(CSIE 4004) Homework 5

due date: 5/20 (0:00)

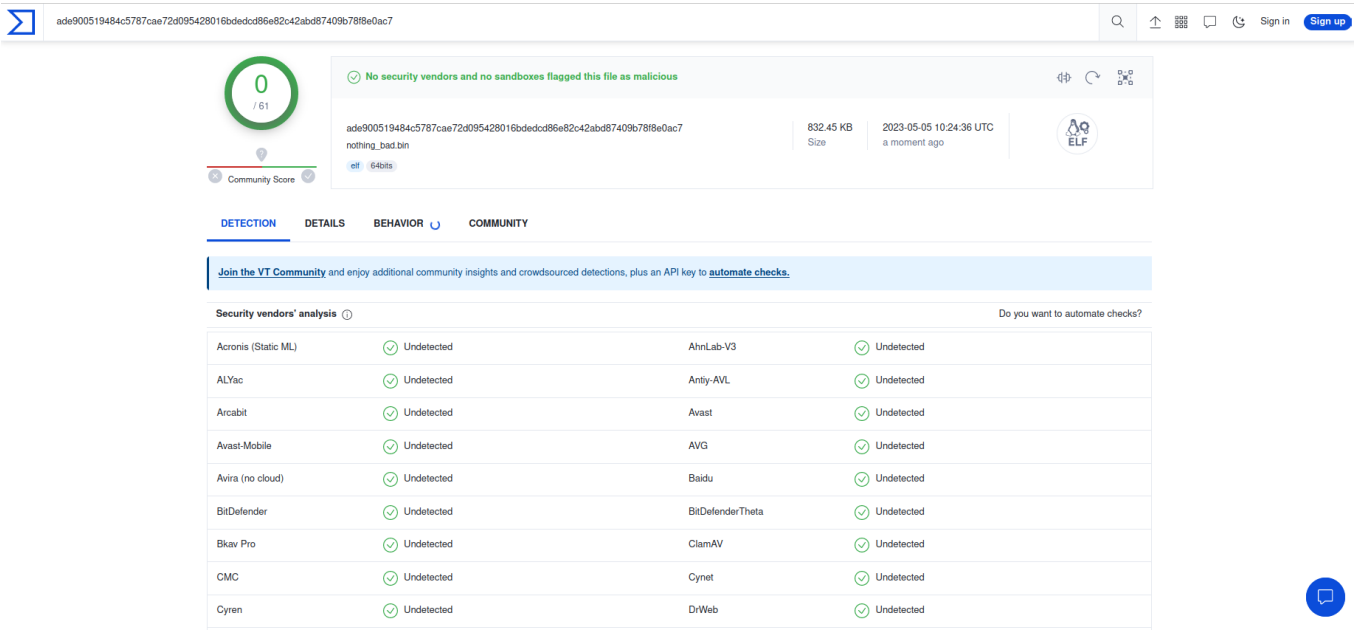
助教: 許哲睿 (r11944076@ntu.edu.tw)
TA hour: (Wed) 9:00 ~ 12:00 at 德田307

Story

In this challenge, a binary was provided.

```
nothing_bad.bin: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, BuildID[sha1]=5618d9c33e1605e479b2c3755ff
798c279e95e93, for GNU/Linux 3.2.0, stripped
```

The IR team informed you that they discovered a binary file on a compromised system which they suspected to be a hacking tool. However, after checking Virus Total, mainstream AV softwares indicated that it was benign.



Can we close the case now?

How to solve?

You runned the binary on your VM, and you found out that port 7373 now opened.

```
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
netstat: no support for `AF INET (sctp)' on this system.
netstat: no support for `AF INET (sctp)' on this system.
tcp        0      0 0.0.0.0:7373            0.0.0.0:*               LISTEN      335808/./nothing_ba
netstat: no support for `AF IPX' on this system.
netstat: no support for `AF AX25' on this system.
netstat: no support for `AF X25' on this system.
netstat: no support for `AF NETROM' on this system.
netstat: no support for `AF ROSE' on this system.
```

A friend told you that the binary might be communicating using the following protocol. (She always knew.)

```
// data format ( in bytes )
// | 0x0 ---------------------------------- 0x0f |  : | "nslab_w31c0m3_U\x00"  |
// | 0x10 -- 0x13 | 0x14 ------------------ 0x1f |  : | command_id | data..... |
```

The command_id seemed to be 4 bytes ascii strings consisting only of digits.

So with the help of pwntools (or any socket programming tools you like), you got a script to interact with the hacking tool.

```
from pwn import *
import time


def connect_to_c2_agent( cmdId, data ):
    io = remote("127.0.0.1", 7373)
    header = b'nslab_w31c0m3_U\x00'
    cmd    = cmdId + b'\x00' * ( 4 - len(cmdId) )
    content = header + cmd + data
    io.send(content)
    // whatever you want.
    io.close()
```

Now you need to figure out the mapping between cmdId and hacktool's behaviours.

Your TODO list

1. fuzz cmdId (All you have to do is brute-forcing the 4 bytes. Need no fancy techniques.)
2. nothing_bad.bin will tell you what to do with each cmdId.
3. nothing_bad.bin seems to write something on your system. locate the file and determine the contents inside.

Submission

upload a [student_id].pdf with following contents

1. (40 pt): Your fuzzing script
2. (30 pt): What does each cmdId mapped to?
3. (30 pt): The location of the file generated by nothing_bad.bin, as well as the contents inside the file.
4. (10 pt) (bonus): How did the contents written by nothing_bad.bin get generated?

Please don't copy your firends' answer.