021 04:06:05.836:2278) : proctitle=/usr/sbin/named -u named -c /etc/nam
type=SYSCALL msg=audit(09/17/2021 04:06:05.836:2278) : arch=x86_64 syscall=bind success=no
    exit=EACCES(Permission denied) a0=0x15 a1=0x7f5183d24660 a2=0x10 a3=0x7f5183d244fc
    items=0 ppid=111613 pid=111615 auid=unset uid=named gid=named euid=named suid=named
    fsuid=named egid=named sgid=named fsgid=named tty=(none) ses=unset
    comm=isc-worker0000 exe=/usr/sbin/named subj=system_u:system_r:named_t:s0 key=(null)
type=AVC msg=audit(09/17/2021 04:06:05.836:2278) : avc:  denied  { name_bind } for  pid=111615
    comm=isc-worker0000 src=8053 scontext=system_u:system_r:named_t:s0
    tcontext=system_u:object_r:unreserved_port_t:s0
    tclass=tcp_socket permissive=0
```

Setting the Boolean to allow access to the http ports

- Switching the SELinux Boolean `named_tcp_bind_http_port` will allow access to ports that are defined for type `http_port_t`

```
# setsebool named_tcp_bind_http_port=on
```

- But port `8053` is not among these ports

```
# semanage port -l | grep http_port_t
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Solving the statistics port issue

- Solution A: use a port already permitted by `http_port_t`

```
statistics-channels {
 inet * port 8008 allow { any; };
};
```

- Solution B: add the statistics port to the list of allowed ports in `http_port_t`

```
# semanage port -a -t http_port_t -p tcp 8053
# semanage port -l | grep http_port_t
http_port_t                    tcp      8053, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Switch named_write_master_zones

- The Boolean SELinux variable `named_write_master_zones` controls if the BIND 9 processes are permitted to write zone files (files with the context label `named_zone_t`)
  - This switch is set to `on` by default, writing to zone files is enabled
  - This is required for secondary servers, as well as for dynamic zones

Disabling write access to master zones

- To enhance security, this Boolean can be switched off on a primary authoritative BIND 9 server with purely static zones

```
# setsebool named_write_master_zones=off
```

Resources

- A Brief Tour of Linux Security Modules https://www.starlab.io/blog/a-brief-tour-of-linux-security-modules/
- SELinux Struggles with BIND Startup https://www.isc.org/blogs/selinux-struggles-bind/
- All-Seeing Eye or Blind Man? Understanding the Linux Kernel Auditing System https://www.sans.org/white-papers/38605/
- `named_selinux` Manual page https://linux.die.net/man/8/named_selinux (the man page on your system is likely more up-to-date)

All content © 2021 Internet Systems Consortium, Inc.

Next webinars

- October 20 - Securing BIND 9 with AppArmor/Firejail/SecompBPF
- November 16 - Instrumenting BIND 9 on Linux with BCC/eBPF
- December 15 - DNS Fragmentation: Real-World measurements, impact and mitigations

All content © 2021 Internet Systems Consortium, Inc.

Questions and Answers

All content © 2021 Internet Systems Consortium, Inc.

Hands-On

- We have prepared a VM machine for every participant
- This time the sessions does not build upon each other and do not need to be done in order
- find the instructions at https://webinar.defaultroutes.de/webinar/06-selinux-workshop.html

All content © 2021 Internet Systems Consortium, Inc.