Introduction of Secure Programming

Study Groups at NCYU

Chih-Hsuan Yang(SCC)

zxc25077667@pm.me

May 10, 2021

About me

- 楊志璿
- NSYSU Information security club founder
- Resume
- Linux, Modern C++

Before talk

A book

Secure Programming

Not hard, no much pages.

Stories

TA experience.

List some cases.

It is impossible of talking about secure programming without programming.

Focus on fundamental qualities, security rather than attacks.

Great oaks from little acorns grow.

Outline

1. Background

2. Programmer's qualities

2.1 Arithmetic overflow

2.2 ReDoS

2.3 RAII

3. Memory safe

3.1 Buffer overflow

3.2 Use after free

3.3 Memory leakage

4. Call out to other routines

4.1 Injections

4.2 Parsing

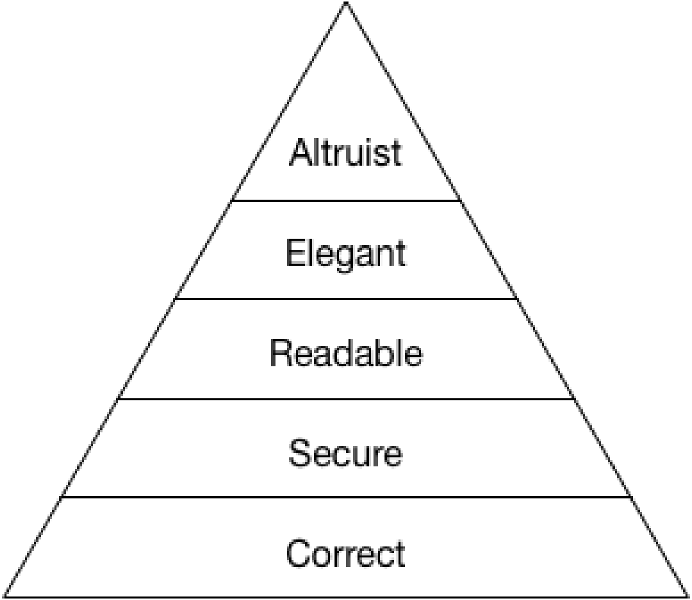
5. Others

5.1 Language features

5.2 Authentication

Background

Maslow’ s pyramid of code review



Altruist

Elegant

Readable

Secure

Correct

Maslow’s pyramid of code review

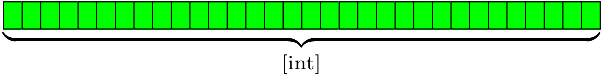
- Correct：做到預期的行為了嗎？能夠處理各式邊際狀況嗎？即便其他人修改程式碼後，主體的行為仍符合預期嗎？
- Secure：面對各式輸入條件或攻擊，程式仍可正確運作嗎？
- Readable：程式碼易於理解和維護嗎？
- Elegant：程式碼夠「美」嗎？可以簡潔又清晰地解決問題嗎？
- Altruist：除了滿足現有的狀況，軟體在日後能夠重用嗎？甚至能夠抽離一部分元件，給其他專案使用嗎？

Programmer's qualities

Arithmetic overflow

Data Model					
Type	LP32	ILP32	LP64	ILP64	LLP64
char	8	8	8	8	8
short	16	16	16	16	16
int	16	32	32	64	32
long	32	32	64	64	32
long long	64	64	64	64	64
pointer	32	32	64	64	64

Bits field



2's complement

Eg:

- 0x1234ABCD
- 0x00BADBAD
- 0xFFFFFFFF

Integer overflow

2002 FreeBSD

```c
#define KSIZE 1024
char kbuf[KSIZE];
int copy_from_kernel(void *user_dest, int maxlen) {
  int len = KSIZE < maxlen ? KSIZE : maxlen;
  memcpy(user_dest, kbuf, len);
  return len;
}
```

What if maxlen < 0?

Take maxlen as -1, try it!

Integer overflow

2002 External data representation (XDR)

```c
void *copy_elements(void *ele_src[], int ele_cnt, int ele_size) {
  void *result = malloc(ele_cnt * ele_size);
  if (result == NULL)
    return NULL;
  void *next = result;
  for (int i = 0; i < ele_cnt; i++) {
    memcpy(next, ele_src[i], ele_size);
    next += ele_size;
  }
  return result;
}
```

What if ele_cnt $= 2^{22}$, ele_size $= 2^{10}$?

Try it!

Binary search

```c
int wrong(int *arr, size_t len, int target)
{
    int begin = 0, end = len;
    while (begin <= end)
    {
        int mid = (begin + end) / 2;
        if (arr[mid] == target)
            return mid;
        else if (arr[mid] < target)
            end = mid;
        else
            begin = mid;
    }
    return -1;
}
```

Binary search

```c
int correct(int *arr, size_t len, int target)
{
    int begin = 0, end = len;
    while (begin <= end)
    {
        int mid = (begin >> 1) + (end >> 1);
        if (arr[mid] == target)
            return mid;
        else if (arr[mid] < target)
            end = mid;
        else
            begin = mid;
    }
    return -1;
}
```

Binary search

- 1946 idea
- 1960 mathematical analysis
- 1988 find bugs.

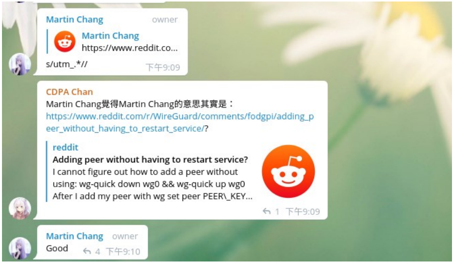
Donald Knuth

Although the basic idea of binary search is comparatively straightforward, the details can be surprisingly tricky.

Appendix here - Donald Knuth

- $\TeX$
- The Art of Computer Programming (TAOCP)

ReDoS



Martin Chang owner

Martin Chang
https://www.reddit.co...

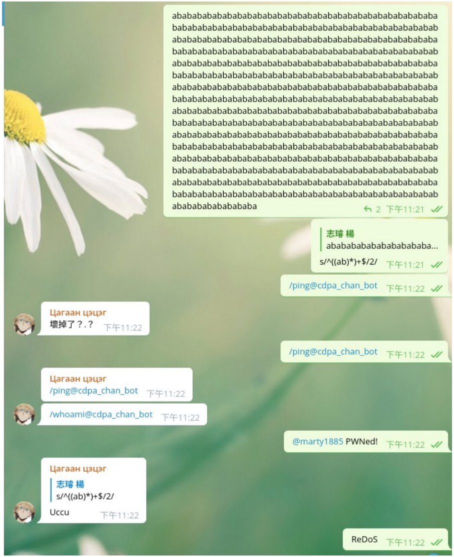
s/utm_.*// 下午9:09

CDPA Chan
Martin Chang覺得Martin Chang的意思其實是：
https://www.reddit.com/r/WireGuard/comments/fodgpi/adding_peer_without_having_to_restart_service/?

reddit
Adding peer without having to restart service?
I cannot figure out how to add a peer without using: wg-quick down wg0 && wg-quick up wg0 After I add my peer with wg set peer PEER_KEY...

1 下午9:09

Martin Chang owner
Good 4 下午9:10



ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababababababababababababababababababababa
bababababababababababababababababababababababababababab
ababababababababa
2 下午11:21

志瑋 楊
ababababababababababa...
s/^((ab)*)+$/2/ 下午11:21

/ping@cdpa_chan_bot 下午11:22

Цагаан цэцэг
壞掉了？.？ 下午11:22

/ping@cdpa_chan_bot 下午11:22

Цагаан цэцэг
/ping@cdpa_chan_bot 下午11:22

/whoami@cdpa_chan_bot 下午11:22

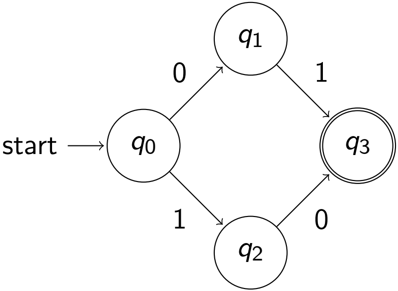
@marty1885 PWNed! 下午11:22

Цагаан цэцэг
志瑋 楊
s/^((ab)*)+$/2/
Uccu 下午11:22

ReDoS 下午11:22

Regex

- Regular expression
- Finite state machine



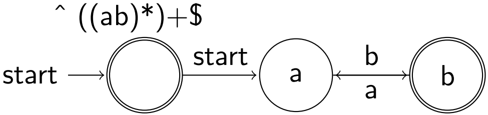
Regex basic

Regex 101

Let's try:

- A brown fox jumps over the lazy dog.
- Student ID.
- Binary search code.
- Email.

Halting problem



Input

ababababababababababababa

Halting problem

The engine will first try (abababababababababababab) but that fails because of that extra a. This causes catastrophic bracktracking, because our pattern (ab)*, in a show of good faith, will release one of it's captures (it will "backtrack") and let the outer pattern try again.

- (abababababababababababab) - Nope
- (ababababababababababab)(ab) - Nope
- (abababababababababab)(abab) - Nope
- (abababababababababab)(ab)(ab) - Nope
- (ababababababababab)(ababab) - Nope
- (ababababababababab)(abab)(ab) - Nope
- (ababababababababab)(ab)(abab) - Nope
- (ababababababababab)(ab)(ab)(ab) - Nope
- . . .
- (ab)(ab)(ab)(ab)(ab)(ab)(ab)(ab)(ab)(ab)(ab)(ab) - Nope

Halting problem

$$dp[0] = 1$$

$$dp[N] = \sum_{i=0}^{N} dp[i] + dp[N - i]$$

$$\sim O(3^N)$$

ReDoS

So, please check what you did.
User provides regex should have a timeout threshold.
Don't believe the user inputs.

RAII

- Resource Acquisition Is Initialization
- Bjarne Stroustrup
- Constructor, destructor
- Don't memorize.

變禿了，也變强了。Modern C++的逆襲

RAII - Obj

The Obj.hpp

```cpp
#include <iostream>
class Object
{
    int *arr;

public:
    Object()
    {
        arr = new int[5];
    }
    ~Object()
    {
        std::cerr << "Freed" << std::endl;
        delete[] arr;
    }
};
```

RAII - lifetime

Recall: Objects.

```
#include <iostream>
#include "Obj.hpp"

Object obj;

int main()
{
    Object _obj;
    return 0;
}
```

RAII - exceptions

```
#include <exception>
#include "Obj.hpp"

void f()
{
    Object o;
    throw std::runtime_error("Oops!!");
    return;
}

int main()
{
    try {
        f();
    } catch (const std::exception &e) {
        std::cerr << e.what() << '\n';
    }
    return 0;
}
```

Memory safe

Common mistakes of junior programmers.

Buffer overflow

```c
#include <stdio.h>
int main()
{
    char arr[8];
    char buf[16] = {0};
    gets(arr);
    printf("%s\n", buf);
    return 0;
}
```

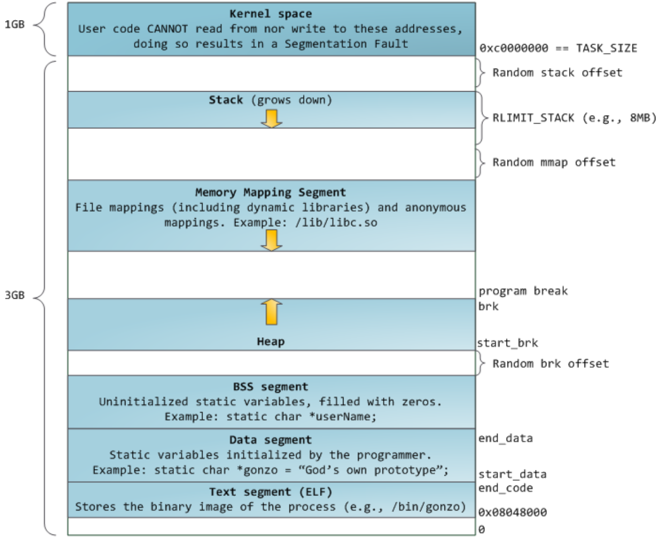
Buffer overflow

strcpy, strncpy

```c
#include <string.h>
int main()
{
    char arr[8];
    char *str = "OVERFLOWAAAAAAAAAAAAAAAAaAA";
    strcpy(arr, str);
    return 0;
}
```

```c
#include <string.h>
int main()
{
    char arr[8];
    char *str = "OVERFLOWAAAAAAAAAAAAAAAAaAA";
    strncpy(arr, str, sizeof(arr));
    return 0;
}
```

Buffer overflow



1GB
3GB
Kernel space
User code CANNOT read from nor write to these addresses, doing so results in a Segmentation Fault
0xc0000000 == TASK_SIZE
Random stack offset
Stack (grows down)
RLIMIT_STACK (e.g., 8MB)
Random mmap offset
Memory Mapping Segment
File mappings (including dynamic libraries) and anonymous mappings. Example: /lib/libc.so
program break
brk
Heap
start_brk
Random brk offset
BSS segment
Uninitialized static variables, filled with zeros.
Example: static char *userName;
end_data
Data segment
Static variables initialized by the programmer.
Example: static char *gonzo = "God's own prototype";
start_data
end_code
Text segment (ELF)
Stores the binary image of the process (e.g., /bin/gonzo)
0x08048000
0

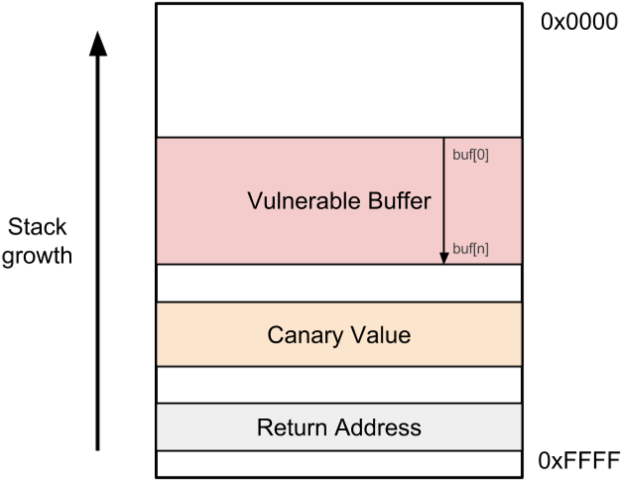
Test for stack pointer.

Buffer overflow

bof on stack

```c
#include <stdio.h>
int main()
{
    char arr[8];
    char buf[16] = {0};
    gets(arr);
    printf("%s\n", buf);
    return 0;
}
```

Buffer overflow



Buffer overflow

canary, so why the strcpy will crash?

```c
#include <string.h>
int main()
{
    char arr[8];
    char *str = "OVERFLOWAAAAAAAAAAAAAAAAaAA";
    strcpy(arr, str);
    return 0;
}
```

Out of bound!

Use after free

Why?

Performance issue.

- History

Use after free

```cpp
#include <functional>
#include <iostream>
#include <string>

auto bad()
{
    std::string loc_str("loc_str");
    return std::ref(loc_str);
}

int main()
{
    auto no = bad();
    std::cout << no.get() << std::endl;
}
```

Use after free

$ cppcheck src/uaf/uaf1.cpp

```
Checking uaf1.cpp ...
uaf1.cpp:7:20: error: Returning object that points to
    local variable ’loc_str’ that will be invalid when
     returning. [returnDanglingLifetime]
    return std::ref(loc_str);
                   ^
uaf1.cpp:7:21: note: Passed to ’ref’.
    return std::ref(loc_str);
                    ^
uaf1.cpp:6:17: note: Variable created here.
    std::string loc_str("loc_str");
                ^
uaf1.cpp:7:20: note: Returning object that points to
    local variable ’loc_str’ that will be invalid when
     returning.
    return std::ref(loc_str);
                   ^
```

Don't underestimate this

Privilege Escalation Via a Use After Free Vulnerability In win32k
Use after free in Network API in Google Chrome
Firefox, Use-after-free in Responsive Design Mode
. . .

Why memory leakage is cirtical?

ATM, cashier counters (Windows 95, XP)
Ubuntu desktop 18.04 (Gnome extension)
Reboot, restart

Memory leakage

```cpp
#include <exception>
#include <iostream>
auto get_ptr()
{
    auto ptr = new int[100];
    throw std::runtime_error("Oh! unexpected runtime error.");
    return ptr;
}

int main()
{
    try
    {
        auto ptr = get_ptr();
    }
    catch (const std::exception &e)
    {
        std::cerr << e.what() << '\n';
    }
}
```

Memory leakage

```cpp
void List::del(List *entry)
{
    List *head = this;
    List **indirect = &head;
    while ((*indirect) != entry)
        indirect = &(*indirect)->next;
    *indirect = entry->next;
    delete entry;
}
int main(int argc, char const *argv[])
{
    List *head = new List;
    head->append(4);
    head->append(10);
    head->append(2);
    print_list(head);
    return 0;
}
```

Memory leakage - Garbage Collection & RAII

GC

Reference counter.

Tracing garbage collection.

RAII

Resource Acquisition Is Initialization

Smart pointer

Call out to other routines

SQL injection

not checked, string concatenation

Test for SQLi

```php
<?php
$_lastname = $_REQUEST['account'];
$query = "SELECT * FROM actors WHERE last_name = '$_lastname'";

var_dump($_lastname);
var_dump($query);
?>
```

Command line injection

Think about the online judge.

```
#include <string>
#include <iostream>
int main()
{
    std::string s;
    std::cin >> s;
    system(s.c_str());
    return 0;
};
```

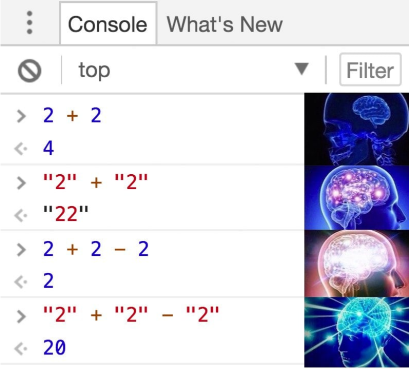
Exploiting URL Parser in Trending Programming Languages

Orange - SSRF HITCON 2017

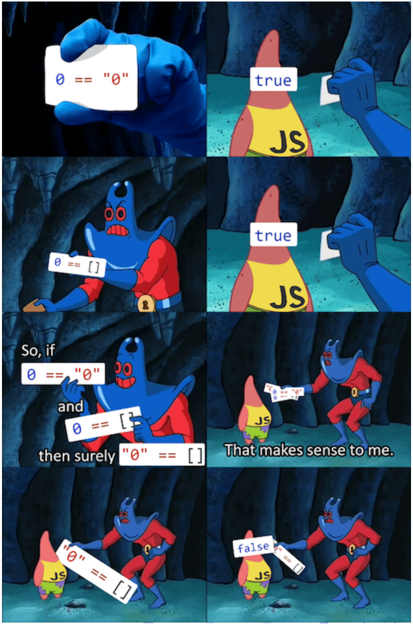
- NodeJS Unicode Failure
- GLibc NSS Features
- Abusing IDNA Standard

Others

Strong types, week types



```
> 2 + 2
<· 4
> "2" + "2"
<· "22"
> 2 + 2 - 2
<· 2
> "2" + "2" - "2"
<· 20
```



Strong types, week types

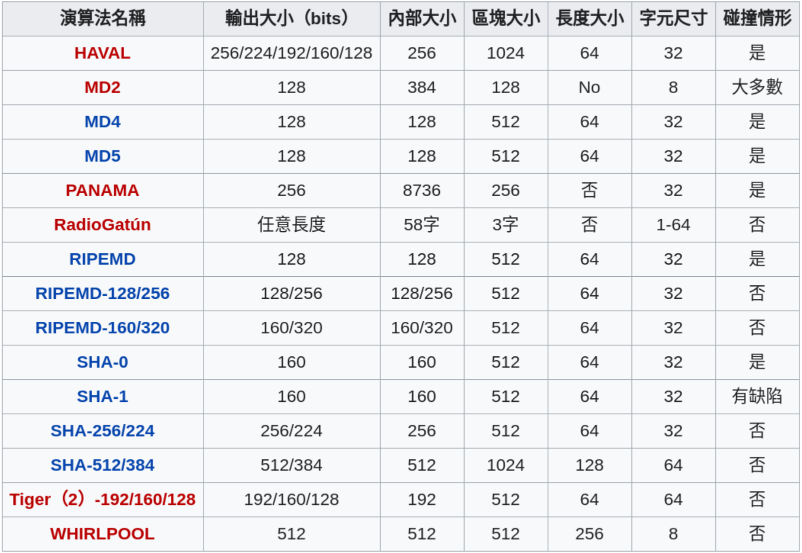
Philosophy: essentialism

Duck typing

If it walks like a duck and it quacks like a duck, then it must be a duck.

Passwords

Hash!! Let's try some.



演算法名稱	輸出大小（bits）	內部大小	區塊大小	長度大小	字元尺寸	碰撞情形
HAVAL	256/224/192/160/128	256	1024	64	32	是
MD2	128	384	128	No	8	大多數
MD4	128	128	512	64	32	是
MD5	128	128	512	64	32	是
PANAMA	256	8736	256	否	32	是
RadioGatún	任意長度	58字	3字	否	1-64	否
RIPEMD	128	128	512	64	32	是
RIPEMD-128/256	128/256	128/256	512	64	32	否
RIPEMD-160/320	160/320	160/320	512	64	32	否
SHA-0	160	160	512	64	32	是
SHA-1	160	160	512	64	32	有缺陷
SHA-256/224	256/224	256	512	64	32	否
SHA-512/384	512/384	512	1024	128	64	否
Tiger（2）-192/160/128	192/160/128	192	512	64	64	否
WHIRLPOOL	512	512	512	256	8	否

Oh, no PHP

```
$hash = "0e878787878787878787878787878787";
if(md5($_GET['pass']) == $hash) {
    echo $FLAG;
}
```

pass: aabg7XSs : Success!

pass: QNKCDZO : Success! ?????

Oh, no PHP

```php
$hash = "0e878787878787878787878787878787";
if(md5($_GET['pass']) == $hash) {
    echo $FLAG;
}
```

pass: aabg7XSs → 0e087386482136013740957780965295

pass: QNKCDZO → 0e830400451993494058024219903391

Oh, no PHP

```
$hash = "0e878787878787878787878787878787";
if(md5($_GET['pass']) == $hash) {
    echo $FLAG;
}
```

”0e087386482136013740957780965295”

→ cast to number

→ scientific notation

→ 0eXXXX = 0