

# Linux Mandatory Access Control

0866007 胡孝德

## AppArmor Installation

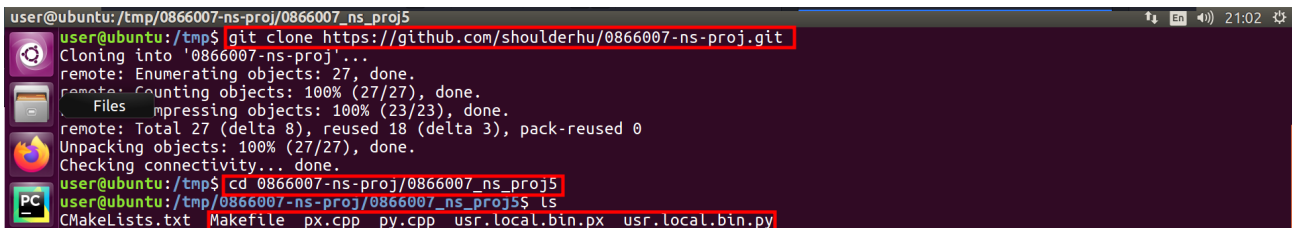
本 project 的實驗環境為 **Ubuntu 16.04**，其本身就預先安裝有 AppArmor，我們只需要安裝一些 command line tools 來幫助我們操作 AppArmor。

```
sudo apt install apparmor-utils
```

## Environment Setup

在開始實驗之前，先將 PoC (Proof of Concept) 程式碼從 GitHub 上 clone 到本地端，有測試程式的 program X, program Y, 還有兩者對應的 Apparmor profile, 用來做 Mandatory Access Control。

```
1 git clone https://github.com/shoulderhu/0866007-ns-proj.git
2 cd 0866007-ns-proj/0866007_ns_proj5
```

A terminal window screenshot showing the execution of commands to clone a repository and navigate to a specific directory. The terminal output includes the command 'git clone https://github.com/shoulderhu/0866007-ns-proj.git' and its output, followed by 'cd 0866007-ns-proj/0866007\_ns\_proj5' and its output. The prompt shows the current directory as '/tmp/0866007-ns-proj/0866007\_ns\_proj5'.

```
user@ubuntu:/tmp/0866007-ns-proj/0866007_ns_proj5
user@ubuntu:/tmp$ git clone https://github.com/shoulderhu/0866007-ns-proj.git
Cloning into '0866007-ns-proj'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 27 (delta 8), reused 18 (delta 3), pack-reused 0
Unpacking objects: 100% (27/27), done.
Checking connectivity... done.
user@ubuntu:/tmp$ cd 0866007-ns-proj/0866007_ns_proj5
user@ubuntu:/tmp/0866007-ns-proj/0866007_ns_proj5$ ls
CMakeLists.txt  Makefile  px.cpp  py.cpp  usr.local.bin.px  usr.local.bin.py
```

建立 `/var/X/` 與 `/var/Y/` 資料夾，然後在兩資料夾內建立測試文件 `test.txt`。還有，為了要讓非 root 使用者能寫入以符合題目需求，用 `chmod` 為兩資料夾加上 `w` 權限。

```

1  sudo mkdir /var/X/ /var/Y/
2  sudo touch /var/X/test.txt /var/Y/test.txt
3
4  sudo chmod -R a+w /var/X/
5  sudo chmod -R a+w /var/Y/

```

新增使用者 `userx`，將其 login shell 設定為 `/bin/sh` (後面實驗會解釋)，密碼為 `userx`

。

```

1  sudo useradd -s /bin/sh userx
2  sudo passwd userx

```

```

user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo useradd -s /bin/sh userx
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo passwd userx
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ tail -1 /etc/passwd
userx:x:1001:1001::/home/userx:/bin/sh

```

使用 `make` 指令編譯 Program X `px`，然後放到 `/usr/local/bin/` 目錄底下。在那之前，因為 `px` 需要用到 `libcurl` library 來從網路上下載 source code，所以需要先安裝該 library。

```

1  # Download and Install libcurl library
2  wget https://curl.haxx.se/download/curl-7.70.0.tar.gz
3  tar xzf curl-7.70.0.tar.gz
4  cd curl-7.70.0/
5  ./configure
6  make -j4
7  sudo make install
8
9  # Compile ProgramX
10 make
11 sudo cp px /usr/local/bin/

```

```

Terminal File Edit View Search Terminal Help
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ make
g++ -c px.cpp
g++ -o px px.o -lcurl
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo cp px /usr/local/bin/
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ ls -l /usr/local/bin/px
-rwxr-xr-x 1 root root 8552 六 7 21:27 /usr/local/bin/px

```

# Experiment

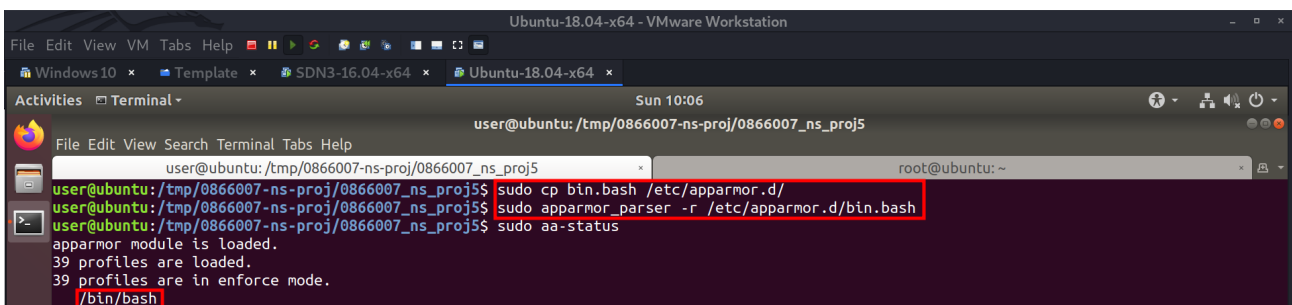
## Only UserX is allowed to execute Program X

我認為這條 policy 是最難達成的，因為 Apparmor 的設計主要是針對執行檔，限制該執行檔能夠存取的檔案或是網路的存取之類的。如果要限制使用者，使用 Discretionary Access Control 比較容易達成。

然後我想到，在使用者登入後會拿到 login shell，如果對 login shell 做 MAC，就能達成此 policy。Ubuntu 的 default login shell 為 `/bin/bash`，於是我只需要提供一份 bash 的 AppArmor profile 即可，並讓 userx 使用其他的 login shell，如 `/bin/sh` 等等就不會受到此限制。

以下動作將 `/bin/bash` 的 AppArmor profile 給載入

```
1 sudo cp bin.bash /etc/apparmor.d/  
2 sudo apparmor_parser -r /etc/apparmor.d/bin.bash
```



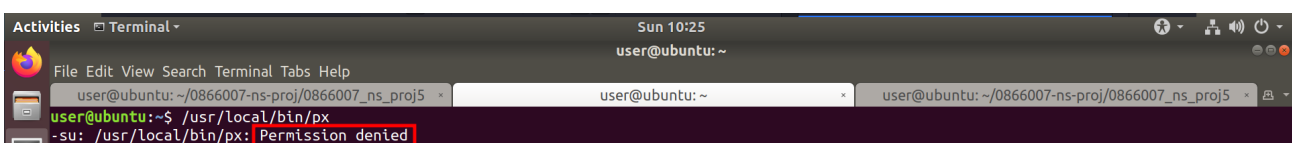
The screenshot shows a terminal window titled 'Ubuntu-18.04-x64 - VMware Workstation'. The user is in a directory `/tmp/0866007-ns-proj/0866007_ns_proj5`. They execute the following commands:

```
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo cp bin.bash /etc/apparmor.d/  
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo apparmor_parser -r /etc/apparmor.d/bin.bash  
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo aa-status
```

The output of `aa-status` is:

```
apparmor module is loaded.  
39 profiles are loaded.  
39 profiles are in enforce mode.  
/bin/bash
```

拿使用者 `user` 與 `userx` 來做比較，便能得知 policy 是否有正確被執行，從圖中可以看到 `user` 的畫面出現 `Permission denied`。



The screenshot shows a terminal window with the same title. The user is in a directory `~/0866007-ns-proj/0866007_ns_proj5`. They execute the following commands:

```
user@ubuntu: ~$ /usr/local/bin/px  
-su: /usr/local/bin/px: Permission denied
```

```
Activities Terminal Sun 10:27 user@ubuntu: ~/0866007-ns-proj/0866007_ns_proj5
File Edit View Search Terminal Tabs Help
user@ubuntu: ~/0866007-ns-proj/0866007_ns_proj5 user@ubuntu: ~ user@ubuntu: ~/0866007-ns-proj/0866007_ns_proj5
$ id
uid=1004(userx) gid=1004(userx) groups=1004(userx)
$ /usr/local/bin/px
$
```

**Program X will retrieve the source code of Program Y from the Internet and build Program Y under /var/X/ & Program X will install Program Y under /var/Y/ & Program X will fork a child process to execute Program Y**

測試用 Program X 的原始碼如下，利用 **libcurl** 到 GitHub 上將 program Y 的原始碼給下載下來，利用 **g++** 對 program Y 的原始碼做編譯並放到 /var/Y，最後透過 **fork()**，來建立 child process 並使用 **execlp** 來執行 program Y。

```
1  #include <stdio>      // perror()
2  #include <stdlib>      // EXIT_FAILURE
3  #include <curl/curl.h>
4  #include <unistd.h>    // fork()
5  #include <fcntl.h>     // mkdir()
6  #include <sys/stat.h>
7  #include <sys/wait.h> // wait()
8
9  int main(int argc, char *argv[]) {
10     FILE *fp;
11     CURL *curl;
12     CURLcode res;
13
14     if ((fp = fopen("/var/X/py.cpp", "w")) == nullptr) {
15         perror("open()");
16         exit(EXIT_FAILURE);
17     }
18
19     if ((curl = curl_easy_init())) {
20         curl_easy_setopt(curl, CURLOPT_URL, "https://raw.githubusercontent.com/0866007-ns-proj/0866007_ns_proj5/main/py.cpp");
21         curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp);
22
23         res = curl_easy_perform(curl);
24         curl_easy_cleanup(curl);
25         fclose(fp);
26
27         if (res != CURLE_OK) {
28             fprintf(stderr, "curl_easy_perform(): %s\n", curl_easy_strerror(res));
29             exit(EXIT_FAILURE);
30         }
31     }
```

```

31
32     } else {
33         perror("curl");
34         exit(EXIT_FAILURE);
35     }
36
37     system("/usr/bin/g++ -o /var/Y/py /var/X/py.cpp");
38
39     switch (fork()) {
40         case -1:
41             perror("fork()");
42             exit(EXIT_FAILURE);
43         case 0: // child
44             if (execl("/var/Y/py", "/var/Y/py", nullptr) == -1) {
45                 perror("execlp()");
46                 exit(EXIT_FAILURE);
47             }
48             break;
49         default: // parent
50             wait(nullptr);
51             break;
52     }
53
54     return 0;
55 }

```

Program X 的 Apparmor profile 如下，`include` 語法用來置入一些常用的 profile rules，如 **base** 中含有 c shared library 的 profile，**nameservice** 中含有 DNS 的 profile，**user-tmp** 中含有 tmp 資料夾的 profile。第 8 行允許 tcp 的網路連線。第 10 - 14 行含有編譯 Program Y 所需用到的檔案。第 16 - 18 行為題目要求之路徑。

```

1  #include <tunables/global>
2
3  /usr/local/bin/px {
4      #include <abstractions/base>
5      #include <abstractions/nameservice>
6      #include <abstractions/user-tmp>
7
8      network tcp,
9
10     /bin/dash mrix,
11     /usr/bin/g++-5 mrix,
12     /usr/bin/x86_64-linux-gnu-* mrix,
13     /usr/include/** r,
14     /usr/lib/gcc/x86_64-linux-gnu/5/* mrix,

```

```
15
16     /var/X/* rw,
17     /var/Y/* rw,
18     /var/Y/py px,
19 }
```

**Program Y is only allowed to read/write files under /var/Y/ & Program Y is not allowed to create or accept network connections.**

測試用 Program Y 的原始碼如下，本程式碼做 3 件事，第一件是開啟 位於 /var/X 下的 test.txt 檔案，第二件是開啟位於 /var/Y/ 下的 test.txt 檔案，第三件是建立 socket。根據題目要求，第一件與第三件應該會被 Apparmor 的 profile 給阻擋下來。

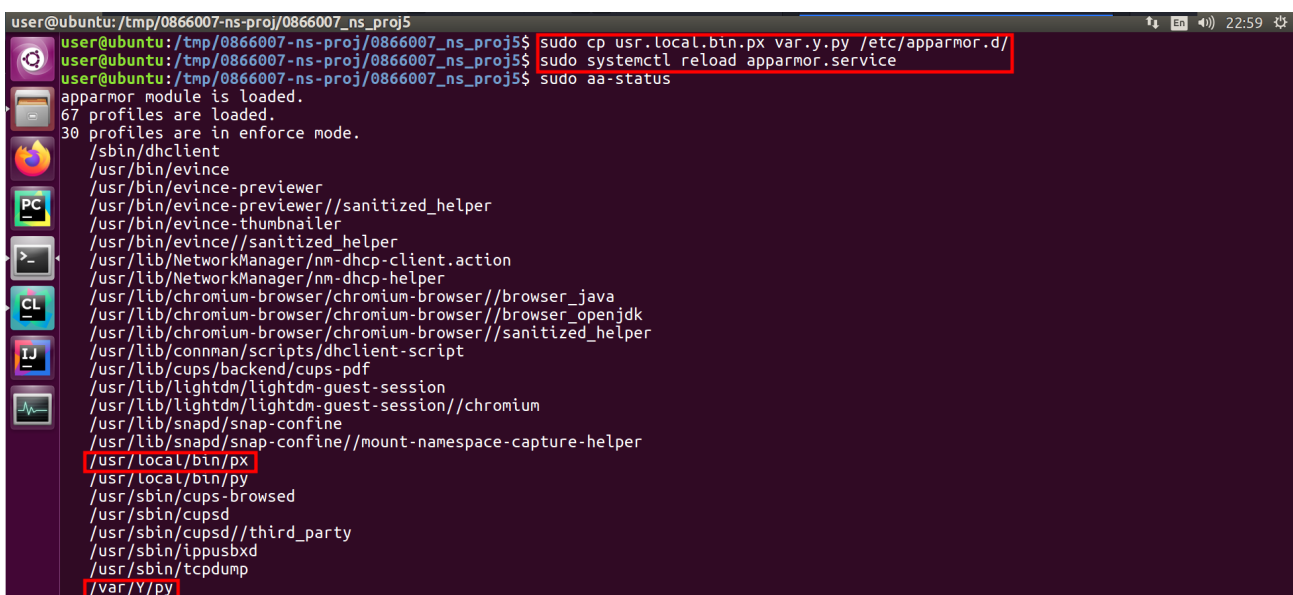
```
1  #include <stdio>
2  #include <unistd.h>
3  #include <fcntl.h>      // open()
4  #include <sys/socket.h> // socket()
5
6  int main() {
7      int fd;
8
9      if ((fd = open("/var/X/test.txt", O_RDWR)) == -1) {
10         perror("/var/X/test.txt");
11     } else {
12         close(fd);
13     }
14
15     if ((fd = open("/var/Y/test.txt", O_RDWR)) == -1) {
16         perror("/var/Y/test.txt");
17     } else {
18         close(fd);
19     }
20
21     if ((fd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
22         perror("socket()");
23     } else {
24         close(fd);
25     }
26
27     return 0;
28 }
```

Program Y 的 AppArmor profile 如下，第 6 行的 deny network 其實可以不需要寫，因為 Apparmor 採用白名單的策略，沒有明確允許的就會被拒絕，在這裡寫上此行只是為了明確起見。

```
1 #include <tunables/global>
2
3 /var/Y/py {
4     #include <abstractions/base>
5
6     deny network,
7
8     /var/Y/*      mrw,
9 }
```

以下指令將 Program X 與 Program Y 的 profile 給載入到 AppArmor 當中，然後執行 Program X，正常的情況下，程式應該會顯示兩行錯誤訊息，一行是開檔的錯誤，另一行是建立 socket 失敗的錯誤。

```
1 sudo cp usr.local.bin.px var.y.py /etc/apparmor.d/
2 sudo systemctl reload apparmor.service
```



```
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo cp usr.local.bin.px var.y.py /etc/apparmor.d/
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo systemctl reload apparmor.service
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_proj5$ sudo aa-status
apparmor module is loaded.
67 profiles are loaded.
30 profiles are in enforce mode.
/sbin/dhclient
/usr/bin/evince
/usr/bin/evince-previewer
/usr/bin/evince-previewer//sanitized_helper
/usr/bin/evince-thumbnailer
/usr/bin/evince//sanitized_helper
/usr/lib/NetworkManager/nm-dhcp-client.action
/usr/lib/NetworkManager/nm-dhcp-helper
/usr/lib/chromium-browser/chromium-browser//browser_java
/usr/lib/chromium-browser/chromium-browser//browser_openjdk
/usr/lib/chromium-browser/chromium-browser//sanitized_helper
/usr/lib/connman/scripts/dhclient-script
/usr/lib/cups/backend/cups-pdf
/usr/lib/lightdm/lightdm-guest-session
/usr/lib/lightdm/lightdm-guest-session//chromium
/usr/lib/snapd/snap-confine
/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/local/bin/px
/usr/local/bin/py
/usr/sbin/cups-browsed
/usr/sbin/cupsd
/usr/sbin/cupsd//third_party
/usr/sbin/ippusbxd
/usr/sbin/tcpdump
/var/Y/py
```

```
Terminal File Edit View Search Terminal Tabs Help
user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_...  user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_...  user@ubuntu: /tmp/0866007-ns-proj/0866007_ns_...
$ id
uid=1001(userx) gid=1001(userx) groups=1001(userx)
$ /usr/local/bin/px
/var/X/test.txt: Permission denied
socket(): Permission denied
```

## Reference



**AppArmor - Community Help Wiki**

<https://help.ubuntu.com/community/AppArmor>



**fervid/AppArmor-Profile-Examples**

<https://github.com/fervid/AppArmor-Profile-Examples>