

PHP 安全编码实战经验

黄敏

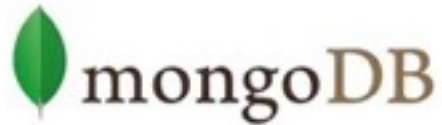
2016/5/11

if you can't explain it
simply,
You don't understand it.

know it then hack it !

一、系统层面的安全





1.1 从 C 缓冲区溢出说起

古老但将持续存在下去的问题

```
#include <stdio.h>

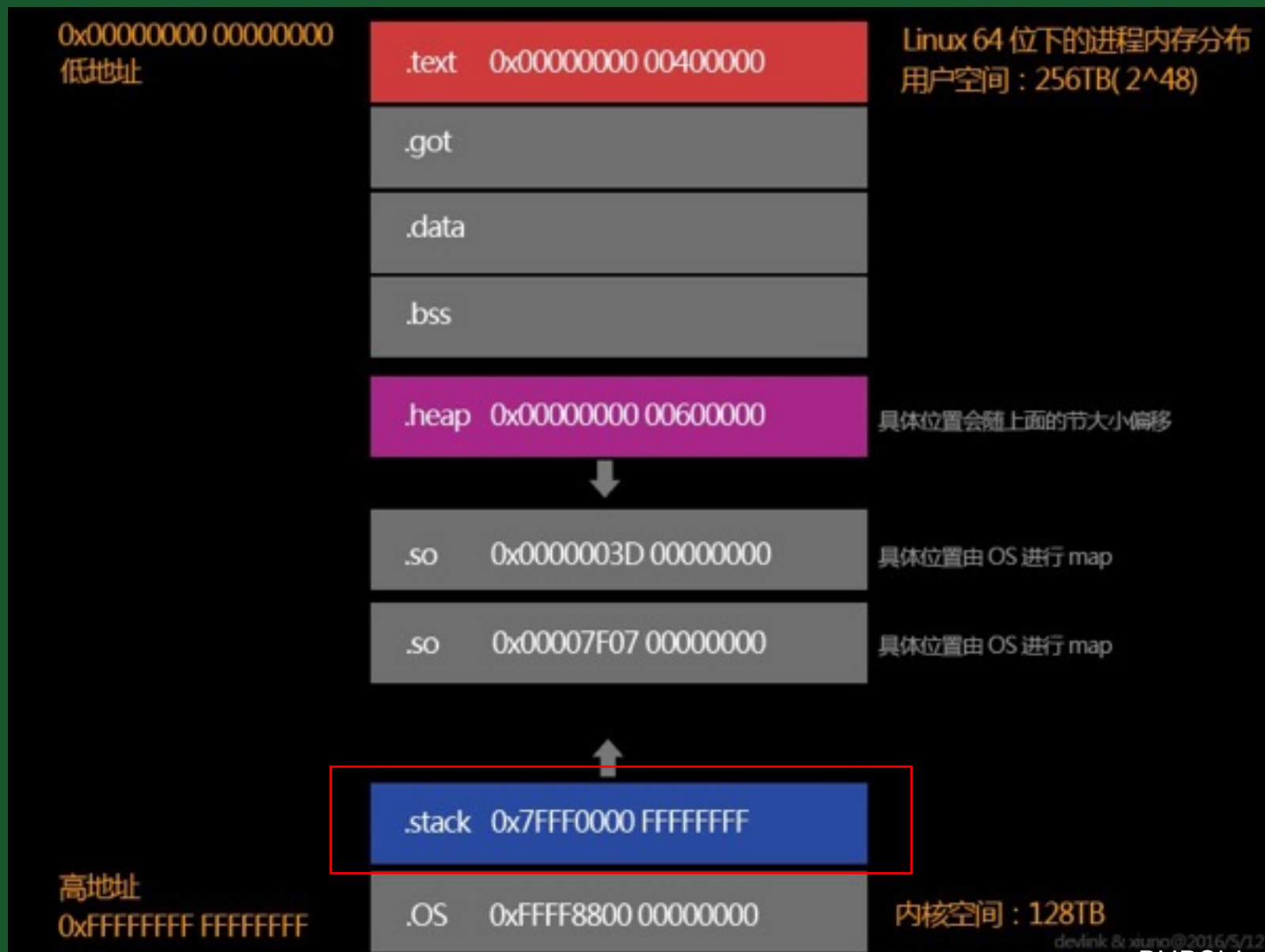
void func() {
    char arr[16];
    strcpy(arr, "ABCDEFGHIJKLMNOPQRSTUVWXYZ");
}

int main() {
    func();
    return 0;
}
```

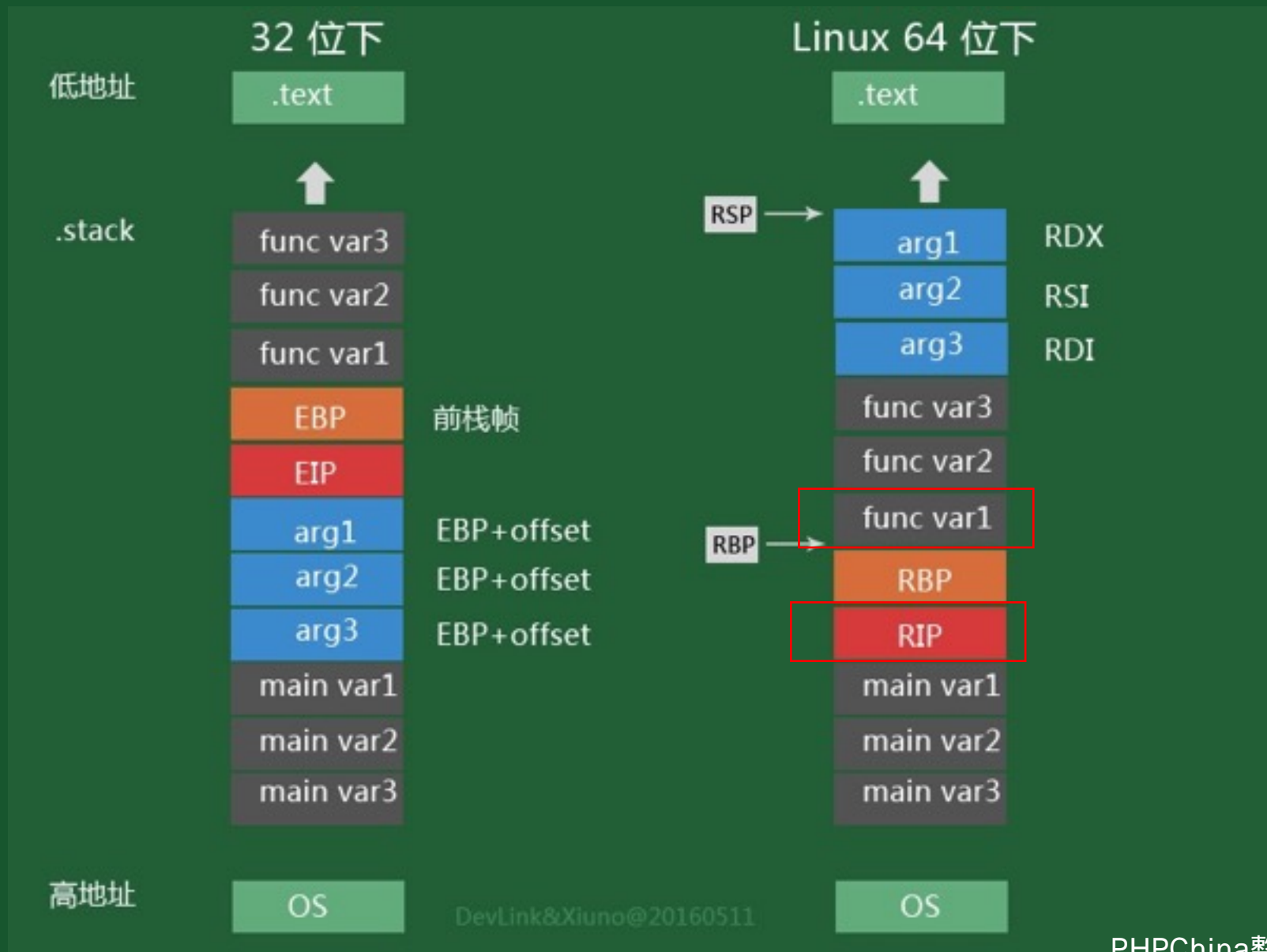
为了贴近实战，我们以 Linux 64 为环境。

事实上，Linux 64 位很具有挑战性。

1.2 Linux 64 位下的内存分布



1.3 Linux 64 位下的栈细节



1.4 Linux 64 位下的栈溢出实例程序

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

```
char **p;
```

```
int func() {
    unsigned char arr[] = "ABCDEFGG":
```

```
    unsigned char shellcode[35] = {
        0xba, 0x58, 0x04, 0x40, 0x00, 0x48, 0xb8, 0x2f,
        0x70, 0x61, 0x73, 0x73, 0x77, 0x64, 0x00, 0x50,
        0x48, 0xb8, 0x63, 0x61, 0x74, 0x20, 0x2f, 0x65,
        0x74, 0x63, 0x50, 0x48, 0x89, 0xe7, 0x48, 0x89,
        0xd0, 0xff, 0xd0};
    system()
```

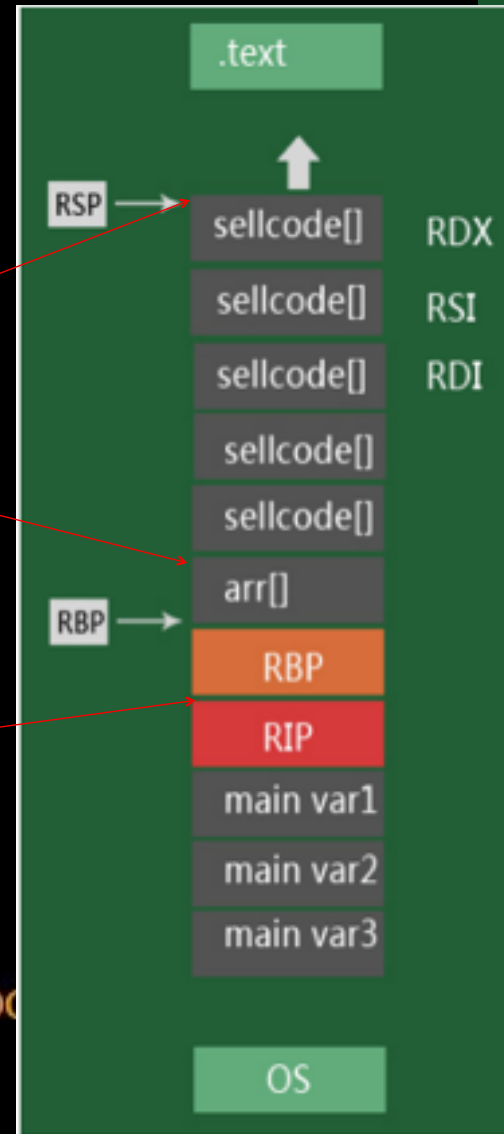
```
    p = (char **)&shellcode;
    memcpy(arr+24, (char*)&p, 8);
```

```
    return 0;
```

```
}

int main() {
    printf("system: %11x\n", system);
    func();
    return 0;
```

```
}
```



gcc -g overflow.c -z execstack -fno-stack-protector -o overflow.o && ./overflow.o

运行结果

```
[root@xiuno shellcode]# ./overflow.o
system: 400458      shellcode call system(" cat /etc/passwd" ) successfully
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
abrt:x:173:173:/:/etc/abrt:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
ntp:x:38:38:/:/etc/ntp:/sbin/nologin
saslauth:x:499:76:"Saslauthd user":/var/empty/saslauth:/sbin/nologin
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
mysql:x:500:500:/:/home/mysql:/sbin/nologin
www:x:501:501:/:/home/www:/sbin/nologin
exim:x:93:93:/:/var/spool/exim:/sbin/nologin
Segmentation fault
```

func() 汇编

```
(gdb) disass /mr func
Dump of assembler code for function func:
9      int func() {
      0x000000000400564: <<0>:      55      push    %rbp
      0x000000000400565: <<1>:      48 89 e5  mov     %r1p,%rbp
      0x000000000400568: <<4>:      48 83 ec 40  sub     $0x40,%rsp

10     unsigned char arr[] = "ABCDEFGH";
      0x00000000040056c: <<8>:      48 8b 05 e5 01 00 00  mov     0x1e5(%rip),%rax      # 0x400758 <__dso_handle+8>
      0x000000000400573: <<15>:     48 89 45 f0  mov     %rax,-0x10(%rbp)

11     unsigned char shellcode[35] = {
12         0xba,0x58,0x04,0x40,0x00,0x48,0xb8,0x2f,
13         0x70,0x61,0x73,0x73,0x77,0x64,0x00,0x50,
14         0x48,0xb8,0x61,0x61,0x74,0x20,0x2f,0x61,
15         0x74,0x61,0x50,0x48,0x89,0xe7,0x48,0x89,
16         0xd0,0xff,0xd0};
      0x000000000400577: <<19>:     c6 45 c0 ba  movb    $0xba,-0x40(%rbp)
      0x00000000040057b: <<23>:     c6 45 c1 58  movb    $0x58,-0x3f(%rbp)
      0x00000000040057f: <<27>:     c6 45 c2 04  movb    $0x4,-0x3e(%rbp)
      0x000000000400583: <<31>:     c6 45 c3 40  movb    $0x40,-0x3d(%rbp)
      0x000000000400587: <<35>:     c6 45 c4 00  movb    $0x0,-0x3c(%rbp)
      0x00000000040058b: <<39>:     c6 45 c5 48  movb    $0x48,-0x3b(%rbp)
      0x00000000040058f: <<43>:     c6 45 c6 b8  movb    $0xb8,-0x3a(%rbp)
      0x000000000400593: <<47>:     c6 45 c7 2f  movb    $0x2f,-0x39(%rbp)
      0x000000000400597: <<51>:     c6 45 c8 70  movb    $0x70,-0x38(%rbp)
      0x00000000040059b: <<55>:     c6 45 c9 61  movb    $0x61,-0x37(%rbp)
      0x00000000040059f: <<59>:     c6 45 ca 73  movb    $0x73,-0x36(%rbp)
      0x0000000004005a3: <<63>:     c6 45 cb 73  movb    $0x73,-0x35(%rbp)
      0x0000000004005a7: <<67>:     c6 45 cc 77  movb    $0x77,-0x34(%rbp)
      0x0000000004005ab: <<71>:     c6 45 cd 64  movb    $0x64,-0x33(%rbp)
      0x0000000004005af: <<75>:     c6 45 ce 00  movb    $0x0,-0x32(%rbp)
      0x0000000004005b3: <<79>:     c6 45 cf 50  movb    $0x50,-0x31(%rbp)
      0x0000000004005b7: <<83>:     c6 45 d0 48  movb    $0x48,-0x30(%rbp)
      0x0000000004005bb: <<87>:     c6 45 d1 b8  movb    $0xb8,-0x2f(%rbp)
      0x0000000004005bf: <<91>:     c6 45 d2 63  movb    $0x63,-0x2e(%rbp)
      0x0000000004005c3: <<95>:     c6 45 d3 61  movb    $0x61,-0x2d(%rbp)
      0x0000000004005c7: <<99>:     c6 45 d4 74  movb    $0x74,-0x2c(%rbp)
      0x0000000004005cb: <<103>:    c6 45 d5 20  movb    $0x20,-0x2b(%rbp)
      0x0000000004005cf: <<107>:    c6 45 d6 2f  movb    $0x2f,-0x2a(%rbp)
      0x0000000004005d3: <<111>:    c6 45 d7 65  movb    $0x65,-0x29(%rbp)
      0x0000000004005d7: <<115>:    c6 45 d8 74  movb    $0x74,-0x28(%rbp)
      0x0000000004005db: <<119>:    c6 45 d9 63  movb    $0x63,-0x27(%rbp)
      0x0000000004005df: <<123>:    c6 45 da 50  movb    $0x50,-0x26(%rbp)
      0x0000000004005e3: <<127>:    c6 45 db 48  movb    $0x48,-0x25(%rbp)
      0x0000000004005e7: <<131>:    c6 45 dc 89  movb    $0x89,-0x24(%rbp)
      0x0000000004005eb: <<135>:    c6 45 dd e7  movb    $0xe7,-0x23(%rbp)
      0x0000000004005ef: <<139>:    c6 45 de 48  movb    $0x48,-0x22(%rbp)
      0x0000000004005f3: <<143>:    c6 45 df 89  movb    $0x89,-0x21(%rbp)
      0x0000000004005f7: <<147>:    c6 45 e0 d0  movb    $0xd0,-0x20(%rbp)
      0x0000000004005fb: <<151>:    c6 45 e1 ff  movb    $0xff,-0x1f(%rbp)
      0x0000000004005ff: <<155>:    c6 45 e2 d0  movb    $0xd0,-0x1e(%rbp)

17
18     p = (char *)&shellcode;
      0x000000000400603: <<159>:    48 8d 45 c0  lea     -0x40(%rbp),%rax
      0x000000000400607: <<163>:    48 89 05 42 04 20 00  mov     %rax,0x200442(%rip)      # 0x600a50 <p>

19     memcpy(arr+24, (char *)&p, 8);
      0x00000000040060e: <<170>:    48 8d 45 f0  lea     -0x10(%rbp),%rax
      0x000000000400612: <<174>:    48 83 c0 18  add     $0x18,%rax
      0x000000000400616: <<178>:    ba 08 00 00 00  mov     $0x8,%edx
      0x00000000040061b: <<183>:    be 70 0a 60 00  mov     $0x600a50,%esi
      0x000000000400620: <<188>:    48 89 c7      mov     %rax,%rdi
      0x000000000400623: <<191>:    e8 40 fe ff callq 0x400468 <memcpy@plt> 覆盖 RIP

20     return 0;
      0x000000000400628: <<196>:    b8 00 00 00 00  mov     $0x0,%eax

21 }
      0x00000000040062d: <<201>:    c9      leaveq
      0x00000000040062e: <<202>:    c3      retq

End of assembler dump.
```


func() RIP 覆盖

```
20      memcpy(arr+24, (char*)&p, 8);
(gdb) x/64xb arr
0x7fffffffefe5e0: 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x00 arr
0x7fffffffefe5e8: 0xe0 0x03 0x40 0x00 0x00 0x00 0x00 0x00
0x7fffffffefe5f0: 0x00 0xe6 0xff 0xff 0xff 0x7f 0x00 0x00
0x7fffffffefe5f8: 0x9d 0x05 0x40 0x00 0x00 0x00 0x00 0x00 RIP
0x7fffffffefe600: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x7fffffffefe608: 0x1d 0xed 0xe1 0x02 0x3d 0x00 0x00 0x00
0x7fffffffefe610: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x7fffffffefe618: 0xe8 0xe6 0xff 0xff 0xff 0x7f 0x00 0x00
(gdb) ni
0x0000000000400572      20      memcpy(arr+24, (char*)&p, 8);
(gdb)
0x0000000000400576      20      memcpy(arr+24, (char*)&p, 8);
(gdb)
0x000000000040057b      20      memcpy(arr+24, (char*)&p, 8);
(gdb)
0x0000000000400580      20      memcpy(arr+24, (char*)&p, 8);
(gdb)
0x0000000000400583      20      memcpy(arr+24, (char*)&p, 8);
(gdb)
31      return 0;
(gdb)
32  }
```

```
(gdb) x/64xb arr
0x7fffffffefe5e0: 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x00
0x7fffffffefe5e8: 0xe0 0x03 0x40 0x00 0x00 0x00 0x00 0x00
0x7fffffffefe5f0: 0x00 0xe6 0xff 0xff 0xff 0x7f 0x00 0x00
0x7fffffffefe5f8: 0xb0 0xe5 0xff 0xff 0xff 0x7f 0x00 0x00 RIP
0x7fffffffefe600: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x7fffffffefe608: 0x1d 0xed 0xe1 0x02 0x3d 0x00 0x00 0x00
0x7fffffffefe610: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x7fffffffefe618: 0xe8 0xe6 0xff 0xff 0xff 0x7f 0x00 0x00
```

进入栈区

```
Breakpoint 1, func () at overflow.c:19
19      memcpy(arr+24, (char*)&p, 8);
Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.132.el6_5.2.x86_64
(gdb) si
0x00000000000400612      19      memcpy(arr+24, (char*)&p, 8);
(gdb) ni
0x00000000000400616      19      memcpy(arr+24, (char*)&p, 8);
(gdb)
0x0000000000040061b      19      memcpy(arr+24, (char*)&p, 8);
(gdb) ni
0x00000000000400620      19      memcpy(arr+24, (char*)&p, 8);
(gdb)
0x00000000000400623      19      memcpy(arr+24, (char*)&p, 8);
(gdb)
20      return 0;
(gdb) si
21  }
(gdb) x/16i $rip
=> 0x40062d <func+201>: leaveq
0x40062e <func+202>: retq
0x40062f <main>: push %rbp
0x400630 <main+1>: mov %rsp,%rbp
0x400633 <main+4>: mov $0x400760,%eax
0x400638 <main+9>: mov $0x400458,%esi
0x40063d <main+14>: mov %rax,%rdi
0x400640 <main+17>: mov $0x0,%eax
0x400645 <main+22>: callq 0x400438 <printf@plt>
0x40064a <main+27>: mov $0x0,%eax
0x40064f <main+32>: callq 0x400564 <func>
0x400654 <main+37>: mov $0x0,%eax
0x400659 <main+42>: leaveq
0x40065a <main+43>: retq
0x40065b: nop
0x40065c: nop
(gdb) si
0x0000000000040062e in func () at overflow.c:21
21  }
(gdb) si
0x00007fffffff5a0 in ?? () 进入了栈区
(gdb) x/16i $rip
=> 0x7fffffff5a0: mov $0x400458,%edx
0x7fffffff5a5: movabs $0x6477737361702f,%rax
0x7fffffff5af: push %rax
0x7fffffff5b0: movabs $0x6374652f20746163,%rax
0x7fffffff5ba: push %rax
0x7fffffff5bb: mov %rsp,%rdi
0x7fffffff5be: mov %rdx,%rax
0x7fffffff5c1: callq *%rax
0x7fffffff5c3: add %al,(%rax)
0x7fffffff5c5: add %al,(%rax)
0x7fffffff5c7: add %al,(%rax)
0x7fffffff5c9: repz sbb %al,(%rbx)
0x7fffffff5cc: cmp $0x41000000,%eax
0x7fffffff5d1: rex.X
0x7fffffff5d2: rex.XB
0x7fffffff5d3: rex.R
(gdb)
```

shellcode

Linux 64 位下 randomize_va_space

```
[root@xiuno shellcode]# echo 0 > /proc/sys/kernel/randomize_va_space
[root@xiuno shellcode]# echo 0 > /proc/sys/kernel/exec-shield
[root@xiuno shellcode]# gcc -g d.c -z execstack -fno-stack-protector
```

```
[root@xiuno shellcode]# echo 1 > /proc/sys/kernel/randomize_va_space
[root@xiuno shellcode]# echo 1 > /proc/sys/kernel/exec-shield
[root@xiuno shellcode]# gcc -g d.c
```

```
cat /proc/2292/maps
```

```
service php-fpm restart
```

```
cat /proc/13511/maps
```

```

7f428daa7000-7f428dabe000 r-xp 00000000 fc:01 3831
7f428dabe000-7f428dcb000 ---p 00017000 fc:01 3831
7f428dcb000-7f428dcbf000 r--p 00017000 fc:01 3831
7f428dcbf000-7f428ddc000 rw-p 00018000 fc:01 3831
7f428ddc000-7f428ddc4000 rw-p 00000000 00:00 0
7f428ddc4000-7f428ddc7000 r-xp 00000000 fc:01 4992
7f428ddc7000-7f428ddcc000 ---p 00003000 fc:01 4992
7f428ddcc000-7f428ddcd7000 r--p 00002000 fc:01 4992
7f428ddcd7000-7f428ddce8000 rw-p 00003000 fc:01 4992
7f428ddce8000-7f428ddf3a000 r-xp 00000000 fc:01 5219
7f428ddf3a000-7f428dfe139000 ---p 00072000 fc:01 5219
7f428dfe139000-7f428e13a000 r--p 00071000 fc:01 5219
7f428e13a000-7f428e13d000 rw-p 00072000 fc:01 5219
7f428e13d000-7f428e13f000 r-xp 00000000 fc:01 3789
7f428e13f000-7f428e33e000 ---p 00002000 fc:01 3789
7f428e33e000-7f428e33f000 r--p 00001000 fc:01 3789
7f428e33f000-7f428e340000 rw-p 00002000 fc:01 3789
7f428e340000-7f428e356000 r-xp 00000000 fc:01 3833
7f428e356000-7f428e35b000 ---p 00016000 fc:01 3833
7f428e35b000-7f428e35f000 r--p 00016000 fc:01 3833
7f428e35f000-7f428e358000 rw-p 00017000 fc:01 3833
7f428e358000-7f428e35a000 rw-p 00000000 00:00 0
7f428e35a000-7f428e364000 r-xp 00000000 fc:01 3807
7f428e364000-7f428e36e4000 ---p 0018a000 fc:01 3807
7f428e36e4000-7f428e36e8000 r--p 0018a000 fc:01 3807
7f428e36e8000-7f428e36e9000 rw-p 0018e000 fc:01 3807
7f428e36e9000-7f428e36ee000 rw-p 00000000 00:00 0
7f428e36ee000-7f428e904000 r-xp 00000000 fc:01 8205
7f428e904000-7f428e9b03000 ---p 00016000 fc:01 8205
7f428e9b03000-7f428eb04000 rw-p 00015000 fc:01 8205
7f428eb04000-7f428eb7e7000 r-xp 00000000 fc:01 2902

```

```

7f481ee000-7f481ef1000 r-xp 00000000 fc:01 3831
7f481ef1000-7f481f15000 ---p 00017000 fc:01 3831
7f481f15000-7f481f16000 r--p 00017000 fc:01 3831
7f481f16000-7f481f17000 rw-p 00018000 fc:01 3831
7f481f17000-7f481f1b000 rw-p 00000000 00:00 0
7f481f1b000-7f481f1e000 r-xp 00000000 fc:01 4992
7f481f1e000-7f481f31d000 ---p 00003000 fc:01 4992
7f481f31d000-7f481f31e000 r--p 00002000 fc:01 4992
7f481f31e000-7f481f31f000 rw-p 00003000 fc:01 4992
7f481f31f000-7f481f391000 r-xp 00000000 fc:01 5219
7f481f391000-7f481f590000 ---p 00072000 fc:01 5219
7f481f590000-7f481f591000 r--p 00071000 fc:01 5219
7f481f591000-7f481f594000 rw-p 00072000 fc:01 5219
7f481f594000-7f481f596000 r-xp 00000000 fc:01 3789
7f481f596000-7f481f795000 ---p 00002000 fc:01 3789
7f481f795000-7f481f796000 r--p 00001000 fc:01 3789
7f481f796000-7f481f797000 rw-p 00002000 fc:01 3789
7f481f797000-7f481f7ad000 r-xp 00000000 fc:01 3833
7f481f7ad000-7f481f9ad000 ---p 00016000 fc:01 3833
7f481f9ad000-7f481f9ae000 r--p 00016000 fc:01 3833
7f481f9ae000-7f481f9af000 rw-p 00017000 fc:01 3833
7f481f9af000-7f481f9b1000 rw-p 00000000 00:00 0
7f481f9b1000-7f481fb3b000 r-xp 00000000 fc:01 3807
7f481fb3b000-7f481fd3b000 ---p 0018a000 fc:01 3807
7f481fd3b000-7f481fd3f000 r--p 0018a000 fc:01 3807
7f481fd3f000-7f481fd40000 rw-p 0018e000 fc:01 3807
7f481fd40000-7f481fd45000 rw-p 00000000 00:00 0
7f481fd45000-7f481fd5b000 r-xp 00000000 fc:01 8205
7f481fd5b000-7f481ff5a000 ---p 00016000 fc:01 8205
7f481ff5a000-7f481ff5b000 rw-p 00015000 fc:01 8205
7f481ff5b000-7f482003e000 r-xp 00000000 fc:01 2902

```

```

/lib64/libpthread-2.12.so
/lib64/libpthread-2.12.so
/lib64/libpthread-2.12.so
/lib64/libpthread-2.12.so

/lib64/libgpg-error.so.0.5.0
/lib64/libgpg-error.so.0.5.0
/lib64/libgpg-error.so.0.5.0
/lib64/libgpg-error.so.0.5.0
/lib64/libgcrypt.so.11.5.3
/lib64/libgcrypt.so.11.5.3
/lib64/libgcrypt.so.11.5.3
/lib64/libgcrypt.so.11.5.3
/lib64/libfreebl3.so
/lib64/libfreebl3.so
/lib64/libfreebl3.so
/lib64/libfreebl3.so
/lib64/libresolv-2.12.so
/lib64/libresolv-2.12.so
/lib64/libresolv-2.12.so
/lib64/libresolv-2.12.so

/lib64/libc-2.12.so
/lib64/libc-2.12.so
/lib64/libc-2.12.so
/lib64/libc-2.12.so

/lib64/libgcc_s-4.4.7-20120601.so
/lib64/libgcc_s-4.4.7-20120601.so
/lib64/libgcc_s-4.4.7-20120601.so
/usr/local/lib/liblibconv.so.2.5.1

```

Linux 64 位下的 shellcode 提取

```
1 <?php
2 echo bin2hex('cat /etc/passwd');
3
4 // 636174202f6574632f706173737764
5
6 // 整理字符串:
7 // 63 61 74 20 2f 65 74 63 2f 70 61 73 73 77 64 00
8 // 63 61 74 20 2f 65 74 63
9 // 2f 70 61 73 73 77 64 00
10
11 // intel little endian
12 echo "\r\n";
13 $s = 'ba d0 03 40 00 48 b8 2f 70 61 73 73 77 64 00 50 48 b8 63 61 74 20 2f 65 74 63 50 48 89 e7 48 89 d0 ff d0';
14 echo str_replace(' ', '0x', $s);
15 echo "\r\n";
16 echo str_word_count($s, NULL, ' ');
17 ?>
```

PHP 处理字符串非常方便。

不能不承认：PHP 世界上最好的编程语言！

Linux 64 位下的 shellcode 提取

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

void func() {
    char *arr;
    asm volatile(
        "mov $0x006477737361702f,%%rax; \n\t"
        "pushq %%rax; \n\t"
        "mov $0x6374652f20746163,%%rax; \n\t"
        "pushq %%rax; \n\t"
        "mov %%rsp,%%rdi; \n\t"
        "mov %0,%%rax; \n\t"
        "callq *%%rax; \n\t"
        :
        : "r"(&system)
        : "%rax"
    );
}

int main() {
    func();
    printf("system: %llx\n", system);

    return 0;
}
```

prepare to fetch shellcode

Linux 64 位下的 shellcode 提取

```
(gdb) disass /mr func
Dump of assembler code for function func:
5      void func() {
      0x00000000004004c4 <+0>:      55      push    %rbp
      0x00000000004004c5 <+1>:      48 89 e5      mov     %rsp,%rbp

6          char *arr;
7          asm volatile(
      0x00000000004004c8 <+4>:      ba d0 03 40 00 mov     $0x4003d0,%edx
      0x00000000004004cd <+9>:      48 b8 2f 70 61 73 73 77 64 00 movabs $0x6477737361702f,%rax
      0x00000000004004d7 <+19>:     50      push    %rax
      0x00000000004004d8 <+20>:     48 b8 63 61 74 20 2f 65 74 63 movabs $0x6374652f20746163,%rax
      0x00000000004004e2 <+30>:     50      push    %rax
      0x00000000004004e3 <+31>:     48 89 e7      mov     %rsp,%rdi
      0x00000000004004e6 <+34>:     48 89 d0      mov     %rdx,%rax
      0x00000000004004e9 <+37>:     ff d0      callq   *%rax

8          "mov $0x006477737361702f,%%rax;\n\t"
9          "pushq %%rax;\n\t"
10         "mov $0x6374652f20746163,%%rax;\n\t"
11         "pushq %%rax;\n\t"
12         "mov %%rsp,%%rdi;\n\t"
13         "mov %0,%%rax;\n\t"
14         "call *%%rax;\n\t"
15         :
16         : "r"(&system)
17         : "%rax"
18     );
19 }
      0x00000000004004eb <+39>:     c9      leaveq  %r15
      0x00000000004004ec <+40>:     c3      retq

End of assembler dump.
```


看起来 64 位 Linux 已经很安全

- 32位下溢出利用难度一般。
- 64 位 Linux 默认开启了地址随机化，传统的 JMP ESP, JMP EBP 难度很高，但是允许在 .text 任意跳转，printf() system() 等可以被调用，仍然具有想象空间和利用的可能。
- 堆栈区域的数据不可以执行，shellcode 即使成功注入也很难执行。
- 64位下溢出利用只有掌握 0day 的人才能做到。
- 防范掌握 0day 的黑客。
- 如何防范？



最小化服务



严格的防火墙策略



隐藏 IP

1.5 最小化服务

- 关闭不必要的服务。
- 服务进程用单独的用户，不要使用 root。
- 服务配置尽量严格。

```
[root@xiuno ~]# cat /root/system_optimizer.sh
#!/bin/bash
SERVICES="abrt-ccpp abrt-oops abrt-d acpid atd auditd avahi-daemon autofs avahi-d
aemon bluetooth certmonger cups cpuspeed dnsmasq firstboot hidd ip6tables kudzu
kdump lvm2-monitor matahari-broker matahari-host matahari-network matahari-servi
ce matahari-sysconfig mdmonitor mcstrans mdmonitor microcode_ctl NetworkManager
netconsole netfs nfslock oddjobd pcsd portmap portreserve postfix psacct rpcgs
sd quota_nld rdisc restorecond rpcidmapd xfs sendmail saslauthd smartd sssd wdac
mon wpa_supplicant ypbind yum-updatesd"
for service in $SERVICES
do
    chkconfig $service off
    service $service stop
done
```

关闭不必要的服务

清楚的知道自己服务器开启了哪些服务，并且挨个配置和测试好服务

```
[root@2015111822513 ~]# netstat -anpt
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:3392           0.0.0.0:*               LISTEN      1968/svnserve
tcp        0      0 0.0.0.0:8008           0.0.0.0:*               LISTEN      1712/nginx
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      1712/nginx
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1730/sshd
tcp        0      0 0.0.0.0:1723           0.0.0.0:*               LISTEN      1783/pptpd
tcp        0      52 103.249.110.61:22      114.111.166.227:32654   ESTABLISHED 14565/sshd
tcp        0      0 103.249.110.61:22      114.111.166.227:46880   ESTABLISHED 14483/sshd
tcp        0      0 103.249.110.61:1723    114.111.166.227:32321   ESTABLISHED 14085/pptpd [114.11
tcp        0      0 :::3306                :::*                   LISTEN      2292/mysqld
tcp        0      0 :::22                  :::*                   LISTEN      1730/sshd
```

1.6 本机严格的防火墙策略

```

# cat /etc/passwd | grep /usr/bin/sudo
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X

# jump sshd
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT

# drop sshd
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

# deny ssh
iptables -t filter -I INPUT 1 -i lo -j ACCEPT
iptables -t filter -I OUTPUT 1 -o lo -j ACCEPT

# deny
iptables -A INPUT -p udp --sport 53 -j ACCEPT
iptables -A OUTPUT -p udp --sport 53 -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT

# ntp
iptables -A INPUT -p tcp --dport=123 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=123 -j ACCEPT
iptables -A INPUT -p tcp --dport=123 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=123 -j ACCEPT

#bind9
iptables -A INPUT -p tcp --dport 23 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 23 -j ACCEPT
iptables -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 25 -j ACCEPT
iptables -A INPUT -p tcp --dport 110 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 110 -j ACCEPT
iptables -A INPUT -p tcp --dport 110 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 110 -j ACCEPT

#rsync
iptables -A INPUT -p tcp --dport 8680 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 8680 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 8680 -j ACCEPT

# samba
iptables -A INPUT -p tcp --dport=139 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=139 -j ACCEPT
iptables -A INPUT -p tcp --dport=445 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=445 -j ACCEPT
iptables -A INPUT -p tcp --dport=137 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=137 -j ACCEPT
iptables -A INPUT -p tcp --dport=138 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=138 -j ACCEPT

# 21, ftp
iptables -A INPUT -p tcp --dport 21 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 21 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 21 -j ACCEPT

# 80
iptables -A INPUT -p tcp --dport=80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=80 -j ACCEPT

# 443
iptables -A INPUT -p tcp --dport=443 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=443 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=443 -j ACCEPT

# 1300, Rsync 192.168.0.2 user
iptables -A INPUT -p tcp --dport=1300 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=1300 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=1300 -j ACCEPT
iptables -A INPUT -p udp --dport=1300 -j ACCEPT
iptables -A OUTPUT -p udp --dport=1300 -j ACCEPT

#typic
iptables -A INPUT -p tcp -s 114.111.229.234 --dport 1024:65535 --dport 873 -j ACCEPT
iptables -A INPUT -p tcp -s 192.168.0.1/3 --dport 1024:65535 --dport 873 -j ACCEPT
iptables -A OUTPUT -p tcp -d 114.111.229.234 --dport 1024:65535 --dport 873 -j ACCEPT
iptables -A OUTPUT -p tcp -d 192.168.0.1/3 --dport 1024:65535 --dport 873 -j ACCEPT

#1690, svn
iptables -A INPUT -p tcp --dport=1690 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=1690 -j ACCEPT
iptables -A OUTPUT -p tcp --dport=1690 -j ACCEPT

# DROP
iptables -A INPUT -p udp --sport 67 --dport 68 -j ACCEPT

# DROP
iptables -A INPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT

#Allow old connection, deny new connection
iptables -A STATE --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT --state --state NEW,INVALID -j DROP

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

/etc/init.d/iptables save
chkconfig iptables --level 2345 on
service iptables start

```

有实力可以考虑硬件防火墙-价格比较昂贵

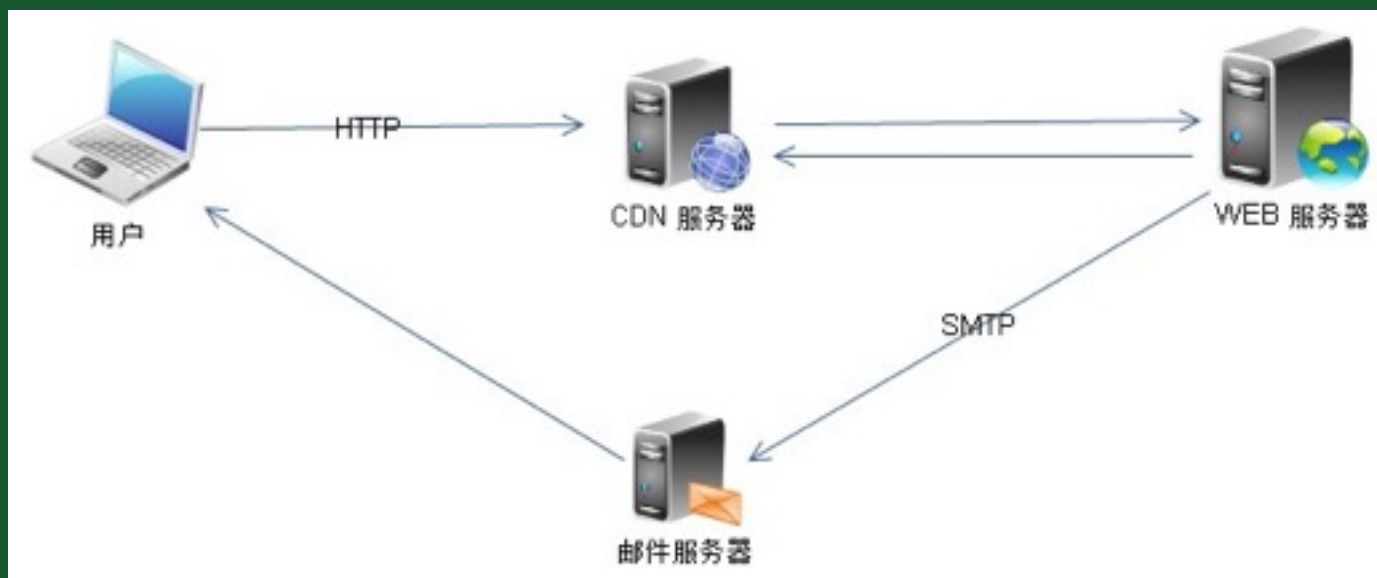


一般都带入侵检测，随时升级策略

1.7 隐藏 IP - CDN

最近几年 CDN 已经普及了，通过 NS 的方式接入，可以完全隐藏 IP。

有可能暴漏主机 IP 的方式：



隐藏 IP - CDN



隐藏 IP - CDN

Delivered-To: axiuno@gmail.com
Received: by 10.129.161.196 with SMTP id y187csp469436ywg: **received by gmail server**
Ved, 11 May 2016 19:46:26 -0700 (PDT)
X-Received: by 10.157.24.49 with SMTP id b46mr1055173ote.93.1463021186708;
Ved, 11 May 2016 19:46:26 -0700 (PDT)
Return-Path: <axiuno@sina.com>
Received: from smtp545-122.mail.sina.com.cn (mail3-166.sinamail.sina.com.cn. [202.108.3.166]) **send from sina mail server**
by mx.google.com with SMTP id qk8si371100ceb.26.2016.05.11.19.46.25
for <axiuno@gmail.com>;
Ved, 11 May 2016 19:46:26 -0700 (PDT)
Received-SPF: pass (google.com: domain of axiuno@sina.com designates 202.108.3.166 as permitted sender) client-ip=202.108.3.166;
Authentication-Results: mx.google.com;
spf=pass (google.com: domain of axiuno@sina.com designates 202.108.3.166 as permitted sender) smtp.mailfrom=axiuno@sina.com
Received: from unknown (HELO bbs.xiuno.com) ([)
by sina.com with ESMTP
12 May 2016 10:46:24 +0800 (CST) **leak the real web server IP**
X-Sender: axiuno@sina.com
X-Auth-ID: axiuno@sina.com
X-MAIL-UID: 2009525767625
Date: Thu, 12 May 2016 10:46:23 +0800
Return-Path: axiuno@sina.com
To: =?UTF-8?B?WG1lbn8g6L276K665Z2b?=<axiuno@gmail.com>
From: =?UTF-8?B?WG1lbn8g6L276K665Z2b?=<axiuno@sina.com>
Reply-To: =?UTF-8?B?WG1lbn8g6L276K665Z2b?=<axiuno@sina.com>
Subject: =?UTF-8?B?6Yn6K6+5a+G56CB6aqM6K+B56CB77yaMjlxMDc0IC0g44OQWG1lbn8g?=
=?UTF-8?B?6L276K665Z2b44CR?=
Message-ID: <c3e5ff7607299afc6038ae52fad47e0e@bbs.xiuno.com>
X-Priority: 3
X-Mailer: PHPMailer 5.2.1 (http://code.google.com/a/apache-extras.org/p/phpmailer/)
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="b1_c3e5ff7607299afc6038ae52fad47e0e"

隐藏 IP - CDN

- SMTP 发送邮件
- 本地化图片的业务
- 其他一些服务端主动发起的 TCP 请求都要严格检查
- 攻击 CDN 服务器（理论上，实际难度比较大）
- IP 段扫描，对比正文，找出所有 CDN 和 真实主机 IP

二、应用编码层的安全



2.1 两种典型的网络编程模型

单（少量）线程 + 异步

nodejs, nginx, lighttpd, redis,
业务相对简单的网游：如 QQ 棋牌

特点：无上下文切换，网络 IO 效率高，
但有异步回调深渊问题，不仅仅存在于 nodejs 中，
而且在 lua, c++ 中同样存在。实现复杂，适合
业务固定且较为简单的业务场景。

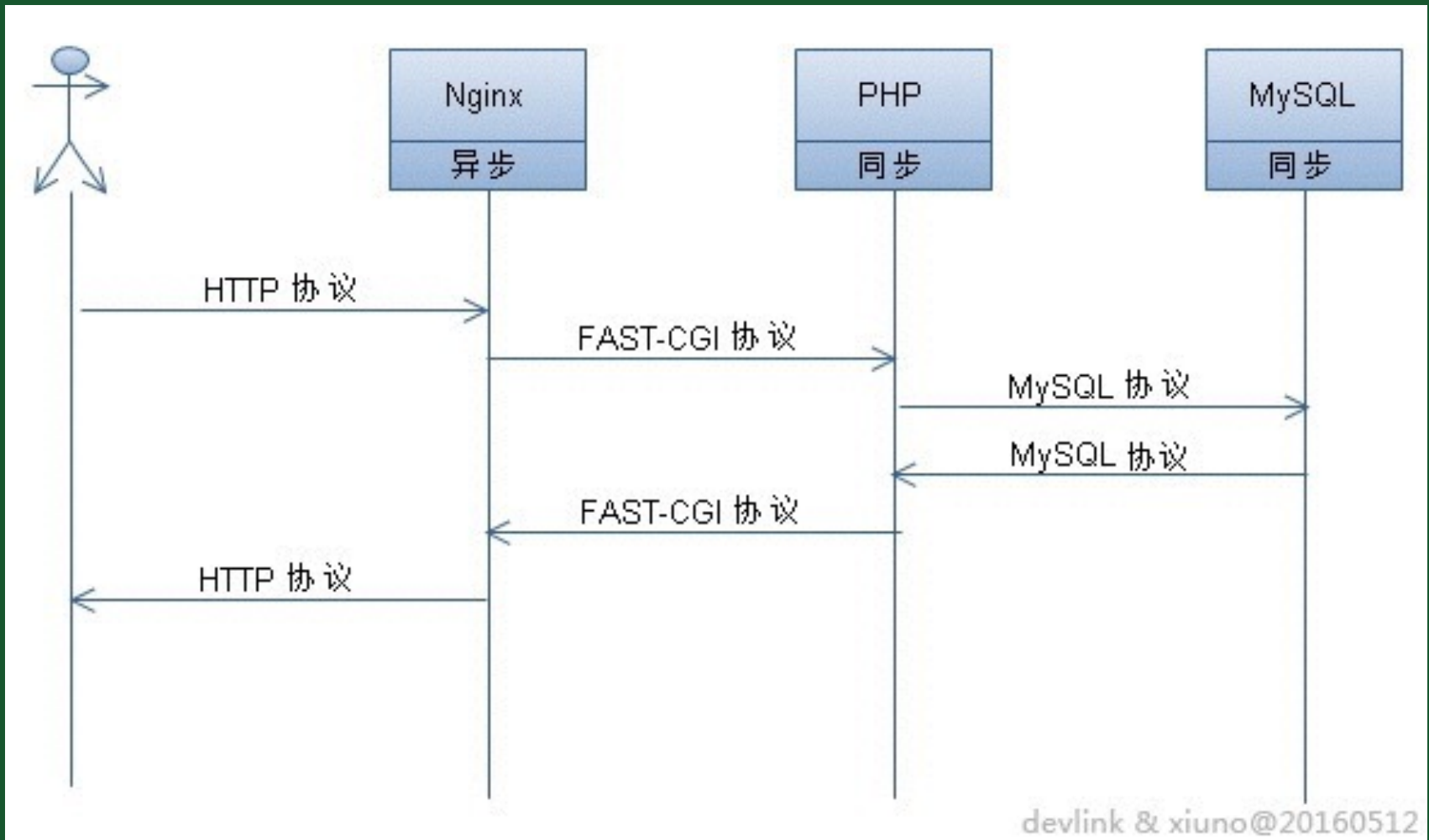
多线程 + 同步

apache, php-cgi, mysql, memcached,
业务复杂的网游：如网络创世纪 Sphere，
魔兽世界 Mangos

特点：上下文切换比较耗费资源，
不能处理大并发业务。
顺序执行，SLEEP 等待 网络 IO，写起来
比较顺手，适合复杂业务场景。

devlink & xiuno@20160512

2.2 典型的 Nginx+PHP+MySQL 架构



容易遭到 CC 攻击的部分，同步的部分

2.3 应对 CC 攻击

治标：屏蔽非正常请求，耗费资源的业务做 IP 和单位时间内请求次数限制

治本：尽量提高每秒完成请求数 (Qps)

五个方面：网络 IO、磁盘 IO、CPU 密集型运算、内存、进程数

网络 IO：fsocketopen() fopen() mysql_connect() memcached_connect(), curl() ...

磁盘 IO：error_log(), fileexists() is_file() is_dir() file_put_contents() ...

CPU 密集型运算：GD 库、加密解密函数、iconv() 转码

内存耗尽：GD 库，MySQL 大的结果集

进程耗尽：进程一直处于 SLEEP 或者繁忙都会导致进程耗尽，参照上面几条。

2.4 老生常谈：SQL 注射成因

```
<?php
```

```
$uid = $_GET['uid'];
```

可被外部控制

```
$link = mysql_connect('localhost', 'root', 'root');
```

```
mysql_select_db('test');
```

```
$query = mysql_query("SELECT * FROM `user` WHERE uid=$uid");
```

```
$arr = mysql_fetch_assoc($query);
```

```
print_r($arr);
```

```
?>
```

现在应该没人这么写了吧！

老生常谈：SQL 注射成因

```
<?php
```

```
$username = $_GET['username'];
```

```
$username = addslashes($username);
```

安全转义

```
$link = mysql_connect('localhost', 'root', 'root');
```

```
mysql_select_db('test');
```

```
$query = mysql_query("SELECT * FROM `user` WHERE username='$username'");
```

```
$arr = mysql_fetch_assoc($query);
```

```
print_r($arr);
```

```
?>
```

最起码也得这样！

2.5 SQL 注射 GBK 漏洞

现在还有不少开源产品和项目仍然在使用 GBK 编码，问题仍然潜在。

```
<?php
error_reporting(E_ALL);
$username = $_GET['username'];
$username = addslashes($username);
$link = mysql_connect('localhost', 'root', 'root');
mysql_select_db('test');
mysql_query('set names gbk');
$query = mysql_query("SELECT * FROM `user` WHERE username='$username'");
if(!$query) echo mysql_error();
$arr = mysql_fetch_assoc($query);
print_r($arr);
?>
```

`%e5%5c -> %e5%5c%5c`

`SELECT * FROM `user` WHERE username='錦'`

`← → ↻ /test/inject_gbk.php?username=%e5%5c`

`SELECT * FROM `user` WHERE username='錦'` You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''錦\'\' at line 1 Warning: mysql_fetch_assoc() expects parameter 1 to be resource, boolean given in /test/inject_gbk.php on line 35

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.5.37 |
+-----+
1 row in set
```


2.6 二次注射

```
<?php
$username = $_GET['username'];
$username = addslashes($username);
$link = mysql_connect('localhost', 'root', 'root');
mysql_select_db('test');
$query = mysql_query("INSERT INTO `user` set username='$username'");
if(!$query) echo mysql_error();
$id = mysql_insert_id();
$query = mysql_query("SELECT * FROM `user` WHERE uid='$id'");
$user = mysql_fetch_assoc($query);
$query = mysql_query("SELECT * FROM `user` WHERE username='$user[username]'");
if(!$query) {
    echo mysql_error();
} else {
    print_r(mysql_fetch_assoc($query));
}
?>
```

username='lisi\"

WHERE usname='lisi"

← → ↻ /inject_twice.php?username=lisi%27

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''lisi'' at line 1

2.7 不安全的单字节操作函数，如： substr()

```
<?php
error_reporting(E_ALL);
$username = $_GET['username'];
$username = addslashes($username);
$username = substr($username, 0, 32); // 安全截取 ?
$link = mysql_connect('localhost', 'root', 'root');
mysql_select_db('test');
mysql_query('set names gbk');
$query = mysql_query("SELECT * FROM `user` WHERE username='$username'");
if(!$query) echo mysql_error();
$arr = mysql_fetch_assoc($query);
print_r($arr);
?>
```

```
SELECT * FROM `user` WHERE
username='0123456789012345678901234567
891\
```

← → ↺ bbs.xiuno.com/test/inject_gbk.php?username=0123456789012345678901234567891%5c 📄 ☆ ☰

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''0123456789012345678901234567891\'\' at line 1 Warning: mysql_fetch_assoc() expects parameter 1 to be resource, boolean given in /home/wwwroot/bbs.xiuno.com/test/inject_gbk.php on line 21

如何规避这些编码不一致导致的 SQL 注射问题？

- 统一使用 UTF-8 编码。
- 字符串操作全部使用 mb_xxx() 。

推荐禁用如下函数：

一般在 shell 里使用，web 环境很少使用。

eval,passthru,exec,system,chroot,scandir,chgrp,chown,shell_exec,
proc_open,proc_get_status,ini_alter,ini_restore,
dl,pfsockopen,openlog,syslog,readlink,symlink,popepassthru,
stream_socket_server

尽量避免使用以下高危函数：

- `extract()`：不安全，有参数设置会覆盖当前变量，如果再加上 `register_global` 那很容易出问题，这也是 dz 早起版本被诟病的原因之一，当然以前大家都爱这么干，正如北方的老司机习惯在冰面的路上开车一样。
- `eval()`：方便，但是不安全，而且也影响性能。黑客倒是很喜欢。
- `preg_reaplace()` e 修饰符：PHP7 由 `preg_replace_callback()` 代替了

2.8 上传文件的文件名处理

```
<?php
// 文件后缀名, 不包含 .
function file_ext($filename) {
    return strtolower(substr(strrchr($filename, '.'), 1));
}

$tmpfile = $_FILES['upfile']['tmp_name'];
$filename = $_FILES['upfile']['name'];
$ext = file_ext($filename);

// 黑名单过滤不安全
if(in_array($ext, array('php', 'asp', 'aspx', 'jsp'))) {
    exit('Not allowed!');
}
// 严格过滤 $filename
move_uploaded_file($tmpfile, './upload/'.$filename);
?>
```

应该采用白名单,

写出这种代码基本就 GAME OVER 了, 各种死法。

123.php%00.jpg

123.asa IIS 默认支持的后缀

../../static/common.js

2.9 危险的 include \$var;

```
<?php
$file = $_GET['file'];
include "./include/$file.php";
?>
```

Warning: include(): Failed opening './include/../../../../../../../../etc/passwd' for inclusion (include_path='.:') in /test/include.php on line 6

php version < 5.3.4

2.10 日志文件存放格式

有时候日志文件防止别人读，会以 xxx.php 存放，最第一行写入 `<?php exit; ?>` 这种安全么？

如果文件因为 IO 出错或者管理员操作，导致文件清空，后面再写入的数据中如果包含有 `<?php xxx ?>` 那么就可能会被执行。在 discuz 某个版本真实发生过。

2.11 不要信任 GPC，也不要信任 S

一般情况，我们会对 GPC 进行过滤，对 S 会相对信任些。但是也不可以完全信任。只要是用户能控制的，都不可信任。

```
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
```

```
mysql_query("INSERT TABLE `user` SET regip='$ip'");
```

```
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];  
$arr = array_filter(explode(',', $ip));  
$ip = end($arr);  
$ip = long2ip(ip2long($ip));
```


2.12 错误信息泄露和获取

构造畸形数据，使 PHP 报错，获取错误信息

```
<?php
error_reporting(E_ALL);

$s = isset($_GET['s']) ? $_GET['s'] : '';
$s = substr($s, 0, 32);

echo $s;

?>
```

← → ↻  [redacted]/test/leak_path.php?s[]=aaa

Warning: substr() expects parameter 1 to be string, array given in
/home/wwwroot/[redacted]/test/leak_path.php on line 6

2.13 多种语言语法搞混，比如 PHP JS C

多种语言有时候出现搞混的情况，下面为一个例子：

```
<?php
$a = 'a';
if($a > 0) {
    echo 'ok';
} else {
    echo 'error';
}
?>
```

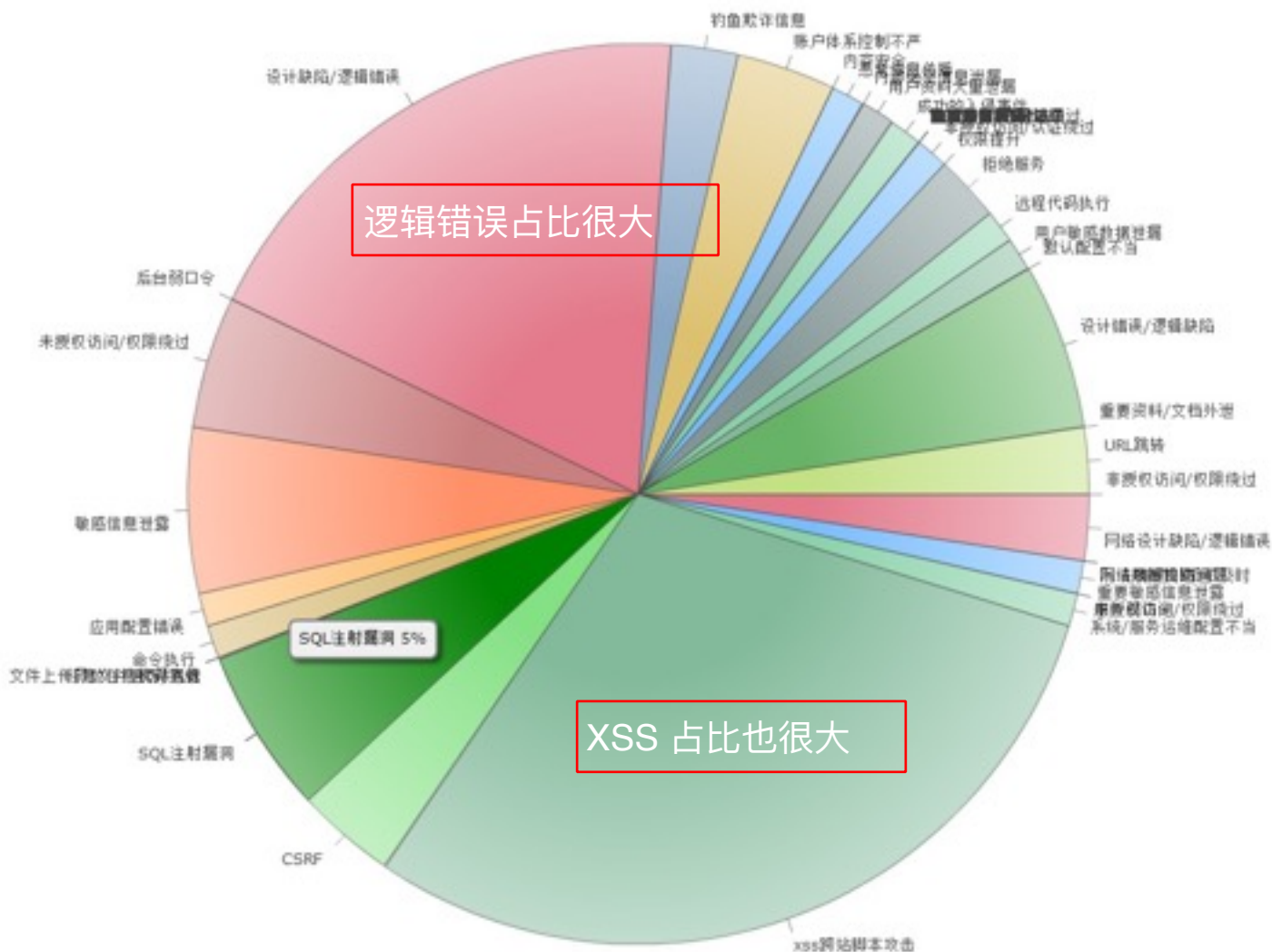
在 C 语言当中，'a' 是 0x61，是 > 0 的，在 PHP 里面会被 intval() 转换，结果就是 == 0 了。

另外比较常见的，就是对空的判断。

各类型漏洞统计

腾讯漏洞类型统计

数据来自乌云 2016/5/11



2.14 低级的普遍存在的逻辑错误

目前已经成为主要的问题，看似低级，但是很难避免，往往影响很大。

下面这个例子来自 wooyun 网白帽子 felixk3y 对用友的一段代码审计：

```
1  if($operateId == 1){
2      $date = date("Ymd");
3      $dest = $CONFIG->basePath."data/files/".$date."/";
4      $COMMON->createDir($dest);
5      //if (lis_dir($dest)) mkdir($dest, 0777);
6      $nameExt = strtolower($COMMON->getFileExtName($FILES['Filedata']['name']));
7      $allowedType = array('jpg', 'gif', 'bmp', 'png', 'jpeg');
8      if(!in_array($nameExt, $allowedType)){
9          $msg = 0;
10     }
11     if(empty($msg)){
12         $filename = getmicrotime().'.'.$nameExt;
13         $file_url = urlencode($CONFIG->baseUrl."data/files/".$date."/".$filename);
14         $filename = $dest.$filename;
15         if(empty($FILES['Filedata']['error'])){
16             move_uploaded_file($FILES['Filedata']['tmp_name'],$filename);
17         }
18         if (file_exists($filename)){
19             //$msg = 1;
20             $msg = $file_url;
21             @chmod($filename, 0444);
22         }else{
23             $msg = 0;
24         }
25     }
26     $outMsg = "fileUrl=".$msg;
27     $_SESSION["eoutmsg"] = $outMsg;
28     exit;
29 }
```

用友ICC网站客服系统远程代码执行漏洞受影响的厂商

https://**.**.**.*/5107/upload/uploadFlash.php 银联

http://**.**.**.*/5107/upload/uploadFlash.php 迅雷

http://**.**.**.*/5107/upload/uploadFlash.php 太平洋保险

http://**.**.**.*/5107/upload/uploadFlash.php 人寿保险

http://**.**.**.*/5107/upload/uploadFlash.php 海南航空

http://**.**.**.*/5107/upload/uploadFlash.php 网维大师

http://**.**.**.*/5107/upload/uploadFlash.php 海尔电器

http://**.**.**.*/5107/upload/uploadFlash.php 金山毒霸

http://**.**.**.*/5107/upload/uploadFlash.php 盛大在线

http://**.**.**.*/5107/upload/uploadFlash.php 大成基金

http://**.**.**.*/5107/upload/uploadFlash.php 上海农商银行

◦ ◦ ◦ ◦

低版本曾经存在的问题

高版本已经解决的问题不再阐述和分析。

Web Server 后缀解析问题：

Apache 低版本会将 123.php.xxx 解析为 php

nginx 低版本会将 123.jpg/123.php 解析为 php

PHP 低版本相关：

get_magic_quote_gpc() 5.4.0 始终返回 FALSE，已经移除

register_global PHP 5.4 开启后很不安全，已经移除

2.15 XSS 的成因

看起来很简单，没有转义用户输入的 HTML 标签 < > “ & ，直接跟模板一起输出。

```

```

如果用户输入 1.jpg“ onload=alert(123)

那么结果就成了：

```

```

```
$avatar_url = htmlspecialchars($_GET['avatar_url']);
```

```
mysql_query("INSERT INTO xxx SET avatar_url='$avatar_url'");
```

XSS 可以做什么？

如果管理员访问了这个 URL，可以获取管理员的 cookie，并且模拟管理员提交 GET POST 请求，非常危险，并不是小问题，不可忽略。

为了避免 cookie 被盗用，可以在服务端设置的时候，设置为 HTTPONLY，高版本浏览器基本都支持了。但仍然无法阻止模拟管理员提交 GET POST 请求，所以我们可以认为被 XSS 就是等于拿到了部分网站管理员操作权限。



如果我们需要富文本怎么办？ 比如 HTML 编辑器。

不要试图用正则去过滤 `<script>` 等标签，还有 `onload onerror`，XSS 有一万种绕过的方式。具体的方式可以去搜索，

服务端采用 HTML 严格的语法解析，对 `tagname attrname attrvalue, css` 才能彻底解决这个问题。

有一个开源的基于 HTML 语法的类库：XML_HTMLSax3，我对它进行了一些改造，兼容了 PHP7，设置好了常用的白名单和正则表达式，在 Xiuno BBS 上目前工作正常，性能也不错。

https://github.com/xiuno/xiunobbs/blob/master/xiunophp/xn_html_safe.func.php

HTML CSS 白名单设置

```

1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 181
```

2.16 基于白名单的 XSS 过滤

```
1715 $s = '  
1716     <script>alert(/alert(123)/)</script>  
1717     <b onclick="ddd">abcc</b>  
1718     <table class="abc" style="width: 103330px;     expre/*xxx*/ssion:(alert(123)); background: url(1.jpg) no-repeat ;">  
1719         <tr><td>内容</td></tr>  
1720     </table>';  
1721  
1722 echo xn_html_safe($s);  
1723  
1724 /* 结果:  
1725  
1726 <b>abcc</b>  
1727 <table class="abc" style="width:100%px; background:url(1.jpg) no-repeat ;">  
1728     <tr><td>内容</td></tr>  
1729 </table>  
1730  
1731 */
```

基于正则表达式很难过滤掉各种奇奇怪怪的语法

三、业务逻辑层&社工



3.1 数据的读写权限的正确判断

```
if($action == 'update') {  
    $pid = intval($_GET['pid']);  
    if($method == 'GET') {  
        $post = post_read($pid);  
        if($post['uid'] != $uid) exit('您无权编辑别人的帖子');  
        include './pc/view/post_update.htm';  
    } elseif($method == 'POST') {  
        $message = $_POST['message'];  
        post_update($pid, array('message'=>$message));  
        echo '更新完毕';  
    }  
}
```

看似低级，但新手容易犯这种错误，目前该类型的漏洞类型已经成为漏洞生力军。

3.2 社会工程

黑客感兴趣的密码：

网站后台密码、管理员邮箱、FTP 密码、SSH 密码、QQ 密码、域名管理密码、手机密码等等。

收集个人信息渠道：搜索引擎、博客、SNS，QQ空间，朋友圈等。

周围的人：你周围的朋友和同事可能会泄露你的信息，往往你周围的人安全意识不高。

伪装成潜在客户：与你和你周围的人聊天，套取信息。

图片隐私泄露：图片中包含大量的信息，目前难搜索，不代表未来不能搜索。

其他站点被脱裤导致的威胁：比如购物网站，邮件服务商，移动服务商。

社工的方法是无限多种可能，防不胜防。

3.3 密码排列组合

张小军 ZhangXiaoJun 19901210 18612345678

常见的组合：

zxj1990

zxj19901210

zxj1210

Zxj1990

zhangxiaojun1990

zhangxiaojun1210

zhang1990

zhang18612345678

12345678zhang

zxj18612345678

zxj@1990

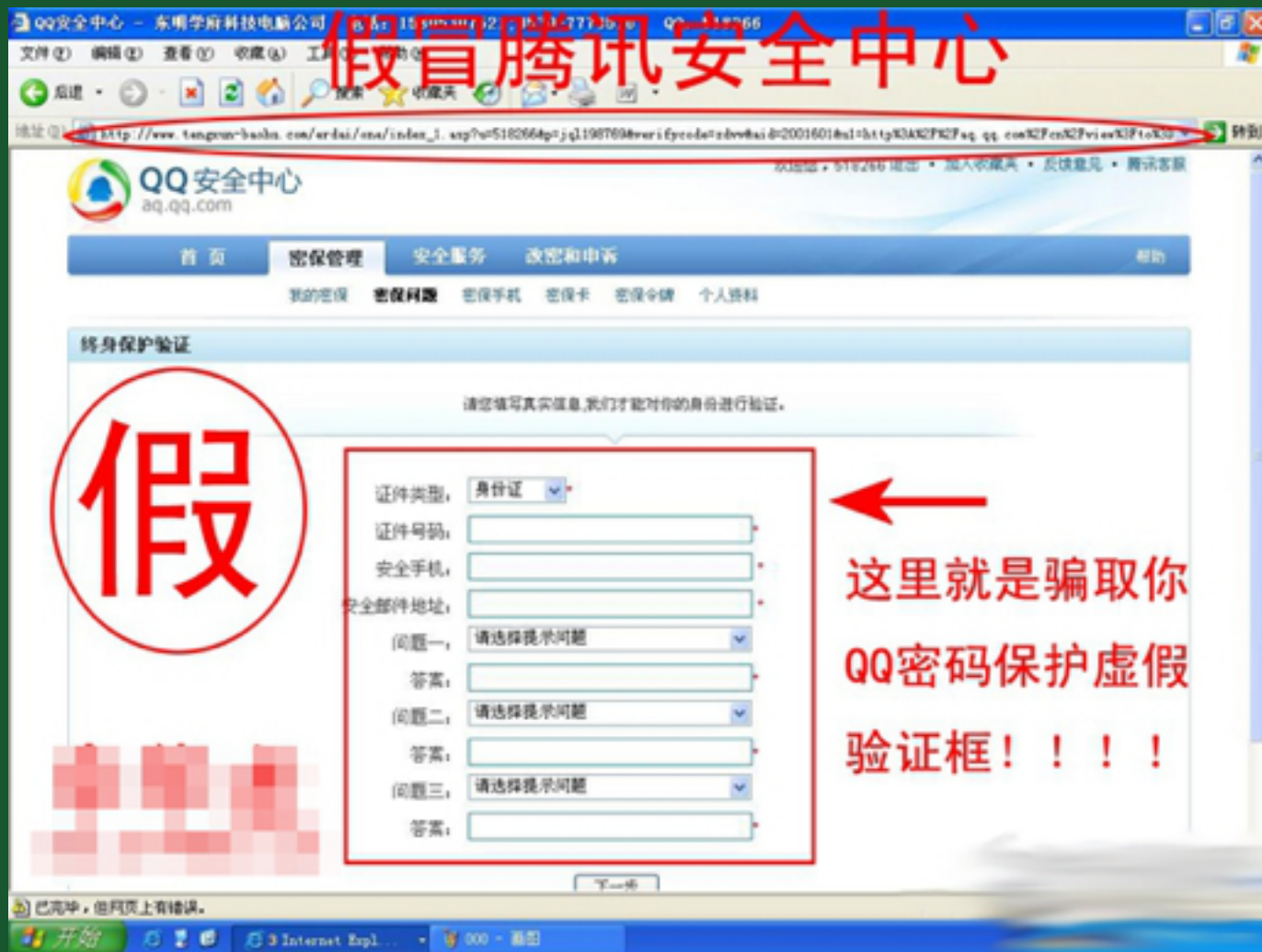
zxj@1234

zxj@123456

zxj12345678

12345678zxj

3.4 网址诈骗



你可以对此嗤之以鼻，但是你父母还有其他的的朋友呢？



谢谢大家