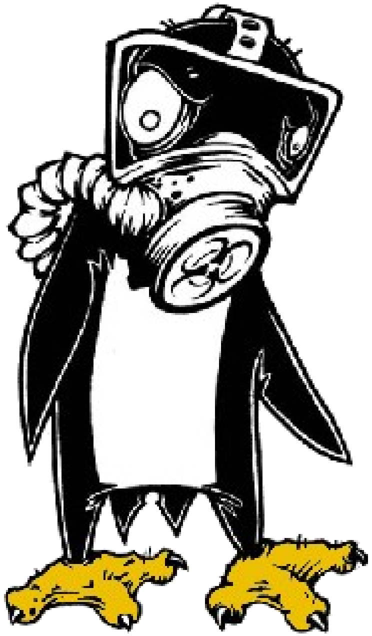
# AppArmor Update 2018

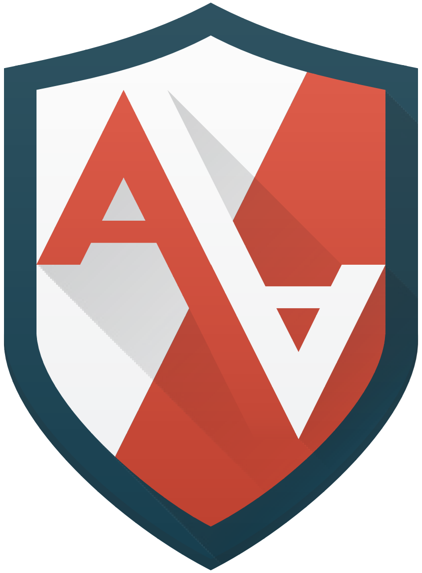
## 2018 Linux Security Summit – North America

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

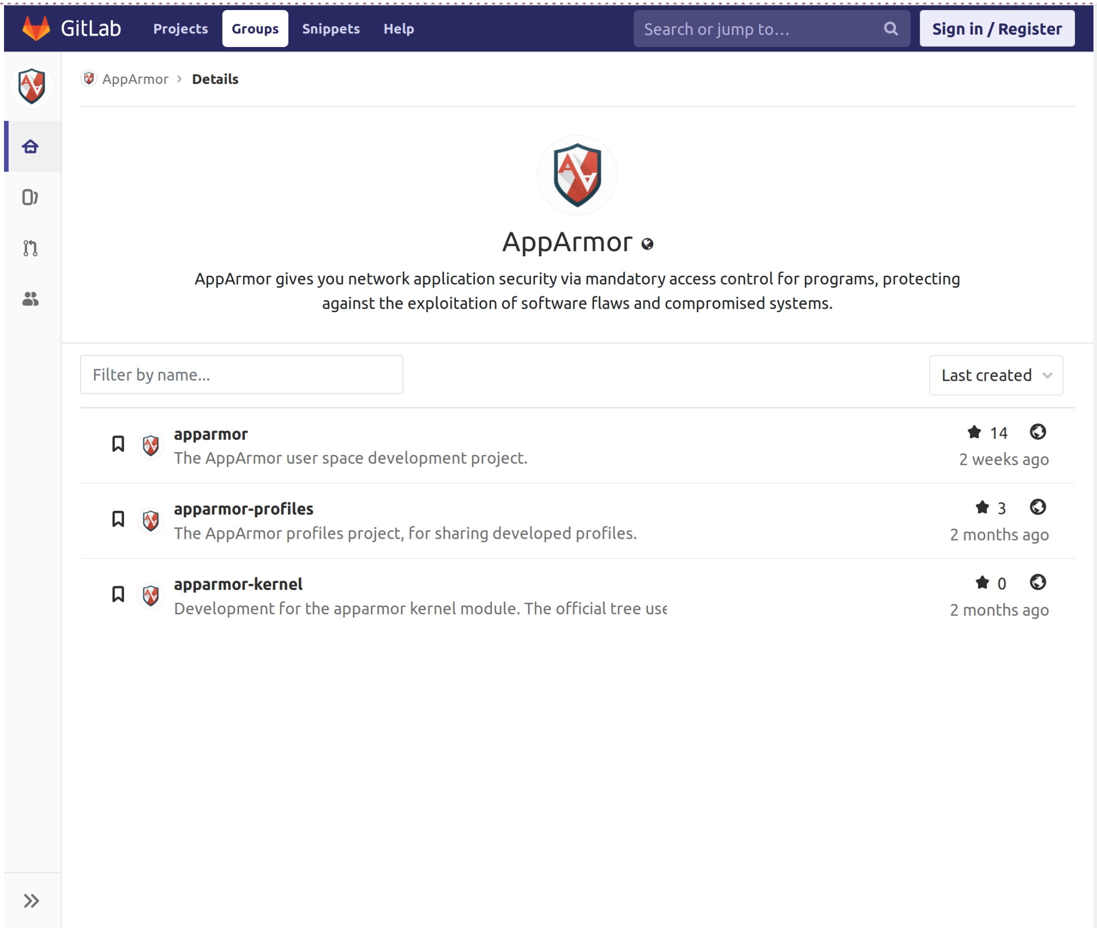
August 2018

# New Logo





# Moved from launchpad to gitlab



AppArmor

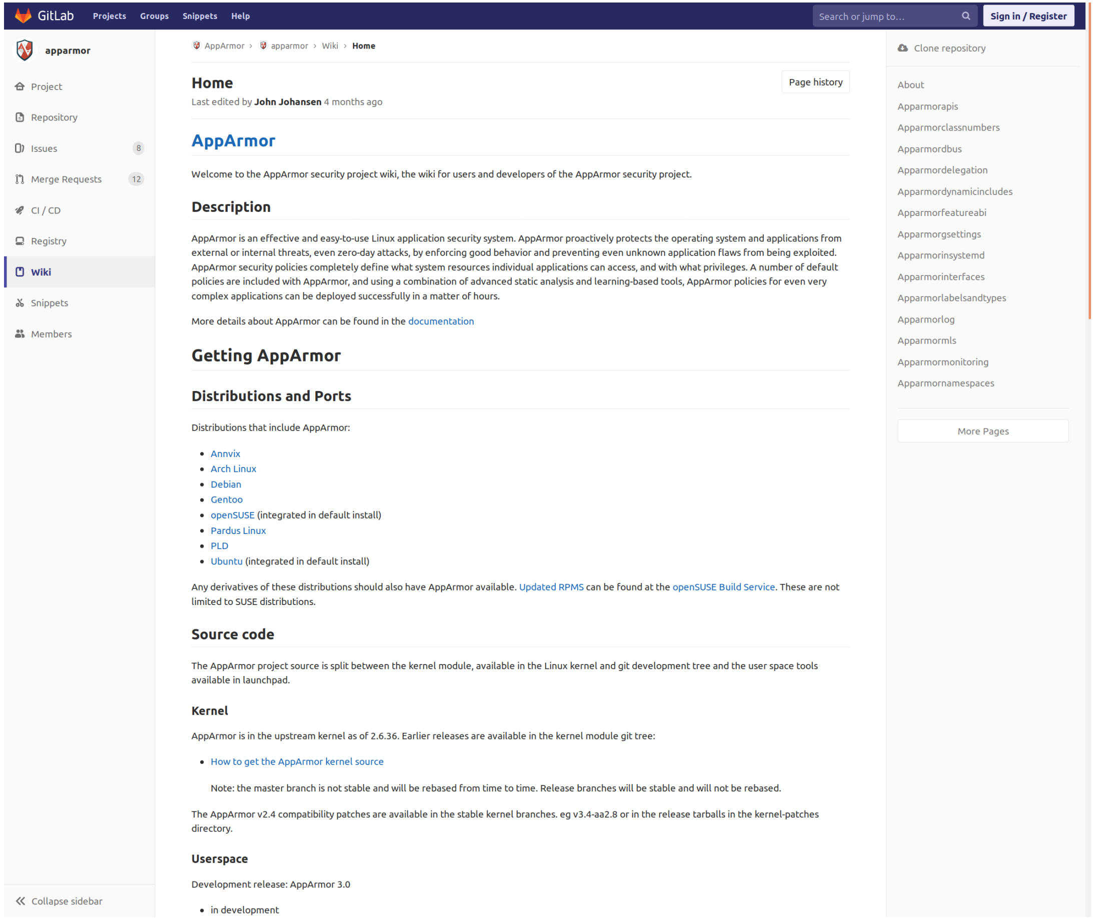
AppArmor gives you network application security via mandatory access control for programs, protecting against the exploitation of software flaws and compromised systems.

**apparmor**
The AppArmor user space development project.
★ 14
2 weeks ago

**apparmor-profiles**
The AppArmor profiles project, for sharing developed profiles.
★ 3
2 months ago

**apparmor-kernel**
Development for the apparmor kernel module. The official tree use
★ 0
2 months ago

# Wiki moved to gitlab too



## Home

Last edited by **John Johansen** 4 months ago

## AppArmor

Welcome to the AppArmor security project wiki, the wiki for users and developers of the AppArmor security project.

### Description

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.

More details about AppArmor can be found in the documentation

## Getting AppArmor

### Distributions and Ports

Distributions that include AppArmor:

- Annvix
- Arch Linux
- Debian
- Gentoo
- openSUSE (integrated in default install)
- Pardus Linux
- PLD
- Ubuntu (integrated in default install)

Any derivatives of these distributions should also have AppArmor available. Updated RPMS can be found at the openSUSE Build Service. These are not limited to SUSE distributions.

### Source code

The AppArmor project source is split between the kernel module, available in the Linux kernel and git development tree and the user space tools available in launchpad.

#### Kernel

AppArmor is in the upstream kernel as of 2.6.36. Earlier releases are available in the kernel module git tree:

- How to get the AppArmor kernel source

  Note: the master branch is not stable and will be rebased from time to time. Release branches will be stable and will not be rebased.

The AppArmor v2.4 compatibility patches are available in the stable kernel branches. eg v3.4-aa2.8 or in the release tarballs in the kernel-patches directory.

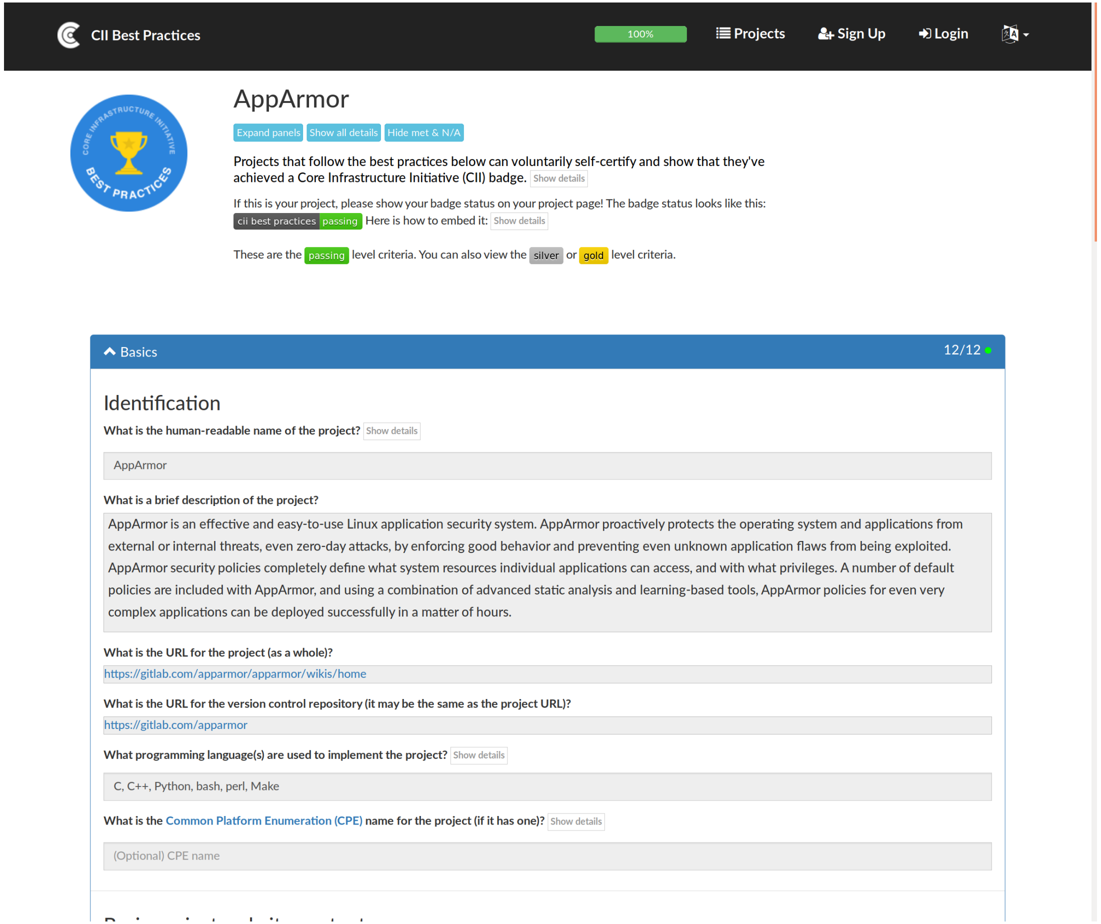
#### Userspace

Development release: AppArmor 3.0

- in development

CANONICAL

# CII Best Practices



## AppArmor

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

If this is your project, please show your badge status on your project page! The badge status looks like this: cii best practices passing Here is how to embed it: Show details

These are the passing level criteria. You can also view the silver or gold level criteria.

Basics 12/12 ●

### 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://gitlab.com/apparmor/apparmor/wikis/home

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

https://gitlab.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

# Upstreaming

Everything except

af_unix

# Upstreaming cont.

- Secids – 4.18
- audit rule filtering (SUBJ_ROLE) – 4.18
- socket mediation – 4.17
- Profile attacment – 4.17
    - IMA
    - Improved overlapping exec attachment resolution
    - nnp subset test

# 4.14
# A New Direction

# Policy tagged with ABI info

```
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,
```

# Policy tagged with ABI info

```
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

| *$(location)*/7f01cf2e.0 | *$(location)*/cache/7f01cf2e.1 | *$(location)*/cache/a035ea11.0 |
|---|---|---|
| bin.ping | bin.ping | bin.ping |
| sbin.klogd | sbin.klogd | sbin.klogd |
| sbin.syslogd | sbin.syslogd | sbin.syslogd |
| sbin.syslog-ng | sbin.syslog-ng | sbin.syslog-ng |
| skype | skype | skype |
| usr.bin.evince | usr.bin.evince | usr.bin.evince |
| usr.bin.firefox | usr.bin.firefox | usr.bin.firefox |
| usr.bin.pidgin | usr.bin.pidgin | usr.bin.pidgin |
| usr.sbin.cupsd | usr.sbin.cupsd | usr.sbin.cupsd |
| usr.sbin.dnsmasq | usr.sbin.dnsmasq | usr.sbin.dnsmasq |
| usr.sbin.dovecot | usr.sbin.dovecot | usr.sbin.dovecot |
| ... | ... | ... |

# Binary Policy Overlay

| $(loc1)/7f01cf2e.0 | $(loc2)/7f01cf2e.0 | $(loc1)/a035ea11.0 | $(loc2)/a035ea11.0 |
|---|---|---|---|
| | bin.ping | | bin.ping |
| | sbin.klogd | | sbin.klogd |
| | sbin.syslogd | | sbin.syslogd |
| | sbin.syslog-ng | | sbin.syslog-ng |
| skype | skype | skype | skype |
| usr.bin.evince | usr.bin.evince | usr.bin.evince | usr.bin.evince |
| usr.bin.firefox | usr.bin.firefox | usr.bin.firefox | usr.bin.firefox |
| | usr.bin.pidgin | | usr.bin.pidgin |
| usr.sbin.cupsd | usr.sbin.cupsd | usr.sbin.cupsd | usr.sbin.cupsd |
| | usr.sbin.dnsmasq | | usr.sbin.dnsmasq |
| | usr.sbin.dovecot | | usr.sbin.dovecot |
| ... | ... | ... | ... |

# 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