Magical Shellcode

AssassinQ

Outline

- What is shellcode
- Know about x86 linux system call
- How to write & test your own shellcode
- Basic shellcode tricks
- ~~Advanced shellcode tricks~~

What is shellcode

- 类似于这样的一长串字符，可以翻译成汇编代码
  - "\x6a\x68\x68\x2f\x2f\x2f....."
- 指的是一串可以被直接执行的字符串
- 为什么会被叫shellcode?
  - 因为我们常常通过执行它来拿到shell

What is shellcode

- 假设攻击者可以执行数据段的代码
- 控制任意数据段的内容 → 任意执行想要的代码
- 很多攻击者会先尝试用其他方法，比如说调用 mprotect() 函数来更改数据段的权限为可执行

Know about x86 linux system call

- x86 linux system call table
- 要调用 system call，必须:
  - 在每个寄存器中放入对应的值（参数，系统调用号）
  - 执行 int 0x80（x86）来调用 system call（x64 要用 syscall 指令）

sys_execve

- int execve(const char *filename, char *const argv[], char *const envp[]);
  - filename (for ebx) : 要执行的程序的路径
  - argv[] (for ecx) : 程序的参数
  - envp[] (for edx) : 环境变量
- 目标 : 执行 execve(“/bin/sh”, NULL, NULL)
- 这里 argv[] 和 envp[] 直接清成 0 (NULL byte)

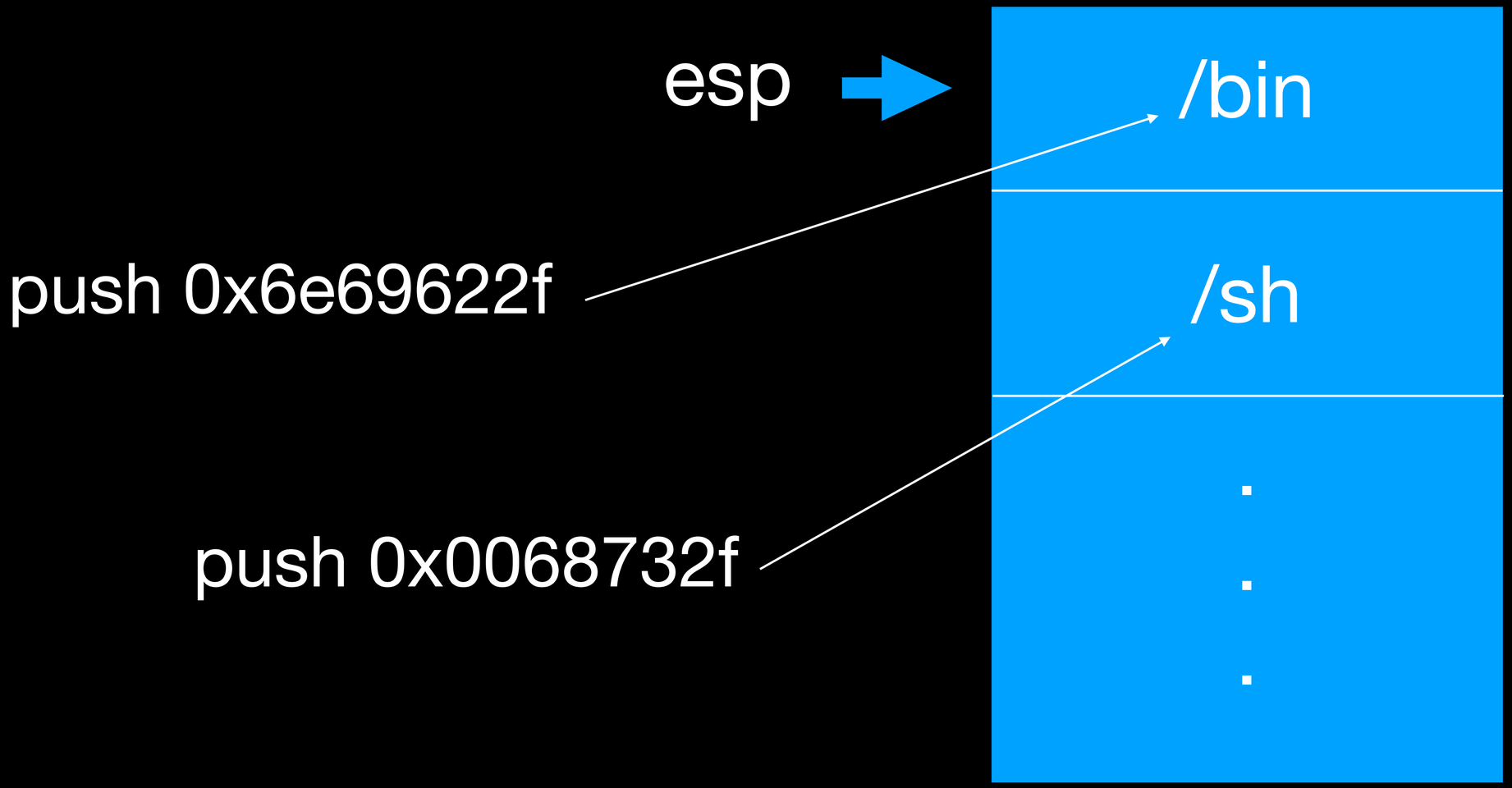
How to write your own shellcode

- 机器码是由汇编转换而来的
- 因此要写一段 shellcode，必须要会用汇编语言
- 工具
  - nasm : assembler (生成 object file)
  - ld : GNU linker (生成 executable)
  - xxd : 方便觀察 shellcode

How to write your own shellcode

- push 0x0068732f
  - 0x0068732f 即字符串 "/sh" (null byte 结尾)
  - 同理 0x6e69622f 即字符串 "/bin"
  - 此时，esp 指向字符串 "/bin/sh"
- mov ebx, esp
  - 把 esp 存放的地址 (指向 /bin/sh) 放入 ebx

How to write your own shellcode



How to test your own shellcode

- "./shell.bin" 执行 shellcode，成功的话就能拿到 shell
- 使用 gdb 来 debug
  - gdb ./shell.bin
  - disas _start，查询具体的代码段

Basic shellcode tricks

- Null free
  - strcpy()， sprintf() …等函数遇到 null byte 的时候会被截断
  - 尽量避免避免含有 null byte 的 shellcode，还可以减短长度

Basic shellcode tricks

- 方法
  - mov eax, 0x0 → xor eax, eax
  - mov eax, 0xb → mov al, 0xb
  - shr eax, 0x8 → set eax = 0x00xxxxxx

Basic shellcode tricks

- int open(const char *pathname, int flags)
  - *pathname (for ebx) : 文件路径
  - flags (for ecx) : 模式 (ex. 只读)

Basic shellcode tricks

- ssize_t read(int fd, void *buf, size_t count)
  - fd (for ebx) : 文件描述符
  - *buf (for ecx) : buffer 地址
  - count (for edx) : 最多读几个byte

Basic shellcode tricks

- ssize_t write(int fd, const void *buf, size_t count)
  - 參数意义基本上跟 read() 一樣

References

- Linux Syscall Reference
- Shellcode — bruce30262
- Shellcode Tricks