SMACK-BASED APPLICATION WHITELISTING ON AGL

Che-Hao Liu
Industrial Technology Research Institue, Taiwan

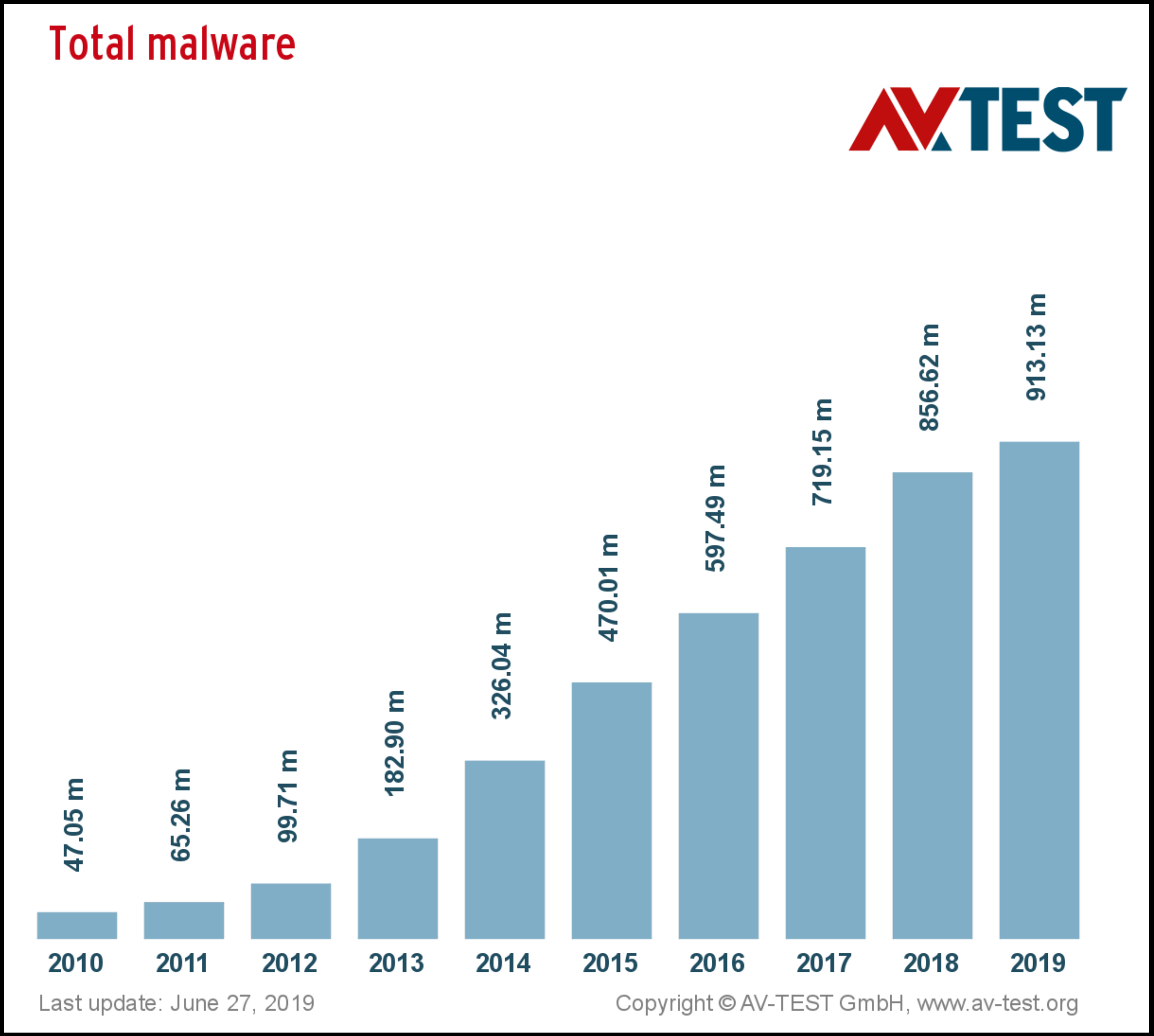
WHO AM I ?

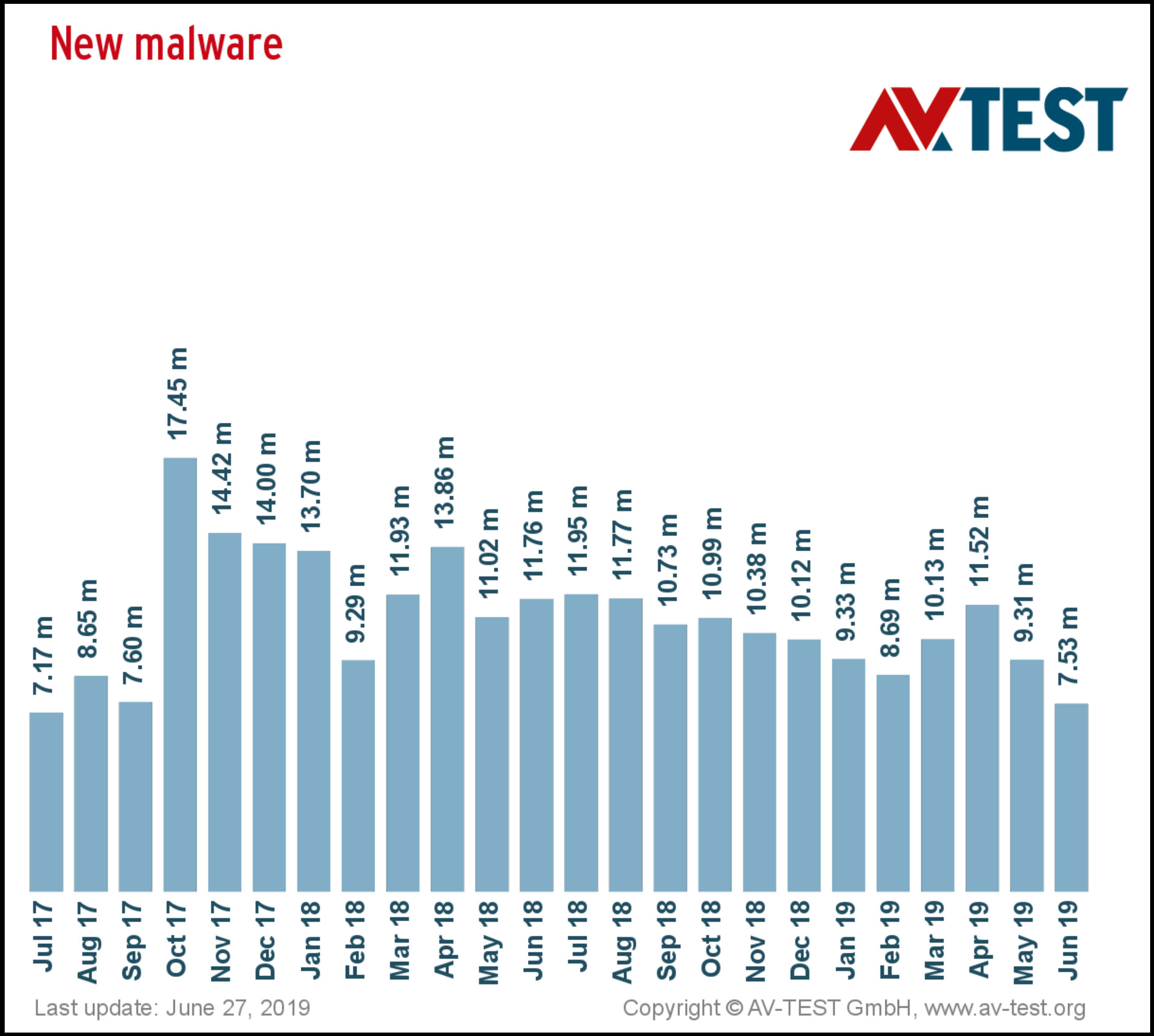
- Che-Hao Liu 劉哲豪
- Engineer at Industrial Technology Research Institute (ITRI) in Taiwan
- Security team focus on system security
- PhD candidate at Advanced Defense Lab in National Central University in Taiwan
- 2018 HITCON CTF second place in Taiwan

WHAT IS APPLICATION WHITELIST ?

- Kind of access control policy
- Opposite of blacklist
- Only allow applications listed in whitelist executed
- Deny applications not in whitelist executed

WHY NEED APPLICATION WHITELIST ?





https://www.av-test.org/en/statistics/malware/

- Application whitelisting is one of the most effective strategies in ensuring the security of systems

  – Implementing Application Whitelisting, Australian Cyber Security Centre

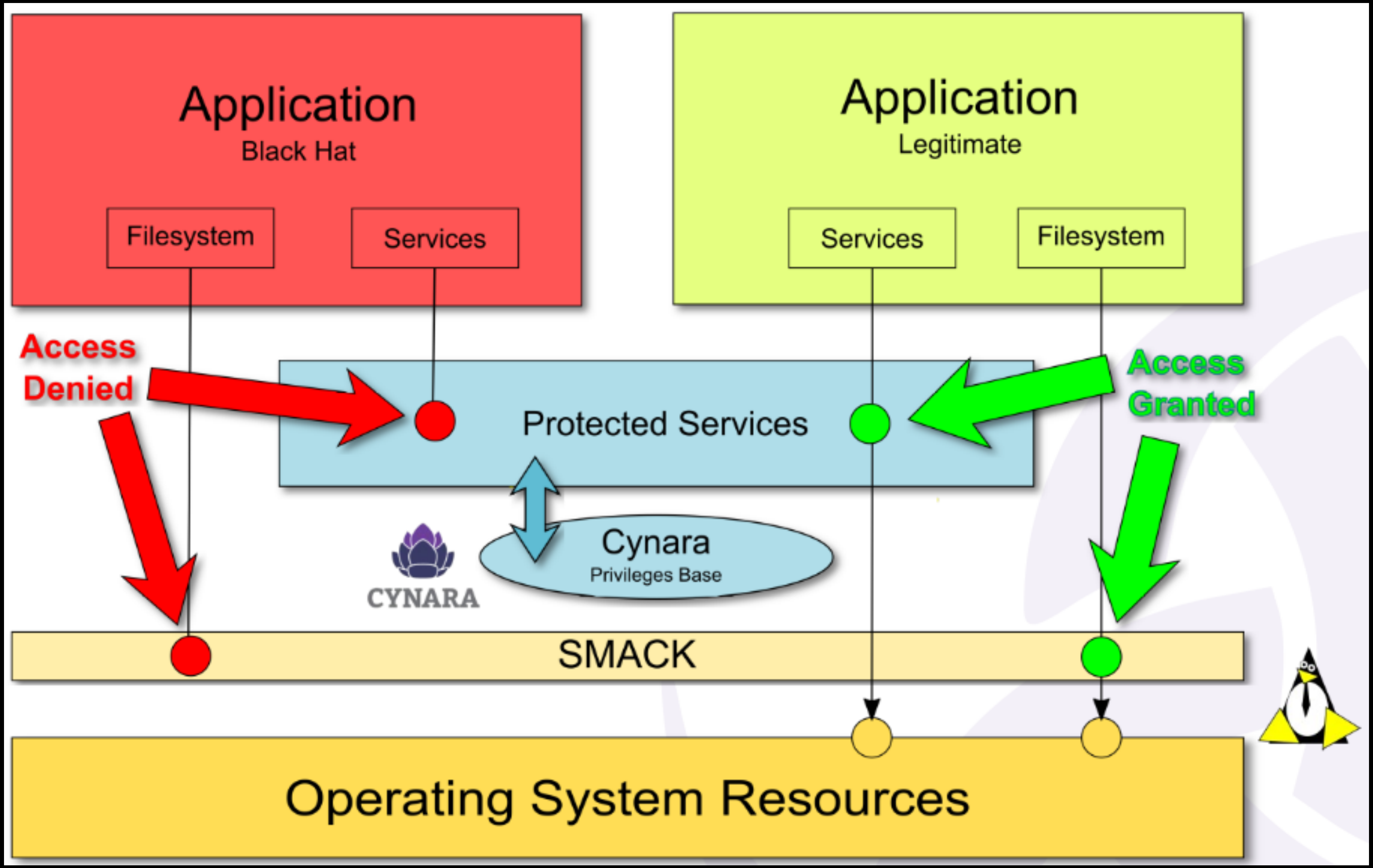
- Application whitelisting is most readily used to stop threats on managed hosts where users are not able to install or run applications without authorization

  – Guide to Application Whitelisting, National Institute of Standards and Technology

GOAL

- Propose an application whitelist prototype on AGL
- 4 features:
  - Block binary execute
  - Block library load
  - Block interpreter script
  - Block kernel module load

AGL SECURITY ARCHITECTURE



https://iot.bzh/download/public/2016/genivi/CyberSecurity-Genivi-2016-IOT-BZH-Fulup.pdf

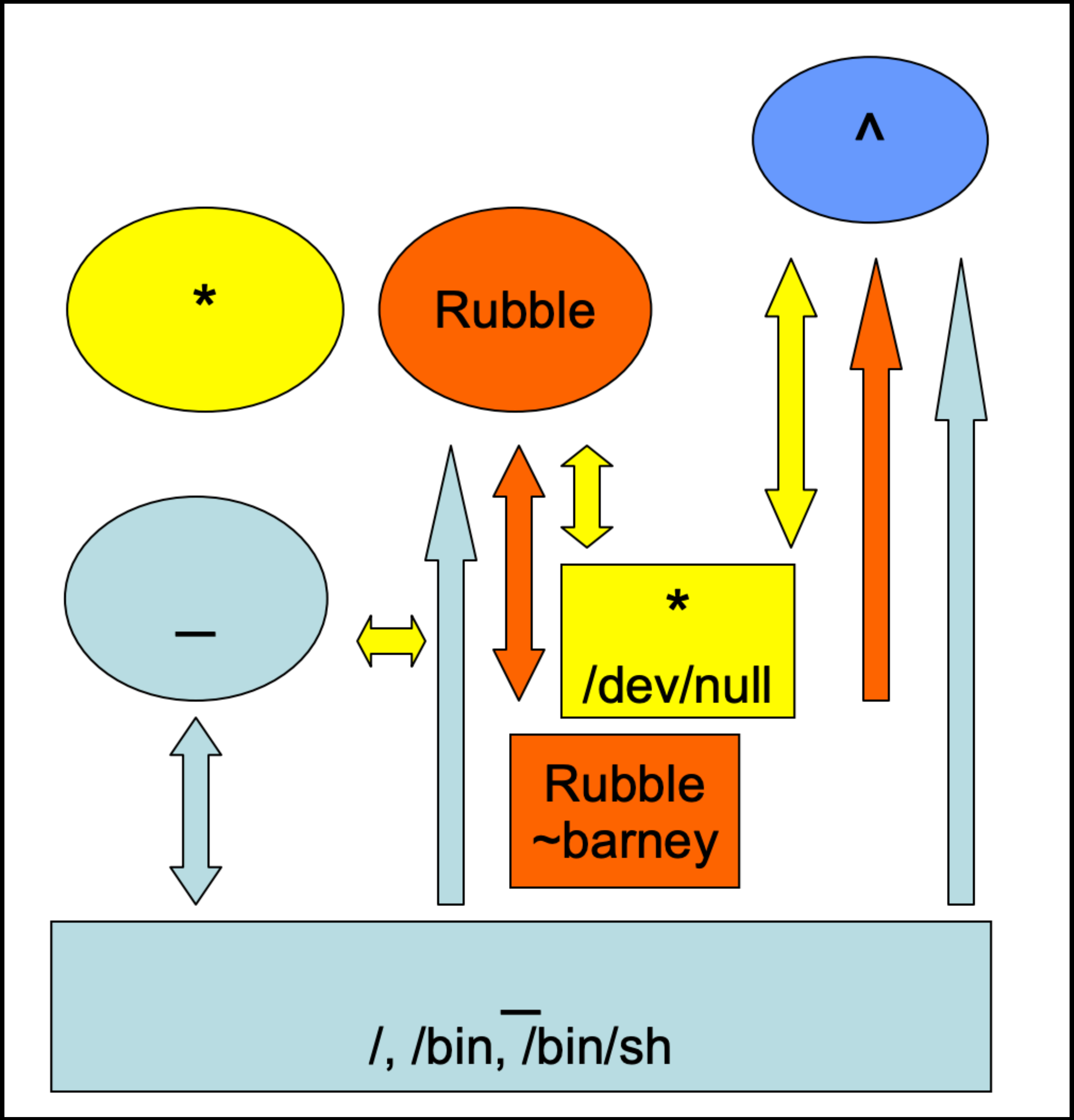
SMACK

- What is SMACK ?
  - Simplified Mandatory Access Control Kernel
  - Linux Security Modules
  - MAC model

SMACK

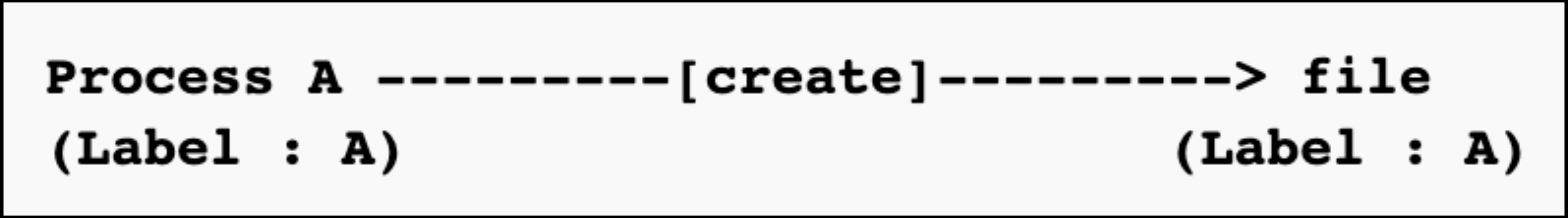
- Use extended attributes to store label
- Use label as identity
  - 6 extended attributes use to store label
    - e.g. SMACK64, SMACK64MMAP…
  - Subject Object rwxat
- There are predefine labels and rules written in kernel code

- SMACK default label & default rules



http://schaufler-ca.com/yahoo_site_admin/assets/docs/SmackWhitePaper.257153003.pdf

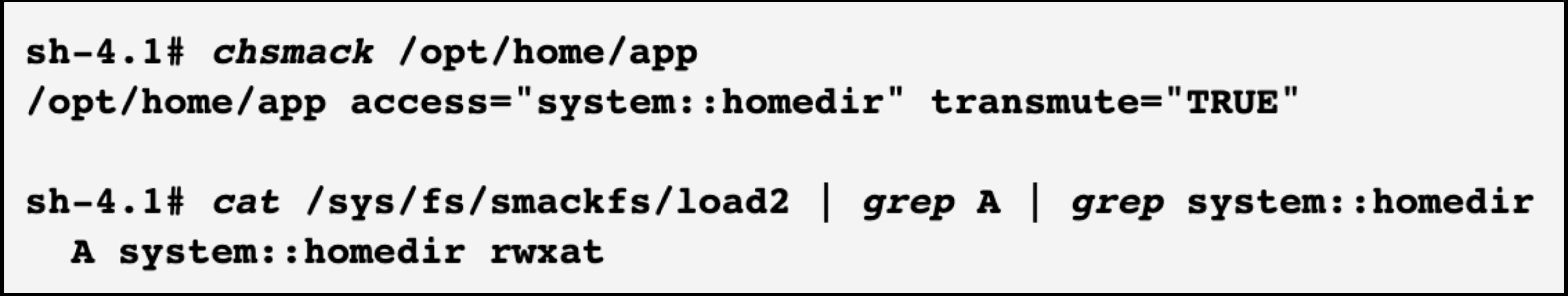
- What label will be set when create new file?
  - Without transmute



```
Process A ----------[create]----------> file
(Label : A)                            (Label : A)
```

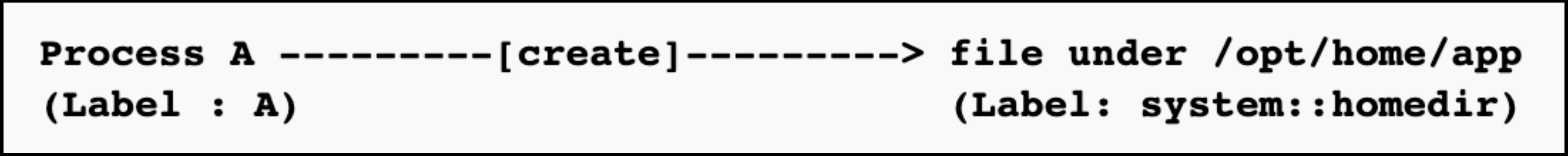
https://wiki.tizen.org/Security/Tizen_2.X_Smack_Developer_Guide

- What label will be set when create new file?
  - With transmute



```
sh-4.1# chsmack /opt/home/app
/opt/home/app access="system::homedir" transmute="TRUE"

sh-4.1# cat /sys/fs/smackfs/load2 | grep A | grep system::homedir
  A system::homedir rwxat
```

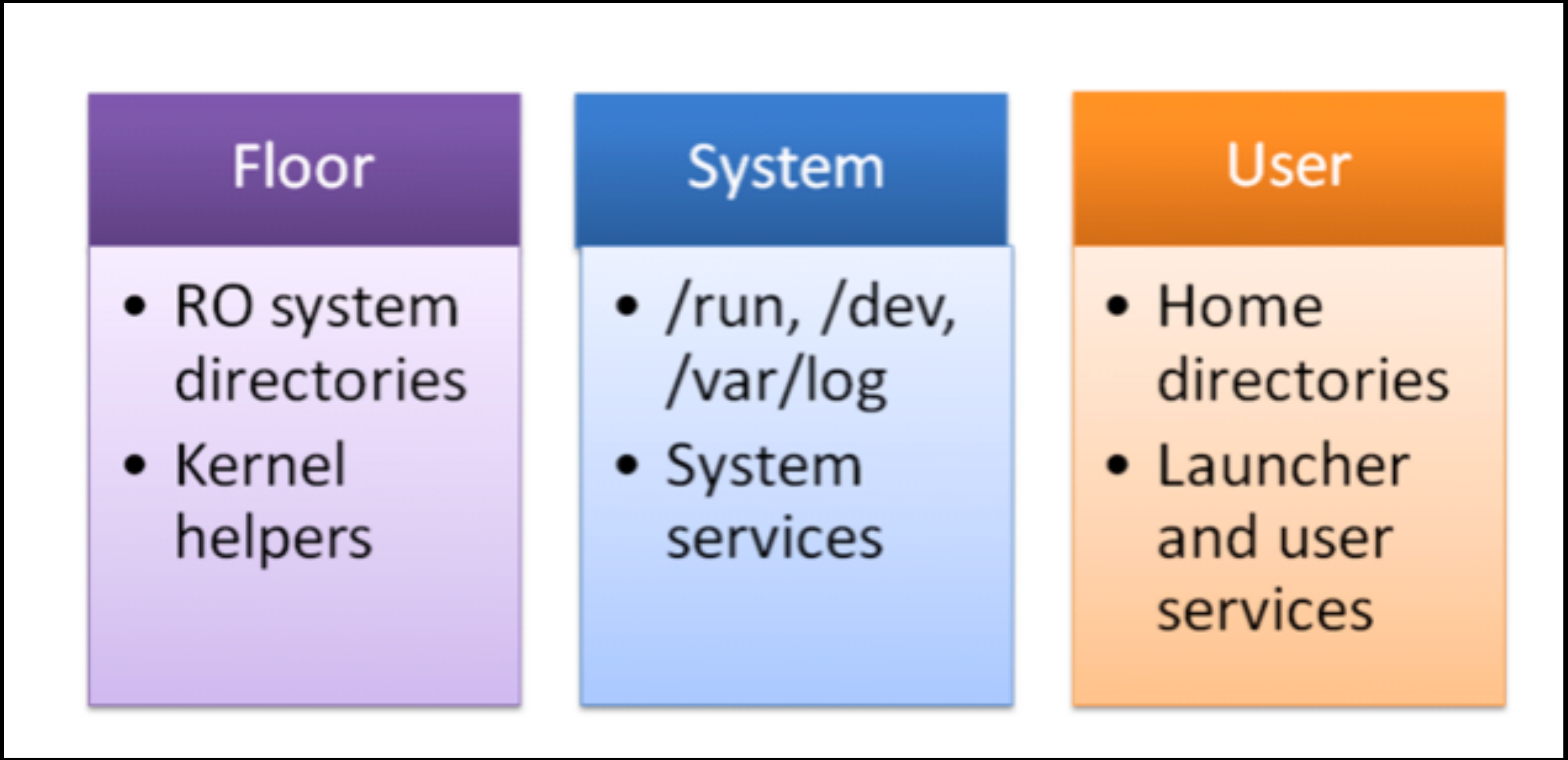


```
Process A ---------[create]---------> file under /opt/home/app
(Label : A)                           (Label: system::homedir)
```

https://wiki.tizen.org/Security/Tizen_2.X_Smack_Developer_Guide

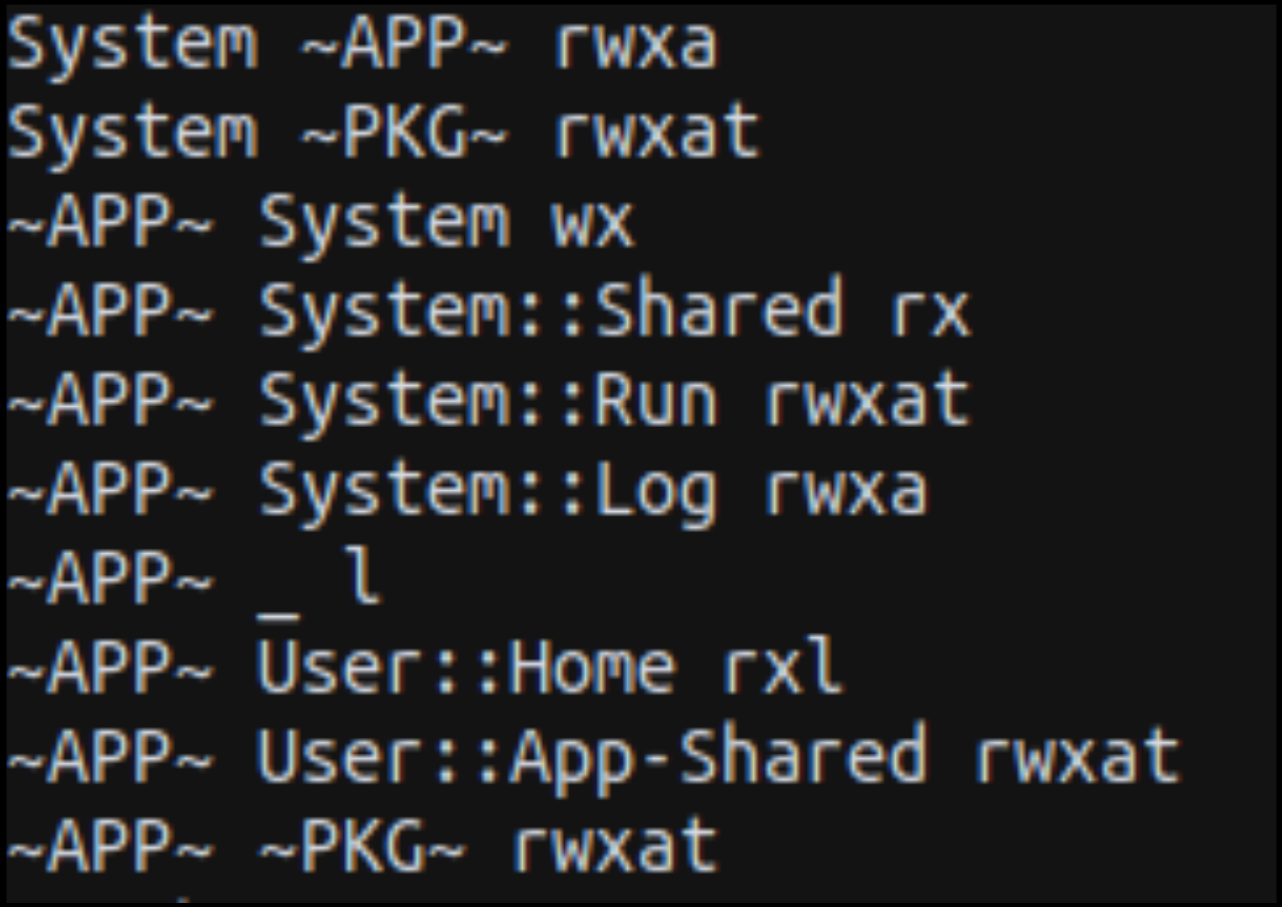
SMACK ON AGL

- Three domain model
  - Floor
  - System
  - User



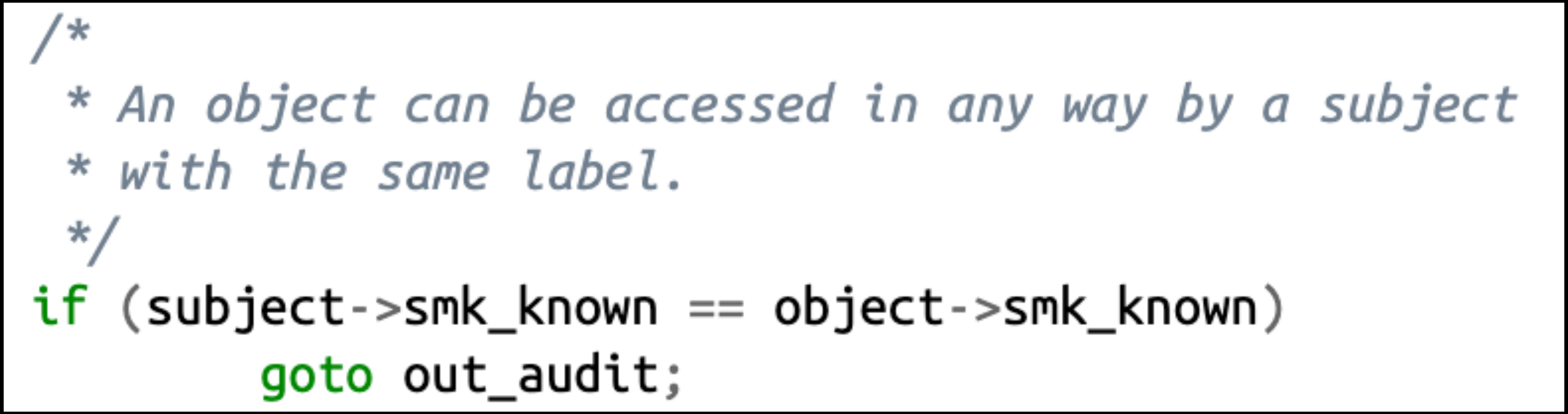
https://wiki.tizen.org/Security/Tizen_3.X_Overview

- SMACK rule template of APP in AGL
  - ~APP~ according to APP id
  - e.g. User::APP::my_app



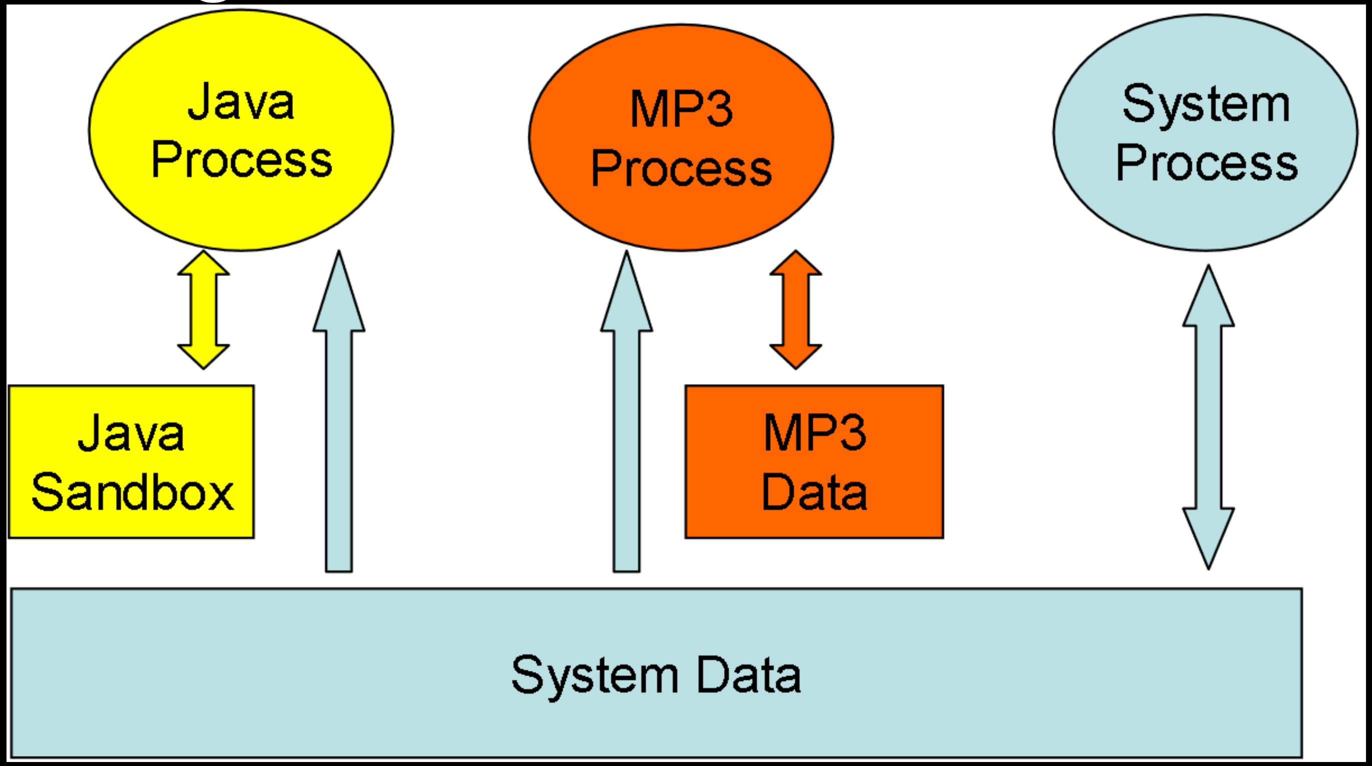
```
System ~APP~ rwxa
System ~PKG~ rwxat
~APP~ System wx
~APP~ System::Shared rx
~APP~ System::Run rwxat
~APP~ System::Log rwxa
~APP~ _ l
~APP~ User::Home rxl
~APP~ User::App-Shared rwxat
~APP~ ~PKG~ rwxat
```

- Limitation of SMACK based application whitelist
    - Subject has full access to object with same label
    - Could not change by setting rule



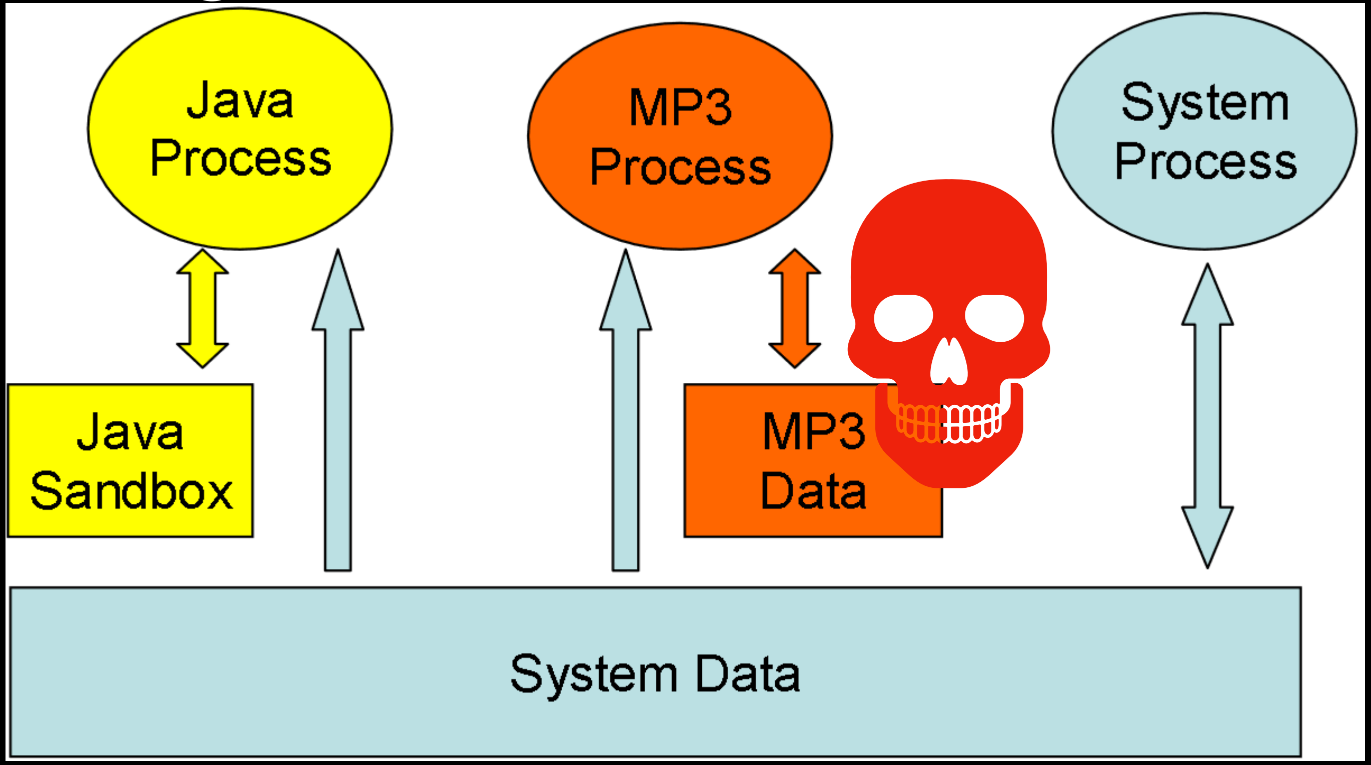
```
/*
 * An object can be accessed in any way by a subject
 * with the same label.
 */
if (subject->smk_known == object->smk_known)
        goto out_audit;
```

- Limitation of SMACK based application whitelist
  - Subject has full access to object with same label
  - Could not change by setting rule



http://schaufler-ca.com/yahoo_site_admin/assets/docs/SmackWhitePaper.257153003.pdf

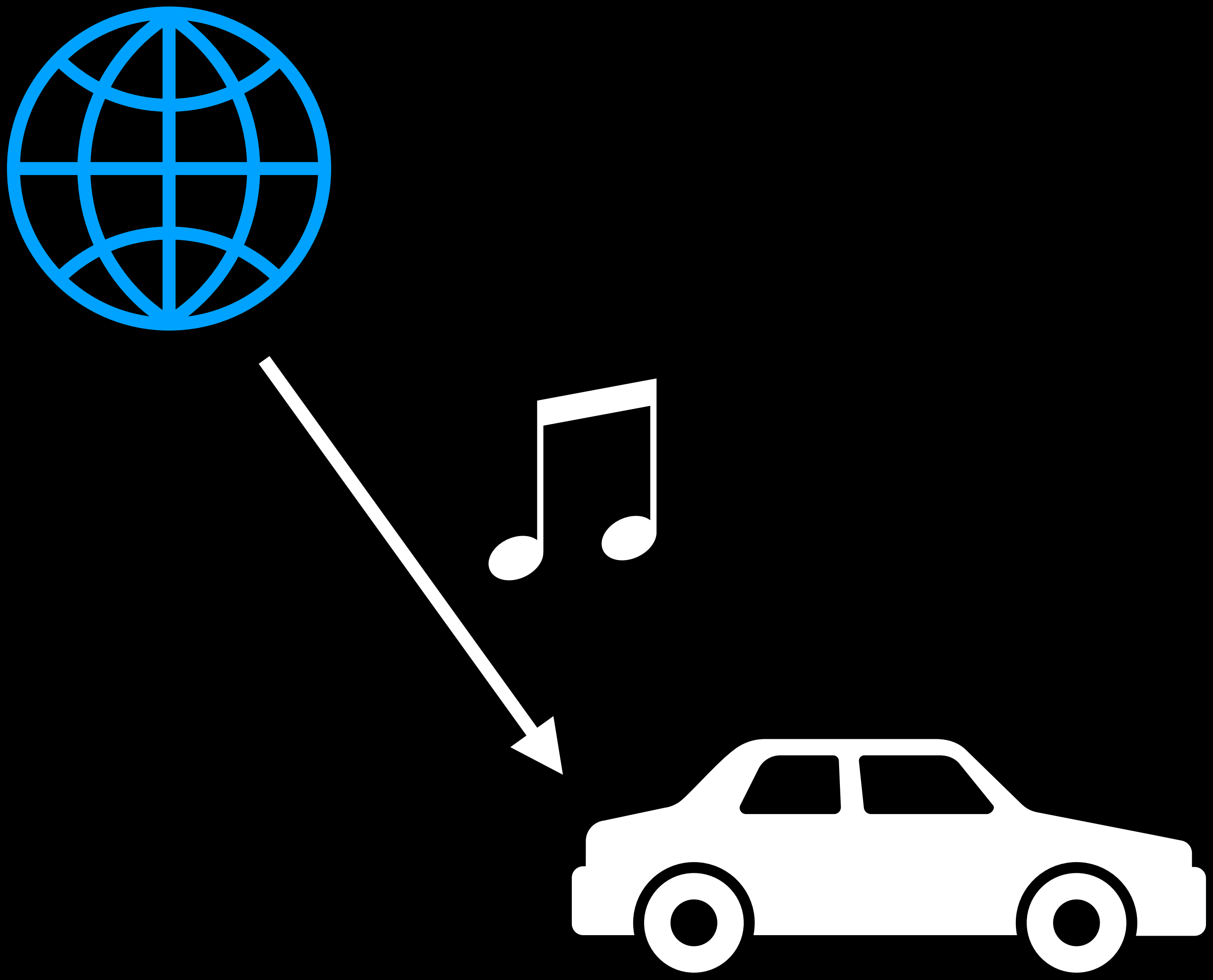
- Limitation of SMACK based application whitelist
    - Subject has full access to object with same label
    - Could not change by setting rule



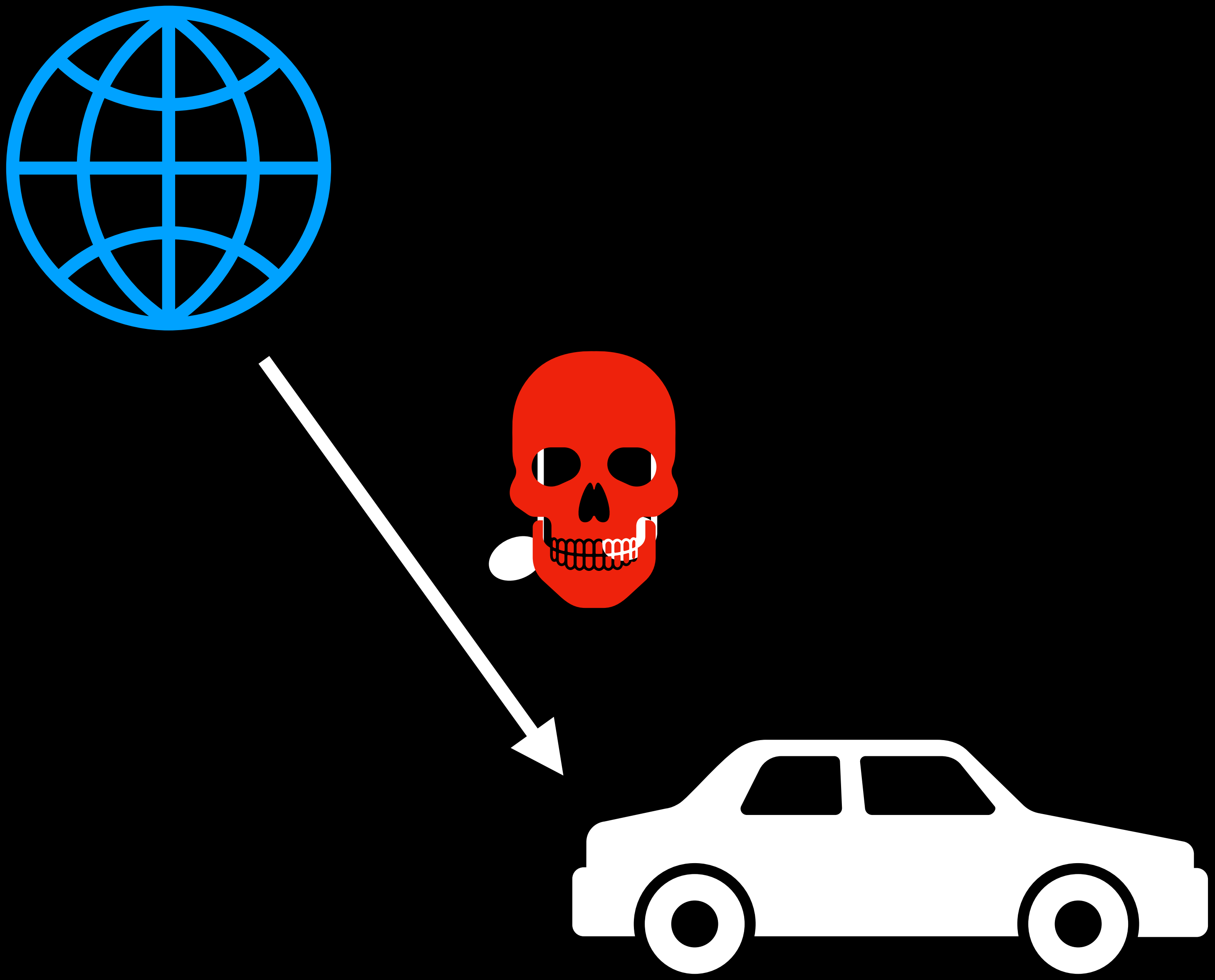
http://schaufler-ca.com/yahoo_site_admin/assets/docs/SmackWhitePaper.257153003.pdf

WHAT MAY HAPPENED IN AGL?

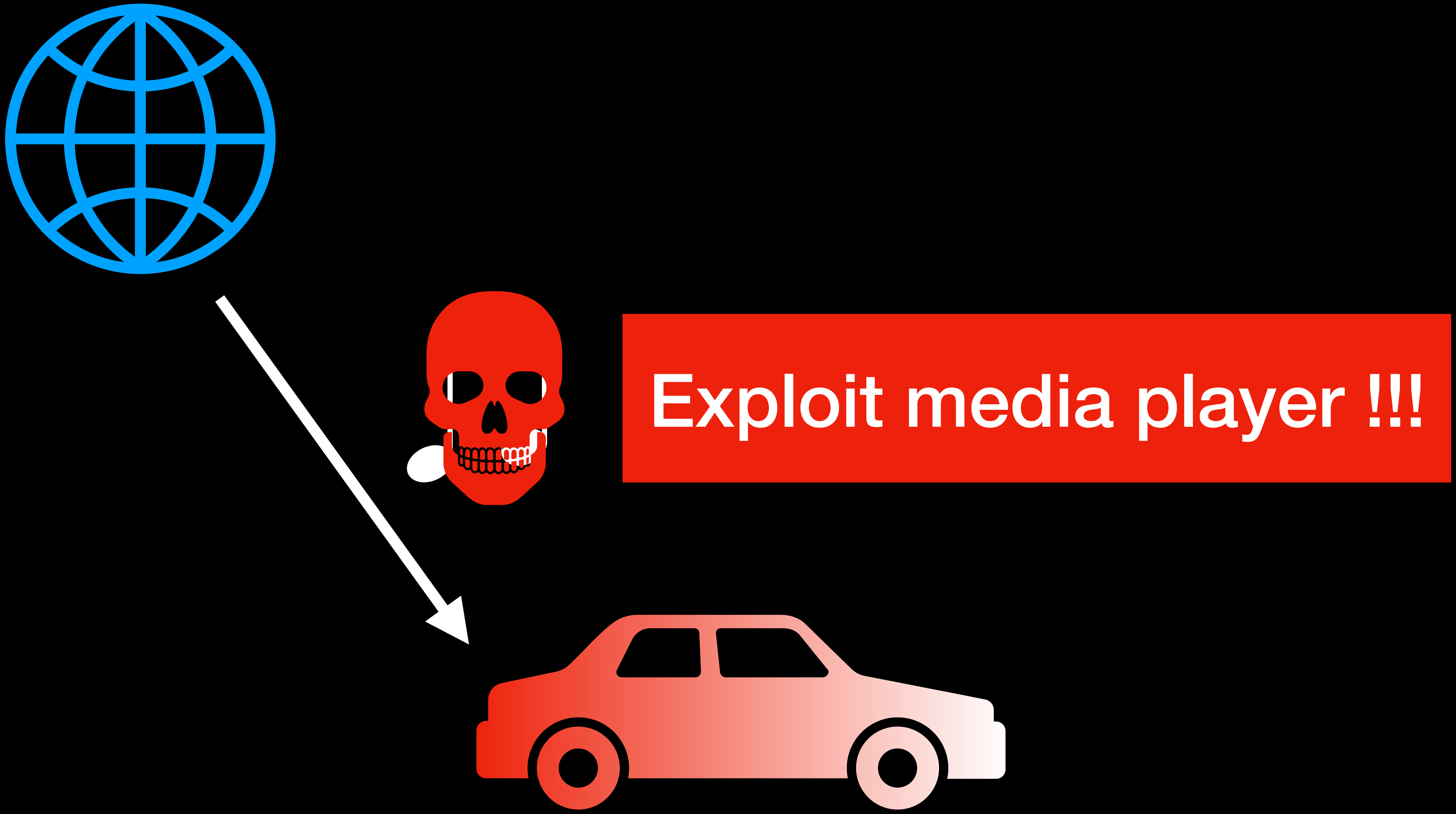
WHAT MAY HAPPENED IN AGL?



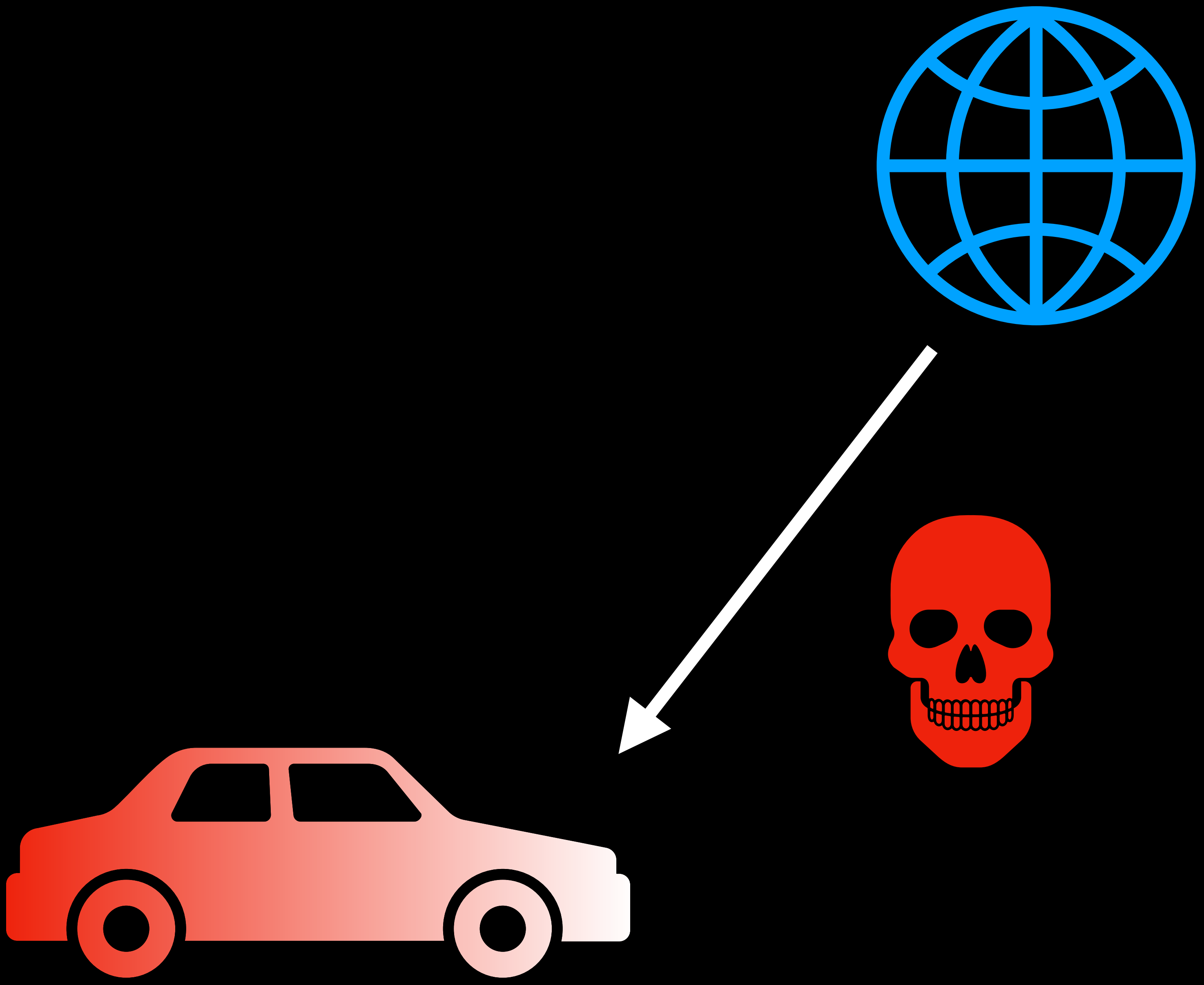
WHAT MAY HAPPENED IN AGL?



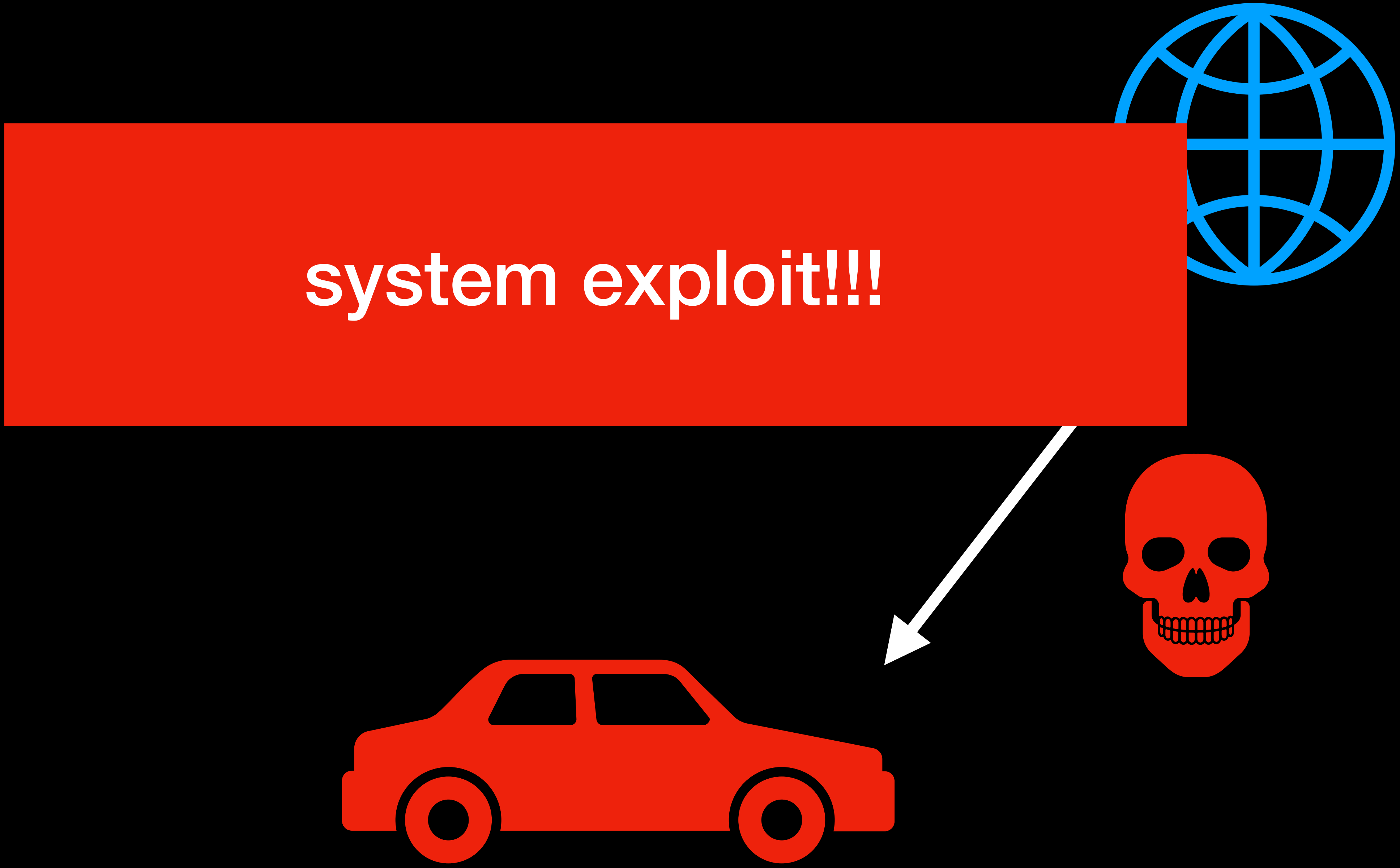
WHAT MAY HAPPENED IN AGL?



WHAT MAY HAPPENED IN AGL?



WHAT MAY HAPPENED IN AGL?



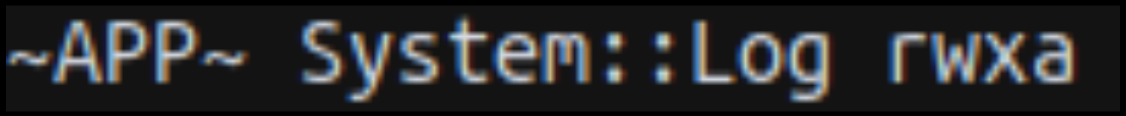
SMACK BASED WHITELIST ON AGL

- Some proposes
  - Not using root permission to run application
    - SMACK rule not effective in root privilege
  - No offline attack
    - May change smack label
  - We suppose not to change code and architecture

- 4 features:
    - **Block binary execute**
    - Block library load
    - Block interpreter script
    - Block kernel module load

BLOCK BINARY EXECUTION

- First try
  - Reference to DEP
    - Could not execute when it could write
    - Could not write when it could execute
  - e.g.

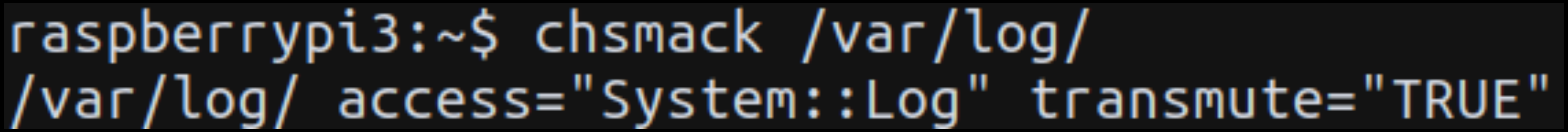


```
~APP~ System::Log rwxa
```

$\Rightarrow$

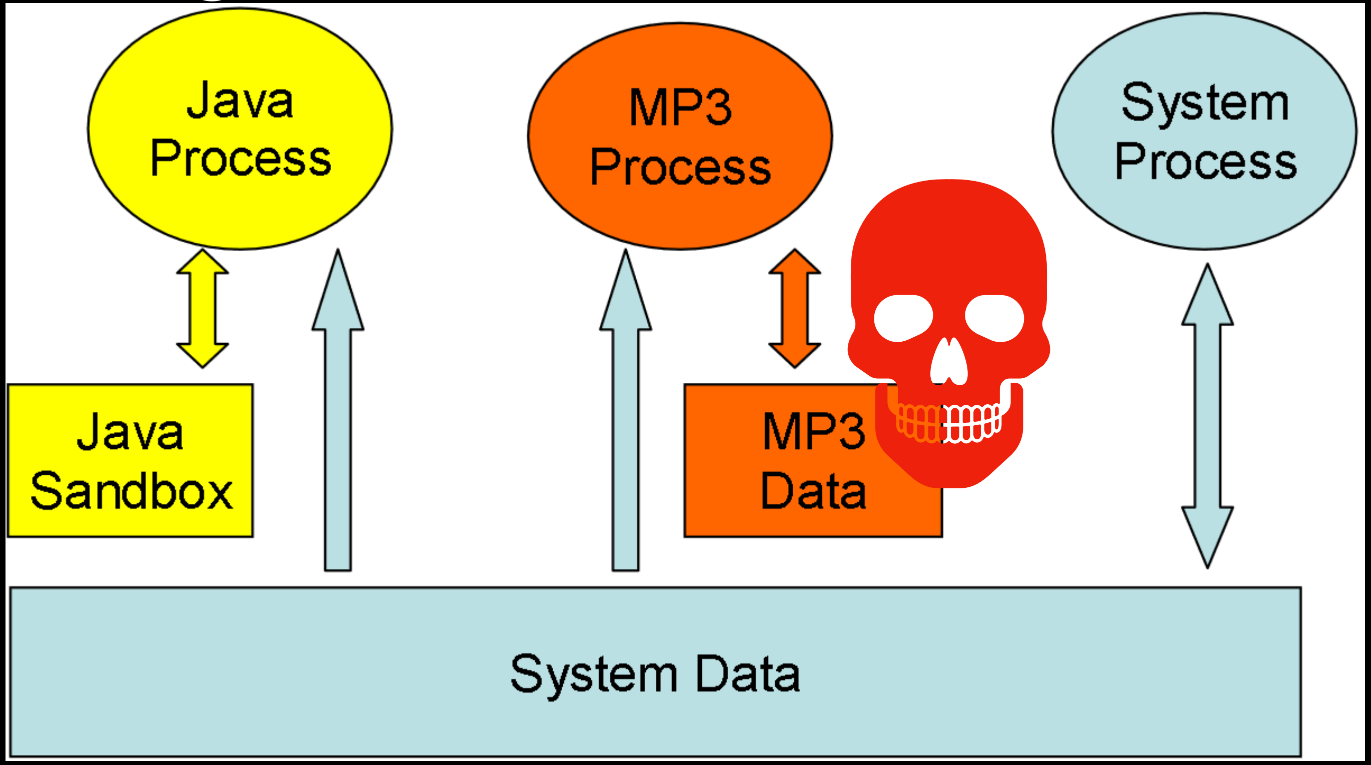
```
~APP~ System::Log rw-a
```

- Problems about our try
  - x permission on directory means permission for access file
  - If we unset x permission, then process cannot access file in the directory.



```
raspberrypi3:~$ chsmack /var/log/
/var/log/ access="System::Log" transmute="TRUE"
```

- Problems about our try
  - Subject have full access to object with same label
  - Could not change by setting rule



http://schaufler-ca.com/yahoo_site_admin/assets/docs/SmackWhitePaper.257153003.pdf

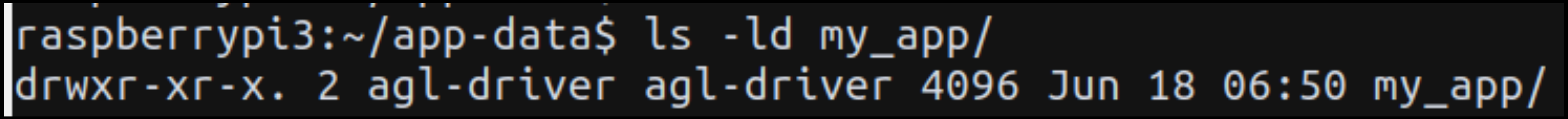
- How to solve?
    - Extend SMACK default rule
    - Let SMACK check rules when process access same label object

- How to solve?
    - ~~Extend SMACK default rule~~
    - ~~Let SMACK check rules when process access same label object~~

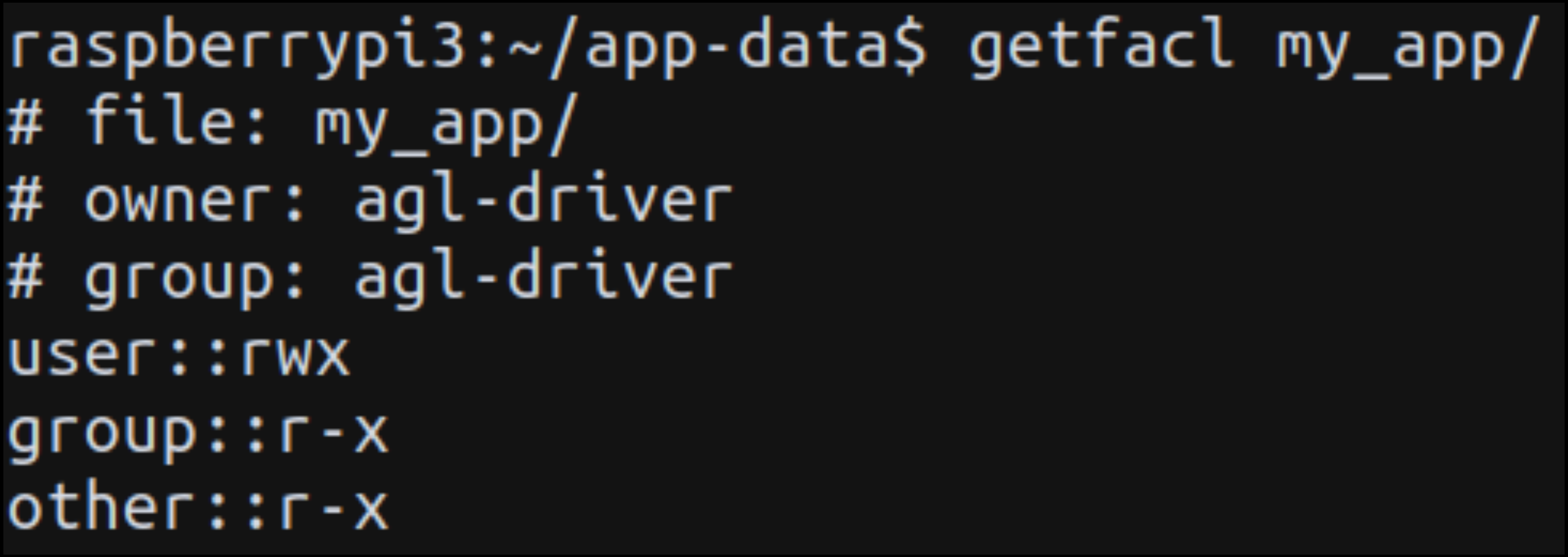
We suppose not to change code !!!!

- What about adding other feature?
  - Use access control list (ACL) to limit permissions
  - A list of permissions attached to an object.
  - Set default DAC permission to directory
  - When create new file, file will apply default permission

- Without ACL

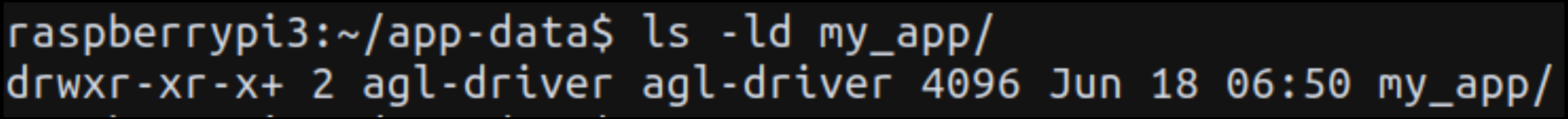


```
raspberrypi3:~/app-data$ ls -ld my_app/
drwxr-xr-x. 2 agl-driver agl-driver 4096 Jun 18 06:50 my_app/
```

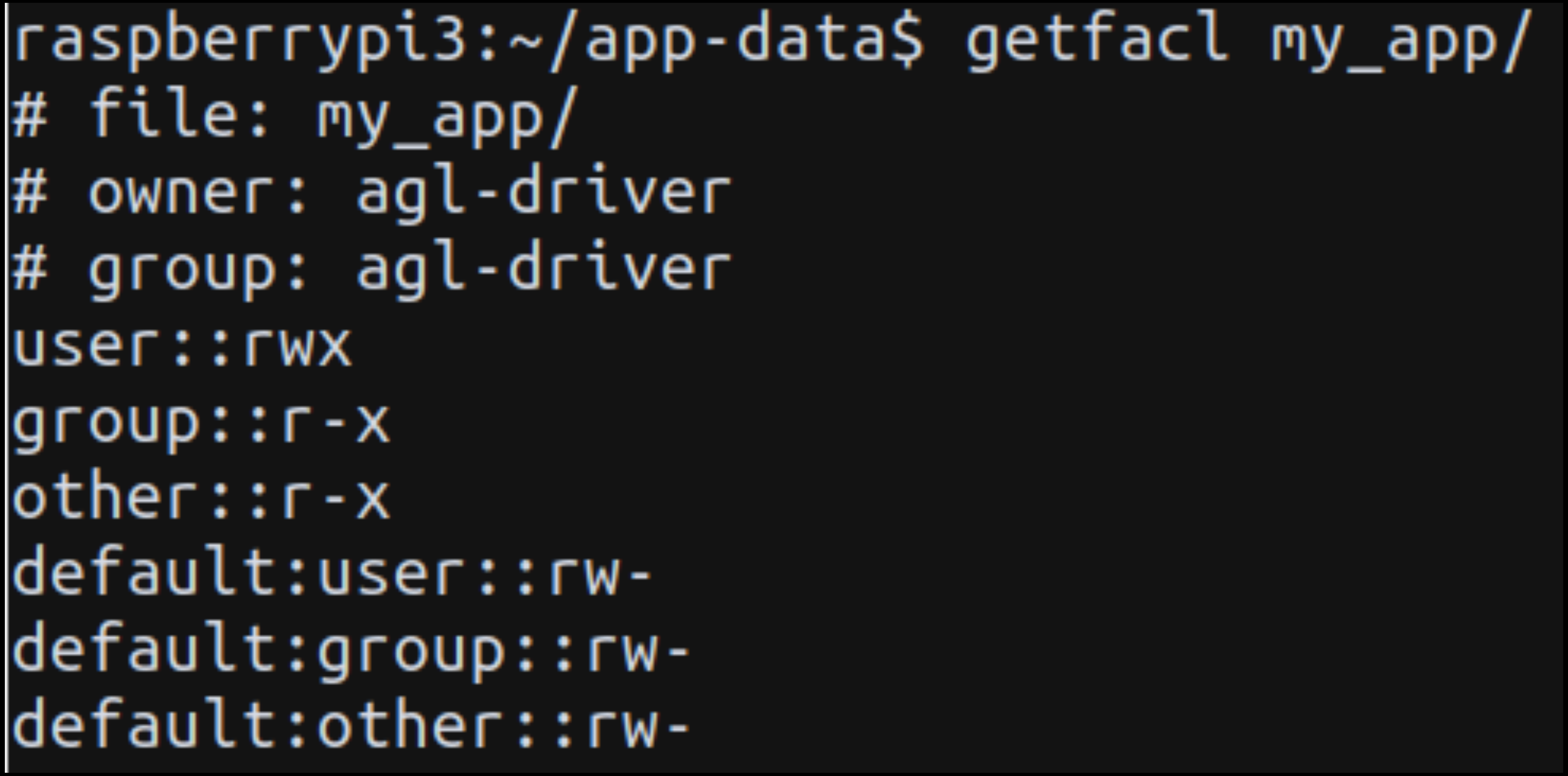


```
raspberrypi3:~/app-data$ getfacl my_app/
# file: my_app/
# owner: agl-driver
# group: agl-driver
user::rwx
group::r-x
other::r-x
```

- With ACL

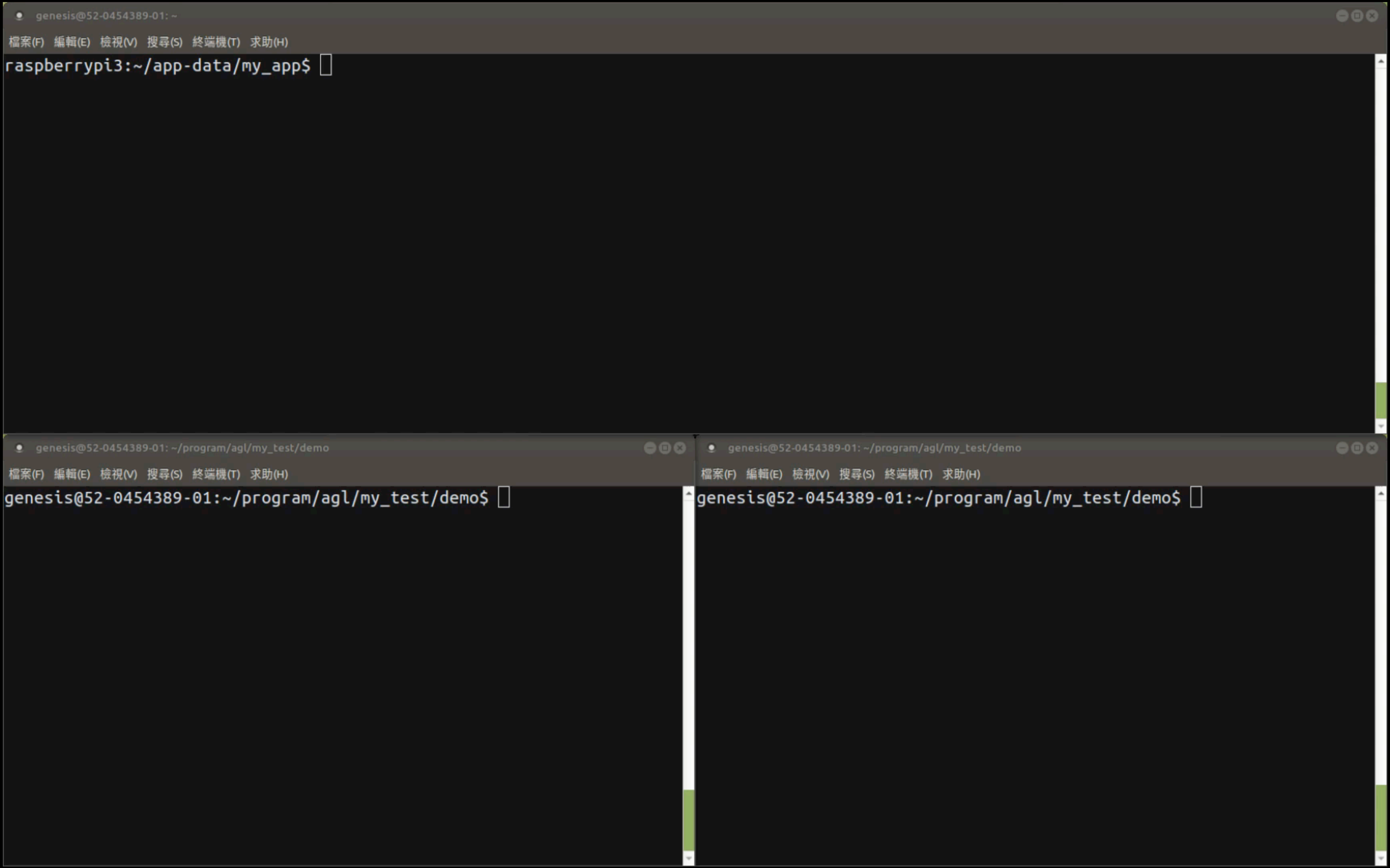


```
raspberrypi3:~/app-data$ ls -ld my_app/
drwxr-xr-x+ 2 agl-driver agl-driver 4096 Jun 18 06:50 my_app/
```



```
raspberrypi3:~/app-data$ getfacl my_app/
# file: my_app/
# owner: agl-driver
# group: agl-driver
user::rwx
group::r-x
other::r-x
default:user::rw-
default:group::rw-
default:other::rw-
```

- Change permission ?
  - Let app not allow to run relative command
  - Mark chmod, setfacl a new label
    - e.g. System::Privileged


genesis@52-0454389-01: ~
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

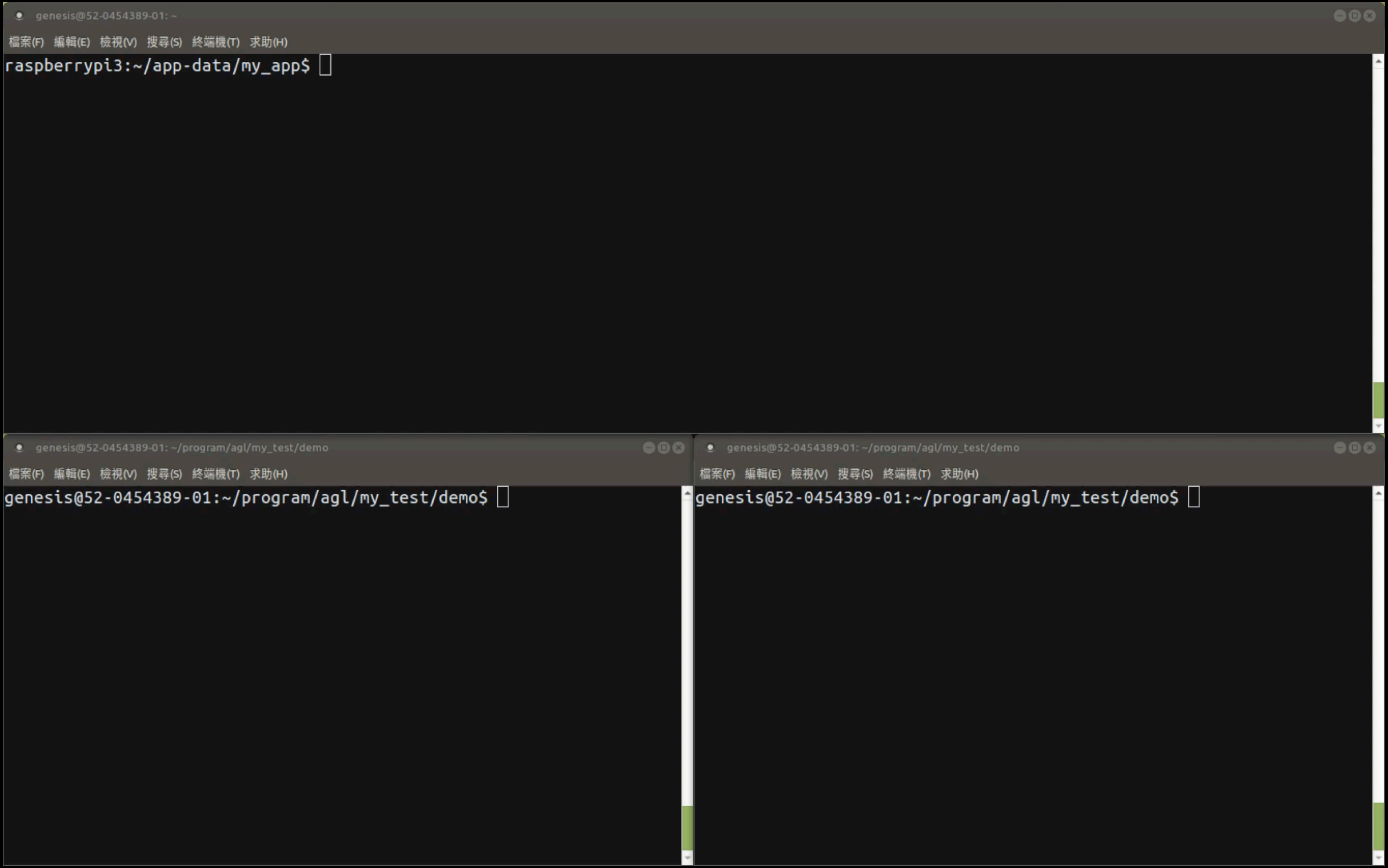
```
raspberrypi3:~/app-data/my_app$ 
```

genesis@52-0454389-01: ~/program/agl/my_test/demo
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
genesis@52-0454389-01:~/program/agl/my_test/demo$ 
```

genesis@52-0454389-01: ~/program/agl/my_test/demo
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
genesis@52-0454389-01:~/program/agl/my_test/demo$ 
```


genesis@52-0454389-01: ~
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
raspberrypi3:~/app-data/my_app$
```

genesis@52-0454389-01: ~/program/agl/my_test/demo
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
genesis@52-0454389-01:~/program/agl/my_test/demo$
```

genesis@52-0454389-01: ~/program/agl/my_test/demo
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
genesis@52-0454389-01:~/program/agl/my_test/demo$
```

- 4 features:
  - Block binary execute ✓
  - **Block library load**
  - Block interpreter script
  - Block kernel module load

BLOCK LIBRARY LOAD

- Use SMACK64MMAP extended attributes
  - Cannot inherit label
- Use ACL to set default permission without read permission
  - False negative

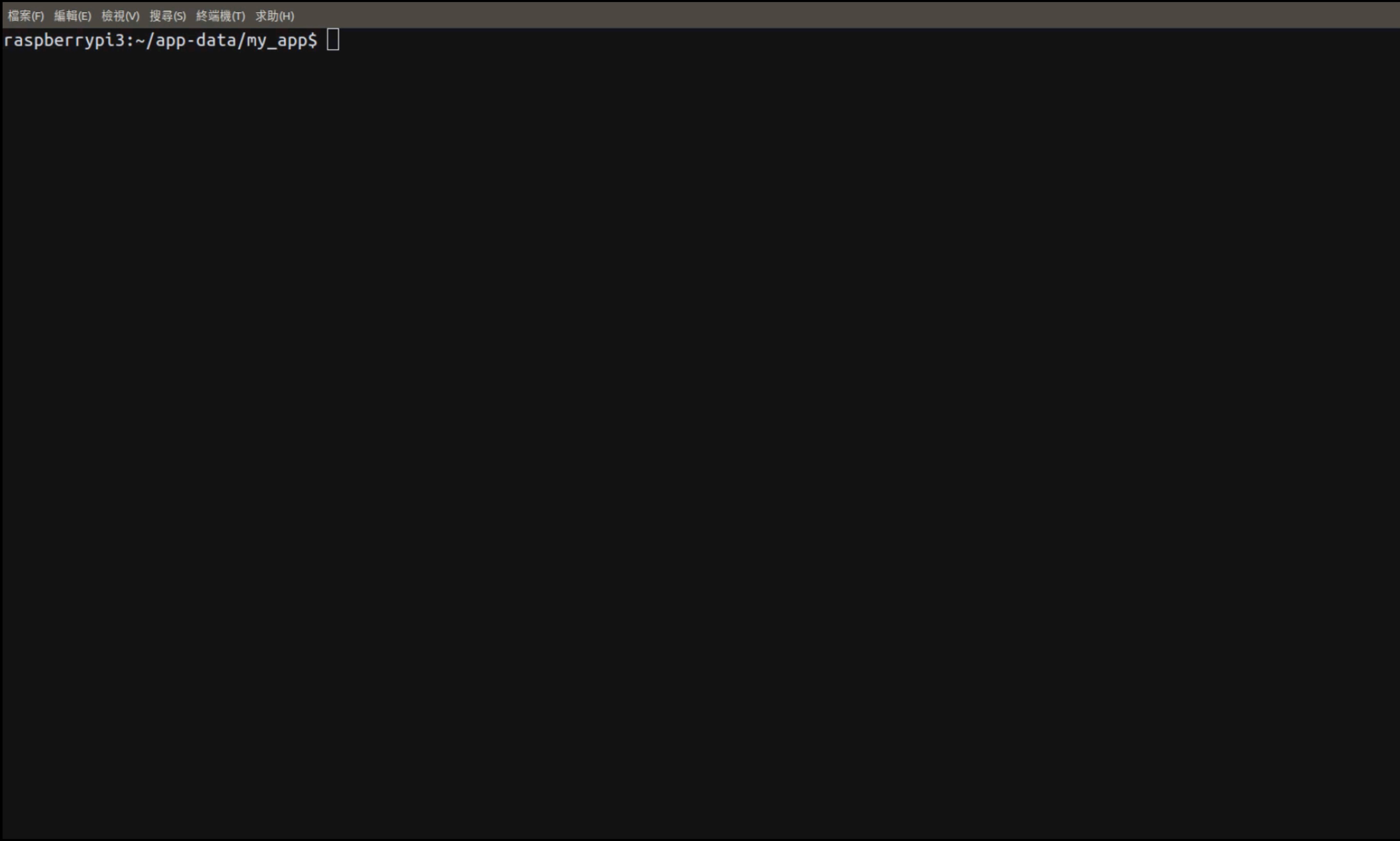
- 4 features:
  - Block binary execute ✓
  - Block library load ✗
  - **Block interpreter script**
  - Block kernel module load

BLOCK INTERPRETER SCRIPTS

- Take python for example
- Set unique label for python and .py, .pyc file
  - e.g. System::Python
- Only allow python to read script with same label

BLOCK INTERPRETER SCRIPTS

- Cases could not be blocked
  - cat test.py | python
  - python -c "print 'hello'"



```
raspberrypi3:~/app-data/my_app$ 
```

- 4 features:
  - Block binary execute ✓
  - Block library load ✗
  - Block interpreter script ✓
  - **Block kernel module load**

BLOCK LOAD KERNEL MODULE

- Currently cannot handle
- Load kernel module need root privilege
- SMACK will ignore rule when in root privilege
- We assume not using root permission

CONCLUSION

- Application whitelist prototype on AGL
- 4 features:
  - Block binary execute ✓
  - Block library load ✗
  - Block interpreter script ✓
  - Block kernel module load ✗

Q&A