

# Mini-Project #5

## Linux Mandatory Access Control

0866007 胡孝德

# **AppArmor Installation & Usage**

# Apparmor

- Install
  - `sudo apt install apparmor-utils`
- Usage
  - `aa-status`
  - `aa-disable / aa-enable`
    - `sudo aa-disable /usr/local/bin/px`
  - `aa-logprof`
    - `sudo aa-logprof -d /usr/local/bin/px`
  - `apparmor_parser`
    - `sudo apparmor_parser -r /etc/apparmor.d/usr.local.bin.px`
    - `sudo apparmor_parser -R /etc/apparmor.d/usr.local.bin.px`

# Environment Setup

# Setup

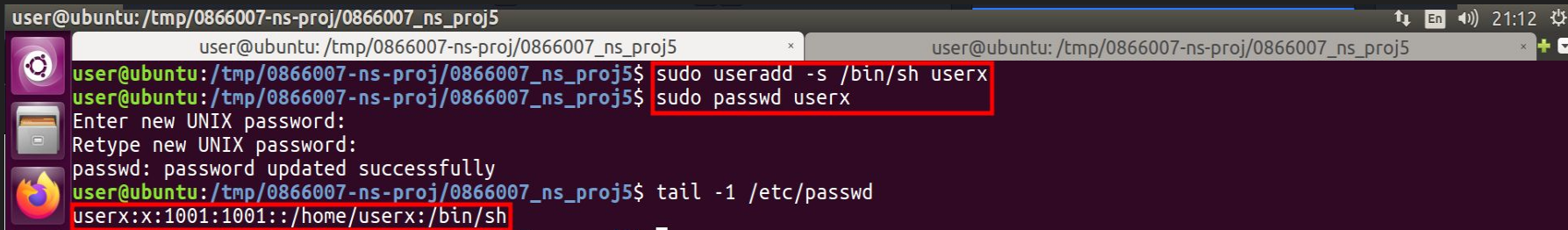
- PoC - AppArmor profiles

```
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
```

- /var/ directory
  - sudo mkdir /var/X/ /var/Y/
  - sudo touch /var/X/test.txt /var/Y/test.txt
  - sudo chmod -R a+w /var/X/
  - sudo chmod -R a+w /var/Y/

# Setup

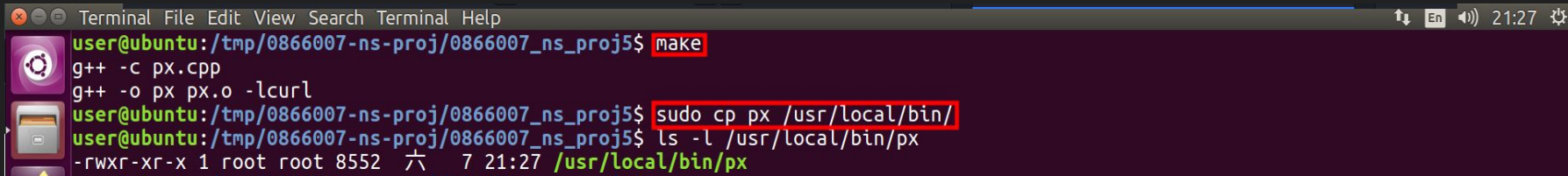
- User - userx

A terminal window with a dark purple background. The title bar shows 'user@ubuntu: /tmp/0866007-ns-proj/0866007\_ns\_proj5'. The terminal shows the following commands and output:

```
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
```

The last two lines of the command sequence are highlighted with a red box.

- Program X - compile

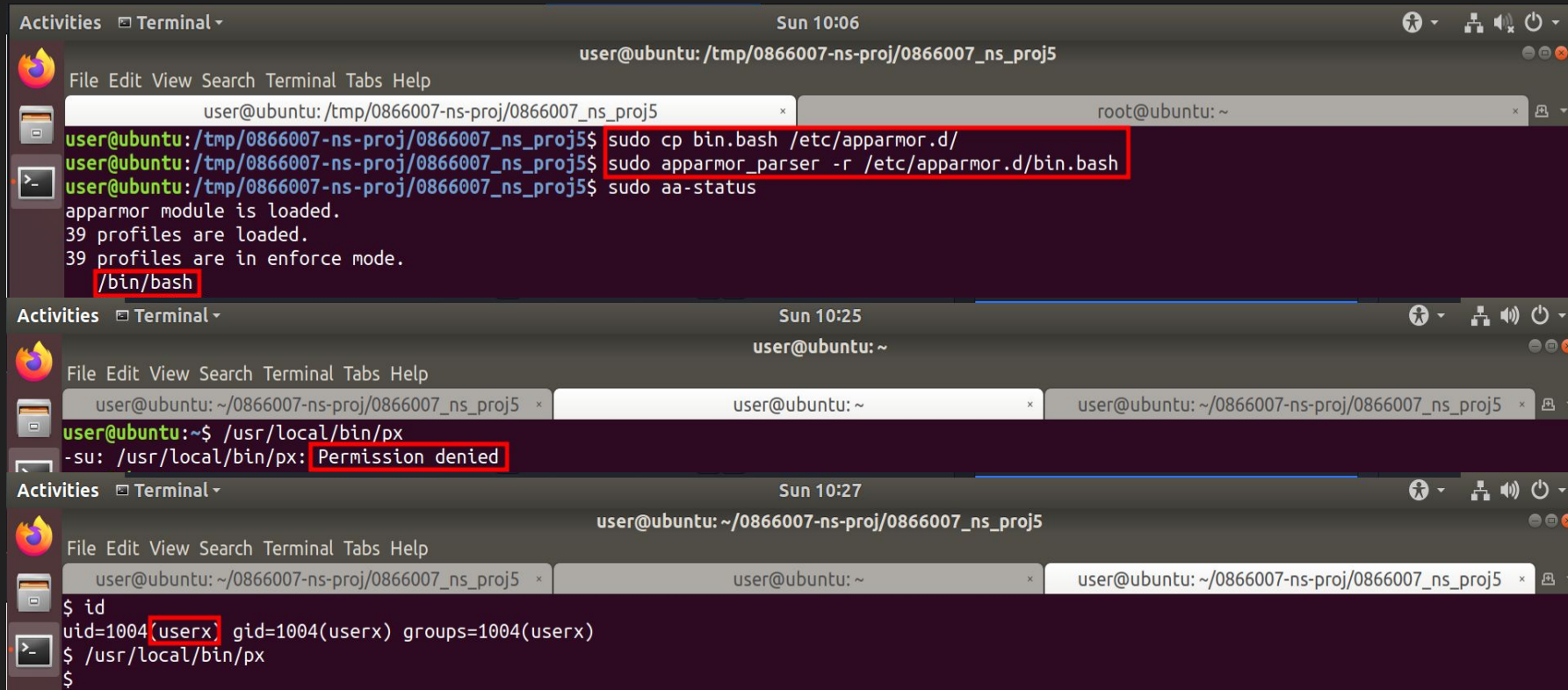
A terminal window with a dark purple background. The title bar shows 'Terminal File Edit View Search Terminal Help'. The terminal shows the following commands and output:

```
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
```

The first two lines of the command sequence are highlighted with a red box.

# Experiment

# Only UserX is allowed to execute Program X



The image displays three sequential terminal windows illustrating the setup and execution of a program within a namespace.

**Terminal 1 (Sun 10:06):** The user is in the namespace `/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` shows that the AppArmor module is loaded, 39 profiles are loaded, and 39 profiles are in enforce mode. The prompt `/bin/bash` is highlighted with a red box.

**Terminal 2 (Sun 10:25):** The user is now at the root prompt `user@ubuntu: ~`. They attempt to execute a program in the namespace:

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

The error message `Permission denied` is highlighted with a red box.

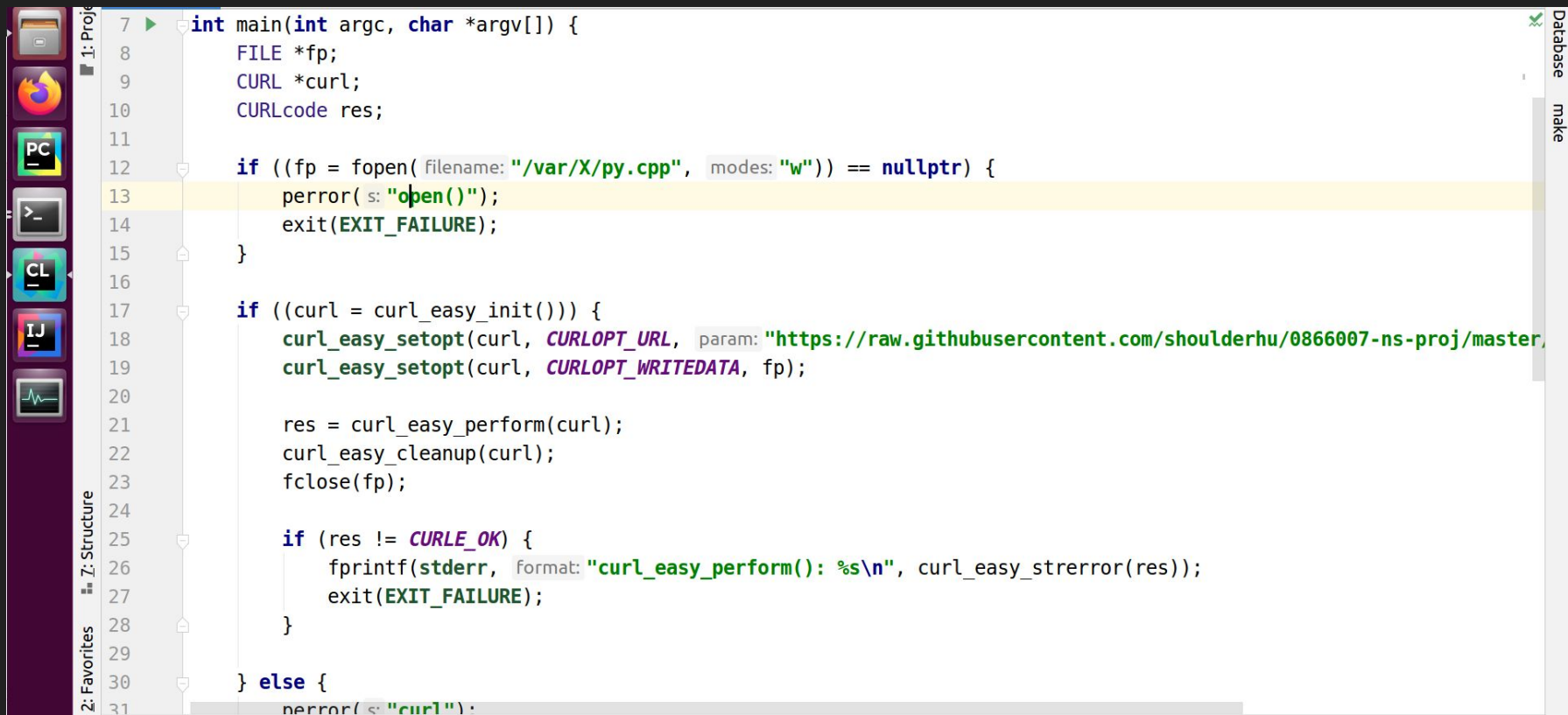
**Terminal 3 (Sun 10:27):** The user is back in the namespace `~/0866007-ns-proj/0866007_ns_proj5`. They run the `id` command to check their identity:

```
$ id  
uid=1004(userx) gid=1004(userx) groups=1004(userx)  
$ /usr/local/bin/px  
$
```

The username `userx` in the output of the `id` command is highlighted with a red box.



# Program X



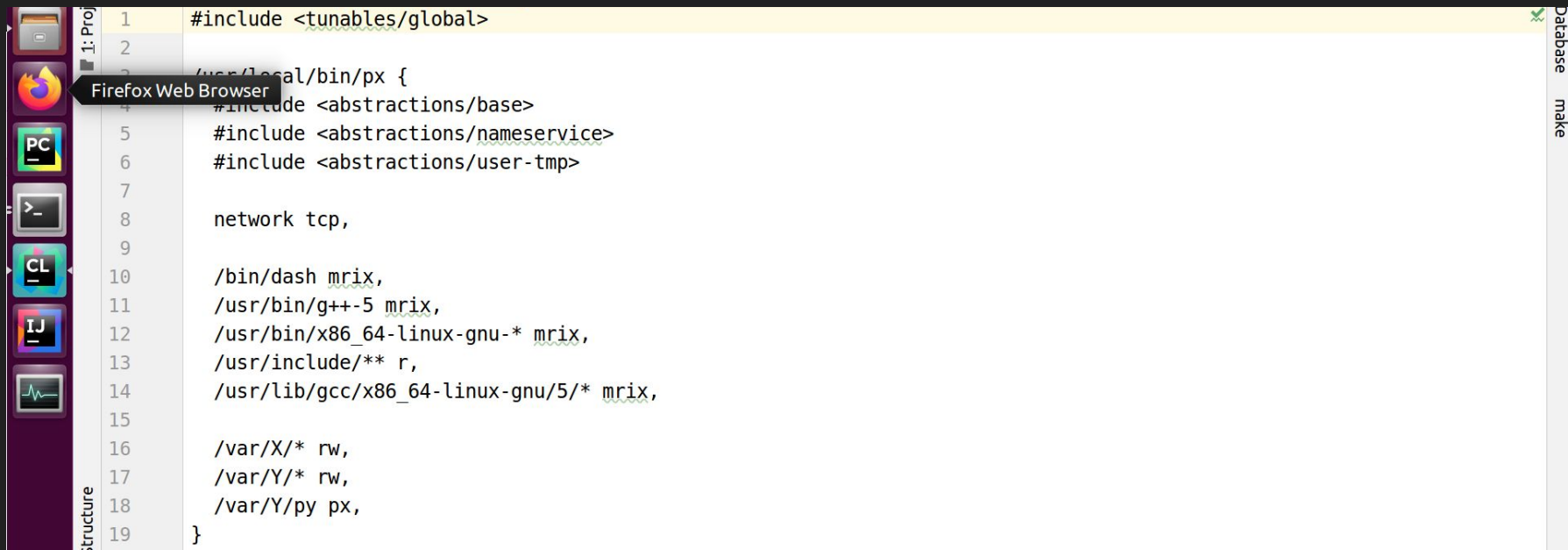
```
7 int main(int argc, char *argv[]) {
8     FILE *fp;
9     CURL *curl;
10    CURLcode res;
11
12    if ((fp = fopen( filename: "/var/X/py.cpp", modes: "w")) == nullptr) {
13        perror( s: "open()");
14        exit(EXIT_FAILURE);
15    }
16
17    if ((curl = curl_easy_init())) {
18        curl_easy_setopt(curl, CURLOPT_URL, param: "https://raw.githubusercontent.com/shoulderhu/0866007-ns-proj/master",
19        curl_easy_setopt(curl, CURLOPT_WRITEDATA, fp);
20
21        res = curl_easy_perform(curl);
22        curl_easy_cleanup(curl);
23        fclose(fp);
24
25        if (res != CURLE_OK) {
26            fprintf(stderr, format: "curl_easy_perform(): %s\n", curl_easy_strerror(res));
27            exit(EXIT_FAILURE);
28        }
29    } else {
30        perror( s: "curl");
31    }
```

# Program X



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

# Program X's Apparmor profile



```
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 }
```

Firefox Web Browser

Database make

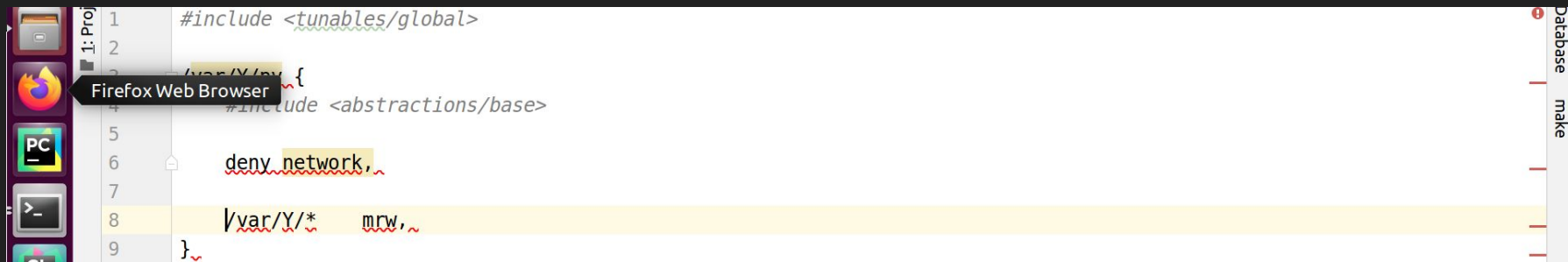
Structure

# Program Y



```
6  int main() {
7      int fd;
8
9      if ((fd = open( file: "/var/X/test.txt", 0_RDWR)) == -1) {
10         perror( s: "/var/X/test.txt");
11     } else {
12         close(fd);
13     }
14
15     if ((fd = open( file: "/var/Y/test.txt", 0_RDWR)) == -1) {
16         perror( s: "/var/Y/test.txt");
17     } else {
18         close(fd);
19     }
20
21     if ((fd = socket(AF_INET, type: SOCK_STREAM, protocol: 0)) == -1) {
22         perror( s: "socket()");
23     } else {
24         close(fd);
25     }
26
27     return 0;
28 }
```

# Program Y's Apparmor profile



The screenshot shows a code editor with a dark theme. On the left, there is a sidebar with icons for a file manager, Firefox Web Browser, a PC icon, a terminal, and a settings icon. The main editor area displays the following code:

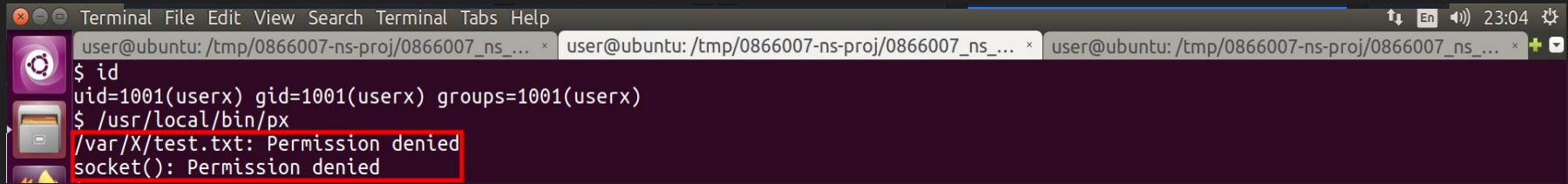
```
1  #include <tunables/global>
2
3  /var/Y/ {
4      #include <abstractions/base>
5
6      deny network,
7
8      /var/Y/* mrw,
9  }
```

The code is syntax-highlighted. The first line is a comment. The second line is empty. The third line starts a block for `/var/Y/`. The fourth line is a comment. The fifth line is empty. The sixth line is `deny network,`. The seventh line is empty. The eighth line is `/var/Y/* mrw,`. The ninth line ends the block with `}`. The code is displayed on a light background with a dark border. The sidebar on the left has a dark background with colorful icons. The Firefox icon is highlighted with a tooltip that says "Firefox Web Browser". The PC icon is a green square with a white "PC" text. The terminal icon is a black square with a white prompt character. The settings icon is a gear. The right sidebar has a dark background with a red "Database" label and a "make" button.

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

```
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
```

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

A terminal window with a dark purple background and white text. The window title bar shows 'Terminal File Edit View Search Terminal Tabs Help' and system icons on the right. Three tabs are open, all showing the same path: 'user@ubuntu: /tmp/0866007-ns-proj/0866007\_ns\_...'. The terminal content shows the execution of 'id' and a program 'px'. The output of 'id' is 'uid=1001(userx) gid=1001(userx) groups=1001(userx)'. The output of 'px' shows two 'Permission denied' messages: '/var/X/test.txt: Permission denied' and 'socket(): Permission denied'. These two lines are highlighted with a red rectangular box.

```
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
```

**The End**