AppArmor Update 2018

2018 Linux Security Summit – North America

Presentation by
John Johansen
john.johansen@canonical.com
www.canonical.com
August 2018
New Logo
Moved from launchpad to gitlab
Wiki moved to gitlab too
AppArmor

Projects that follow the best practices below can voluntarily self-certify and show that they’ve achieved a Core Infrastructure Initiative (CII) badge.

If this is your project, please show your badge status on your project page! The badge status looks like this:

---

These are the **Basic** level criteria. You can also view the **Medium** or **Gold** level criteria.

---

### Identification

What is the human-readable name of the project? [Show details]

- **AppArmor**

What is a brief description of the project?

AppArmor is an effective and easy-to-use Linux application security system. AppArmor proactively protects the operating system and applications from external or internal threats, even zero-day attacks, by enforcing good behavior and preventing even unknown application flaws from being exploited.

AppArmor security policies completely define what system resources individual applications can access, and with what privileges. A number of default policies are included with AppArmor, and using a combination of advanced static analysis and learning-based tools, AppArmor policies for even very complex applications can be deployed successfully in a matter of hours.

What is the URL for the project (as a whole)?

https://github.com/apparmor/apparmor/wiki/home

What is the URL for the version control repository (it may be the same as the project URL)?

https://github.com/apparmor

What programming language(s) are used to implement the project? [Show details]

- C, C++, Python, bash, perl, Make

What is the Common Platform Enumeration (CPE) name for the project (if it has one)? [Show details]

(Optional: CPE name)
Everything except `af_unix`
Upstreaming cont.

- Secids – 4.18
- audit rule filtering (SUBJ_ROLE) – 4.18
- socket mediation – 4.17
- Profile attachment – 4.17
  - IMA
    - Improved overlapping exec attachment resolution
  - np subset test
4.14
A New Direction
Policy tagged with ABI info

```plaintext
profile ping /{usr/,}bin/ping {
    include <abstractions/base>
    include <abstractions/consoles>
    include <abstractions/nameservice>

    capability net_raw,
    capability setuid,
    network inet raw,
    network inet6 raw,

    file mixr /{,usr/}bin/ping,
    file r /etc/modules.conf,
```
feature-abi=<features/upstream-4.18>

profile ping /{usr/,}bin/ping {
    include <abstractions/base>
    include <abstractions/consoles>
    include <abstractions/nameservice>

    capability net_raw,
    capability setuid,
    network inet raw,
    network inet6 raw,

    file mixr /{,usr/}bin/ping,
    file r /etc/modules.conf,
Single Binary Policy Cache

/etc/apparmor.d/cache

bin.ping
sbin.klogd
sbin.syslogd
sbin.syslog-ng
skype
usr.bin.evince
usr.bin.firefox
usr.bin.pidgin
usr.sbin.cupsd
usr.sbin.dnsmasq
usr.sbin.dovecot
...

...
Per Kernel binary policy

<table>
<thead>
<tr>
<th>$(location)/7f01cf2e.0</th>
<th>$(location)/cache/7f01cf2e.1</th>
<th>$(location)/cache/a035ea11.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin.ping</td>
<td>bin.ping</td>
<td>bin.ping</td>
</tr>
<tr>
<td>sbin.klogd</td>
<td>sbin.klogd</td>
<td>sbin.klogd</td>
</tr>
<tr>
<td>sbin.syslogd</td>
<td>sbin.syslogd</td>
<td>sbin.syslogd</td>
</tr>
<tr>
<td>sbin.syslog-ng</td>
<td>sbin.syslog-ng</td>
<td>sbin.syslog-ng</td>
</tr>
<tr>
<td>skype</td>
<td>skype</td>
<td>skype</td>
</tr>
<tr>
<td>usr.bin.evince</td>
<td>usr.bin.evince</td>
<td>usr.bin.evince</td>
</tr>
<tr>
<td>usr.bin.firefox</td>
<td>usr.bin.firefox</td>
<td>usr.bin.firefox</td>
</tr>
<tr>
<td>usr.bin.pidgin</td>
<td>usr.bin.pidgin</td>
<td>usr.bin.pidgin</td>
</tr>
<tr>
<td>usr.sbin.cupsd</td>
<td>usr.sbin.cupsd</td>
<td>usr.sbin.cupsd</td>
</tr>
<tr>
<td>usr.sbin.dnsmasq</td>
<td>usr.sbin.dnsmasq</td>
<td>usr.sbin.dnsmasq</td>
</tr>
<tr>
<td>usr.sbin.dovecot</td>
<td>usr.sbin.dovecot</td>
<td>usr.sbin.dovecot</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Location 1</td>
<td>Location 2</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>$(loc1)/7f01cf2e.0</td>
<td>$(loc2)/7f01cf2e.0</td>
<td></td>
</tr>
<tr>
<td>skype</td>
<td>bin.ping</td>
<td></td>
</tr>
<tr>
<td>usr.bin.evince</td>
<td>sbin.klogd</td>
<td></td>
</tr>
<tr>
<td>usr.bin.firefox</td>
<td>sbin.syslogd</td>
<td></td>
</tr>
<tr>
<td>usr.sbin.cupsd</td>
<td>sbin.syslog-ng</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>skype</td>
<td></td>
</tr>
<tr>
<td>$(loc1)/a035ea11.0</td>
<td>$(loc2)/a035ea11.0</td>
<td></td>
</tr>
<tr>
<td>skype</td>
<td>bin.ping</td>
<td></td>
</tr>
<tr>
<td>usr.bin.evince</td>
<td>sbin.klogd</td>
<td></td>
</tr>
<tr>
<td>usr.bin.firefox</td>
<td>sbin.syslogd</td>
<td></td>
</tr>
<tr>
<td>usr.bin.pidgin</td>
<td>sbin.syslog-ng</td>
<td></td>
</tr>
<tr>
<td>usr.sbin.cupsd</td>
<td>skype</td>
<td></td>
</tr>
<tr>
<td>usr.sbin.dnsmasq</td>
<td>usr.bin.evince</td>
<td></td>
</tr>
<tr>
<td>usr.sbin.dovecot</td>
<td>usr.bin.firefox</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>usr.sbin.cupsd</td>
<td></td>
</tr>
</tbody>
</table>
|...

...
WIP
Current WIP

- Internal cleanups and improvements
- Rework early policy loading
  - Systemd integration
  - Default profile
  - initrd/initramfs hooks
- Fine grained networking
  - af_unix
  - ipv4/ipv6
- Improved mount mediation
- Missing mediation
  - Keys mediation
  - ioctl mediation
WIP continued

- Improvements to auditing
  - Get audit data off the stack
  - Caching and grouping
- Improvements to complain/learning
  - Caching of recently audited events
  - Direct to daemon logging
  - Daemon interaction
- Further attachment conditionals (user, ...)
- Extended conditionals, and permissions
- Policy namespaces
  - Separate scope & view work
  - Open up policy to users and applications
- Delegation
Questions please

Thank you

John Johansen
john.johansen@canonical.com
www.canonical.com