

Foundations of Cybersecurity C and C++ Secure Coding

Gianluca Dini

Dept. of Information Engineering

University of Pisa

Email: gianluca.dini@unipi.it

Version: 2022-03-17

1

Credits



UNIVERSITÀ DI PISA

- These slides come from a version originally produced by Dr. Pericle Perazzo

Mar-22

Pointer subterfuge

2

2

C and C++ Secure Coding

POINTER SUBTERFUGE

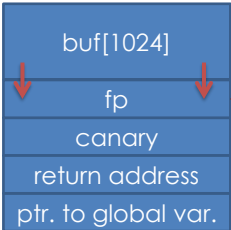
3

3

Pointer Subterfuge



stack:




```
void func() {
  char buf[1024];
  void (*fp)() = &good_func;
  /* ... */
  if (gets(buf) == NULL) {
    /* Handle error */
  }
  /* ... */
  fp();
}
void good_func() { /*...*/ }
void bad_func() { /*...*/ }
```



4

4




UNIVERSITÀ DI PISA

Pointer Subterfuge

- Pointer subterfuge is an exploit that modifies a pointer's value
- Two classes
 - Object pointer subterfuge
 - Pointer used only for read: data leakage
 - Pointer used for write: integrity violation and arbitrary code execution
 - Function pointer subterfuge
 - Arbitrary code execution
 - Some function pointers are «implicit»
 - Global Offset Table (GOT)

Mar-22
Pointer subterfuge
5

5



UNIVERSITÀ DI PISA

Countermeasures

- Fix pointer overwrite vulnerabilities
 - The best countermeasure
- Pointer encryption
 - Pointer stays in memory in encrypted form
 - Program decrypts it before use
 - Pointer subterfuge is still possible, but it has random effects (most probably segmentation fault)
 - Encryption key must be kept secret

Mar-22
Pointer subterfuge
6

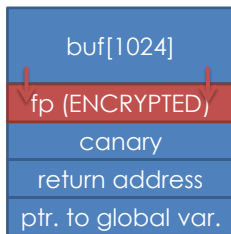
6

Pointer Encryption

No equivalent in Unix

- Nothing in kernel
- Nothing in std. Libs
- Nothing in gcc options

stack:



```
#include <Windows.h>
```

```
void func() {
    char buf[1024];
    void (*fp)() = EncodePointer(&good_func);
    /* ... */
```

```
    if (gets(buf) == NULL) {
        /* Handle error */
    }
```

```
    /* ... */
```

```
    void (*decr_fp)() =
        (void (*)()) DecodePointer(fp);
    decr_fp();
}
```

```
void good_func() { /*...*/ }
void bad_func() { /*...*/ }
```

→ ?
(segmentation fault)

