Reverse start

1. file <something>
    - 起手式，先確認檔案是什麼類型的檔案
2. strings -n <min-len> <something>
    - 通常會搭配 grep: strings <something> | grep "bin/sh"
3. 如果是執行檔就執行看看
4. objdump -M intel -d <binary> or IDA pro(windows)
5. strace / ltrace
    - strace <binary>: 查看 binary 執行時的 system call 和 signal
    - ltrace <binary>: 查看 binary 執行時的 library call
6. gdb gogo !!

C / C++ Reverse

c reverse - 這裡沒有筆記！

c++ Name Mangling

- 修飾名稱並附加資訊
- 目的:
    - compiler 和 li

Reverse startnker 可以辨識同名參數不同的 function

```
* 在不同的 class
```

Reverse start、template、namespace 底下可以有同樣名稱的 function

- 還原方法：c++filt

Reverse start<name>

Reverse start

IDA - 這裡沒有筆記

Reverse start！

Reverse start

代碼混淆

Reverse start

插入垃圾指令

- 不影響原程式正常執行的插入一些沒有用的指令

替換與原指令等價的指令

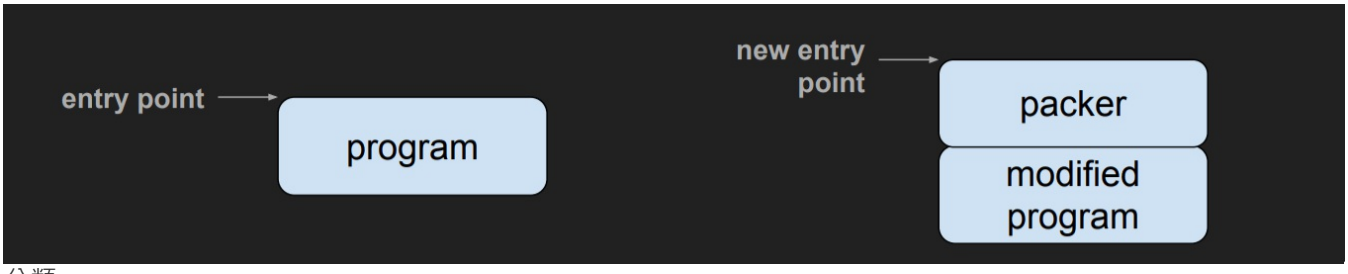
```
jmp label       =>  push label
                    ret
```

```
mov rdi, rax    =>  push rax
                    pop rdi
```

```
add rax, 5      =>  add rax, 10
                    sub rax, 3
                    sub rax, 5
                    add rax, 3
```

殼：

- 在原程式外加一層保護，用來防止修改或反編譯，在 `runtime` 才將真正的 `program` 解回來執行



- 分類

    - 壓縮殼: UPX、ASPCAK、TELOCK
    - 加密殼: ASPROTECT、ARMADILLO
    - 自己實作的殼
- 查殼: PEiD (不見得找的到)
- UPX: https://github.com/upx/upx
    - 判定 UPX: strings 尋找 UPX 字串

VM 保護

- 自行實作指令集並交由 VM 執行，難以破解 Orz

windows reverse - 待補