GNU/Linux- 网络监听

网络监听的使用

GNU/Linux-Sniffer

tcpdump

GNU/Linux-Sniffer

tcpdump

tcpdump 是 Linux/UNIX 常用的抓包 / 监听工具

tcpdump 用法

#tcpdump [-AennqX] [-i 网络适配器] [-w 数据写入的文件名] [-c 次数] [-r 档案] [打算所要抓取的数据包格式]

GNU/Linux-Sniffer

tcpdump

参数说明

参数	说明
-A	数据包内容以 ASCII 码显示
-e	抓取 MAC 数据包 (即抓取数据链路层数据包)
-nn	直接以 IP 地址及端口号显示
-q	仅列出较为简短的数据包
-X	可以列出 16 进制数据及 ASCII 数据包的内容
-i	指定网络适配器 (eth0,ppp0 等)
-w	将监听的数据写入指定文件中
-r	读取指定文件中的数据
-c	监听数据包，如不指定次数则 tcpdump 将一直监听，直至用户手工中断

GNU/Linux-Sniffer

tcpdump
参数说明

参数	说明
-v	输出一个比较详细的信息
-vv	输出一个详细的连接信息
-c	收到指定数据包的数据后，停止 tcpdump 的运行
-a	将网络地址和广播地址变为名字
-d	将匹配的数据包以汇编格式输出
-dd	将匹配的数据包以 C 语言格式输出
-ddd	将匹配的数据包以十进制格式输出
-f	将外部 Internet 地址以数字形式输出

GNU/Linux-Sniffer

tcpdump

例：

本地

#tcpdump –i eth1 –nn

#tcpdump –i eth0 –nn port 80

#tcpdump –i eth0 –nn –X ‘prot 80’

GNU/Linux-Sniffer

tcpdump

例：

远程

截获 192.168.1.1 数据包

```
#tcpdump host 192.168.1.1
```

截获 192.168.1.1 和 192.168.1.2 数据包

```
#tcpdump host 192.168.1.1 and \
(192.168.1.2\)
```

GNU/Linux-Sniffer

tcpdump

例：

远程

截获 192.168.1.1 和除了 192.168.1.2 数据包

#tcpdump host 192.168.1.1 and ! 192.168.1.2

获取主机 192.168.1.1 发送及接收的 www 数据包

#tcpdump tcp port 80 host 192.168.1.1

GNU/Linux-Sniffer

tcpdump

例：

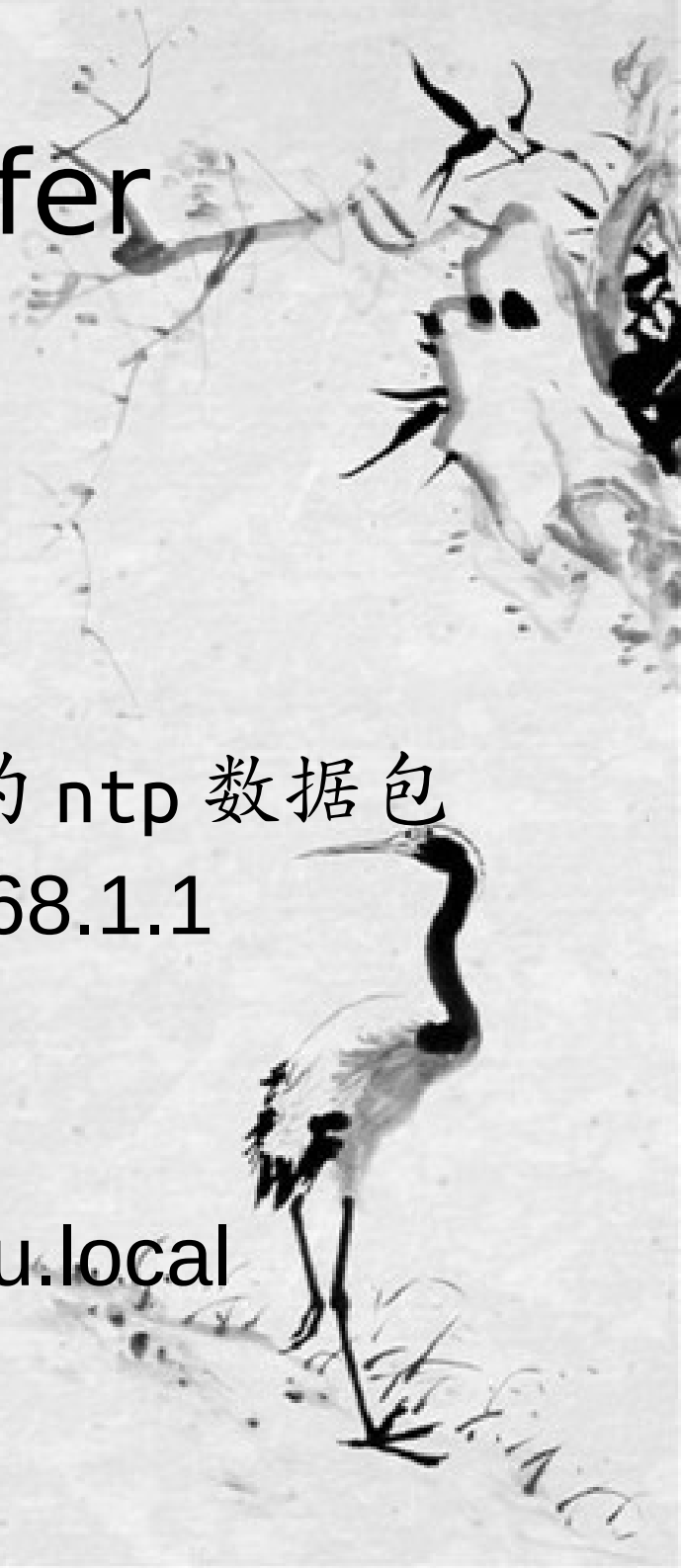
远程

获取主机 192.168.1.1 发送及接收的 ntp 数据包

#tcpdump udp port 123 host 192.168.1.1

监视 www.niliu.local 发出的数据

#tcpdump –i eth1 src host www.niliu.local



GNU/Linux-Sniffer

tcpdump

例：

远程

监视 www.niliu.local 接收的数据

#tcpdump –i eth1 dst host www.niliu.local

监控 ping/pong

```
#tcpdump -i any -n  -v \ 'icmp[icmptype] =
icmp-echoreply or icmp[icmptype] = icmp-
echo'
```

GNU/Linux-Sniffer

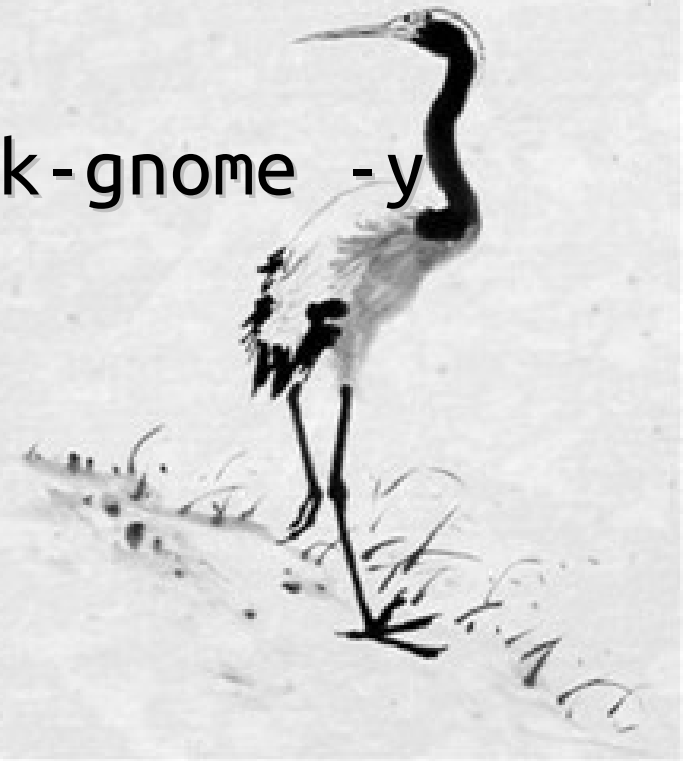
wireshark

GNU/Linux-Sniffer

wireshark 属于 GUI 模式的监听 / 抓包工具

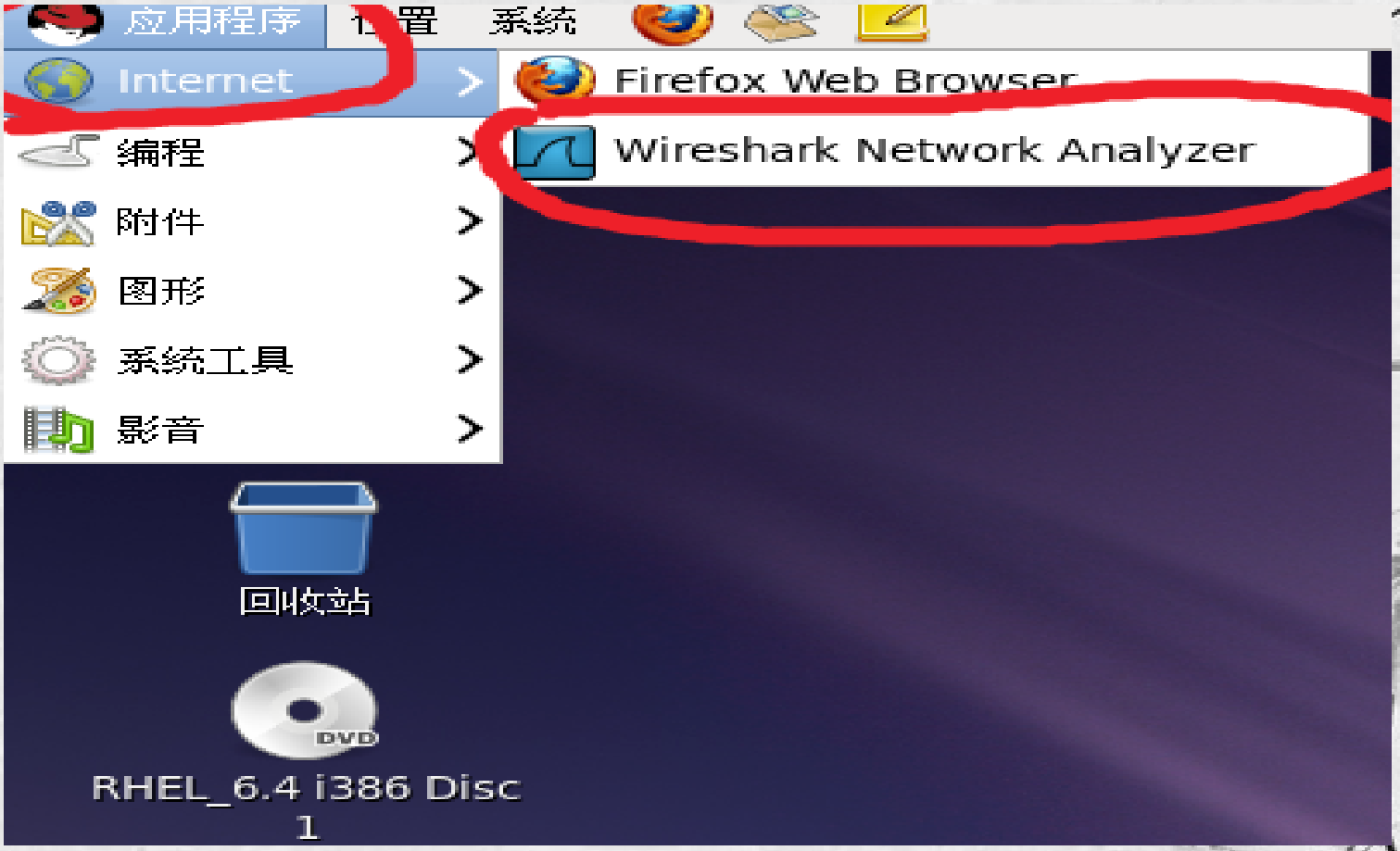
安装 Wireshark

```
#yum install wireshark wireshark-gnome -y
```



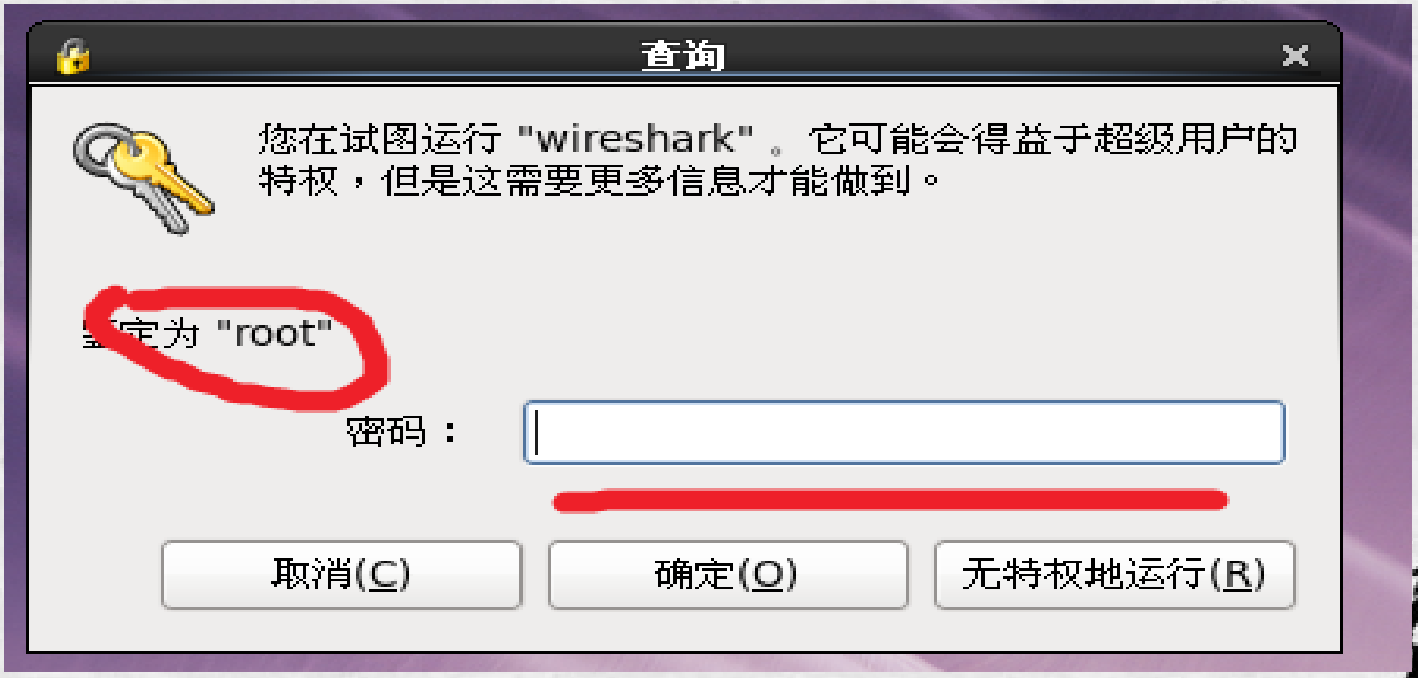
GNU/Linux-Sniffer

启动 wireshark



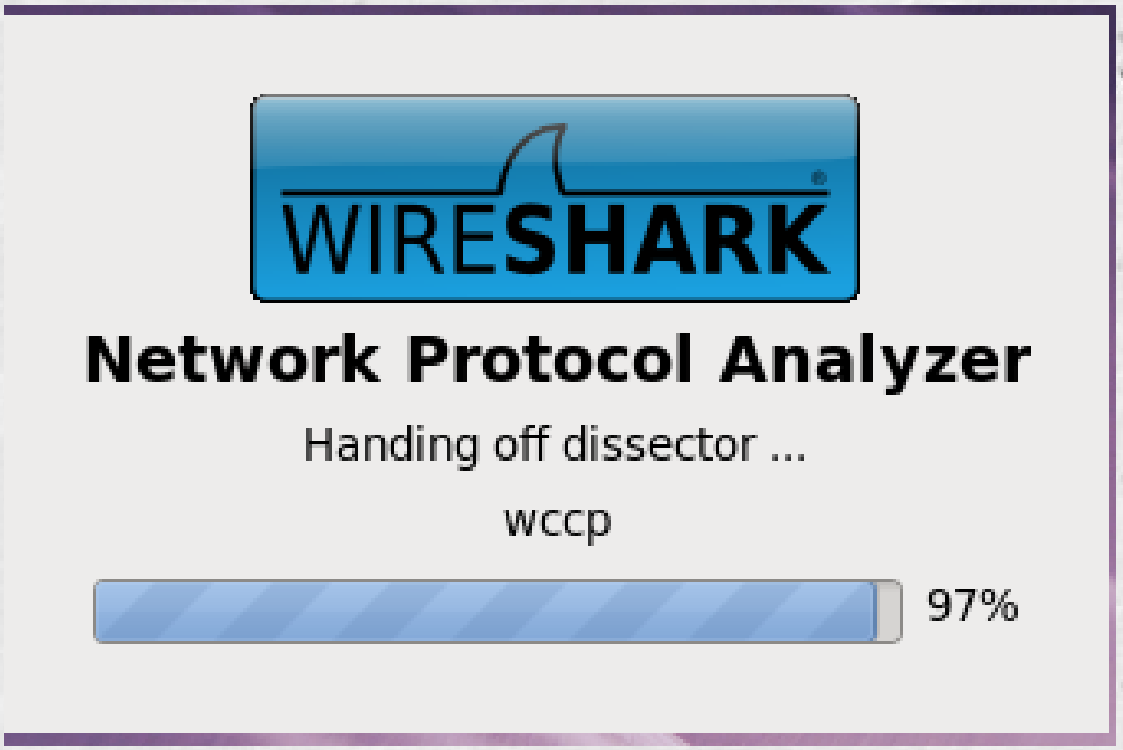
GNU/Linux-Sniffer

启动 wireshark



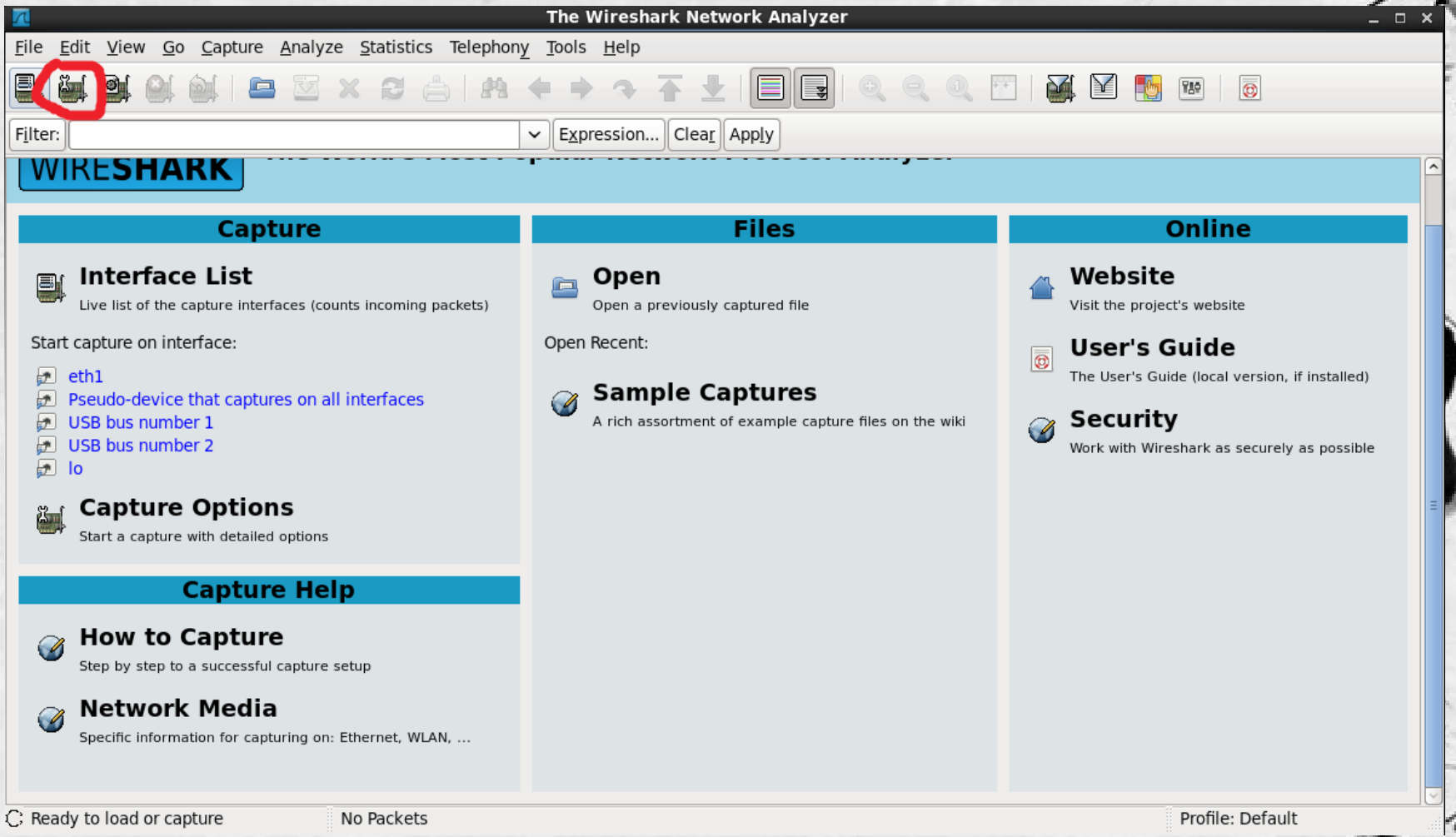
GNU/Linux-Sniffer

启动 wireshark



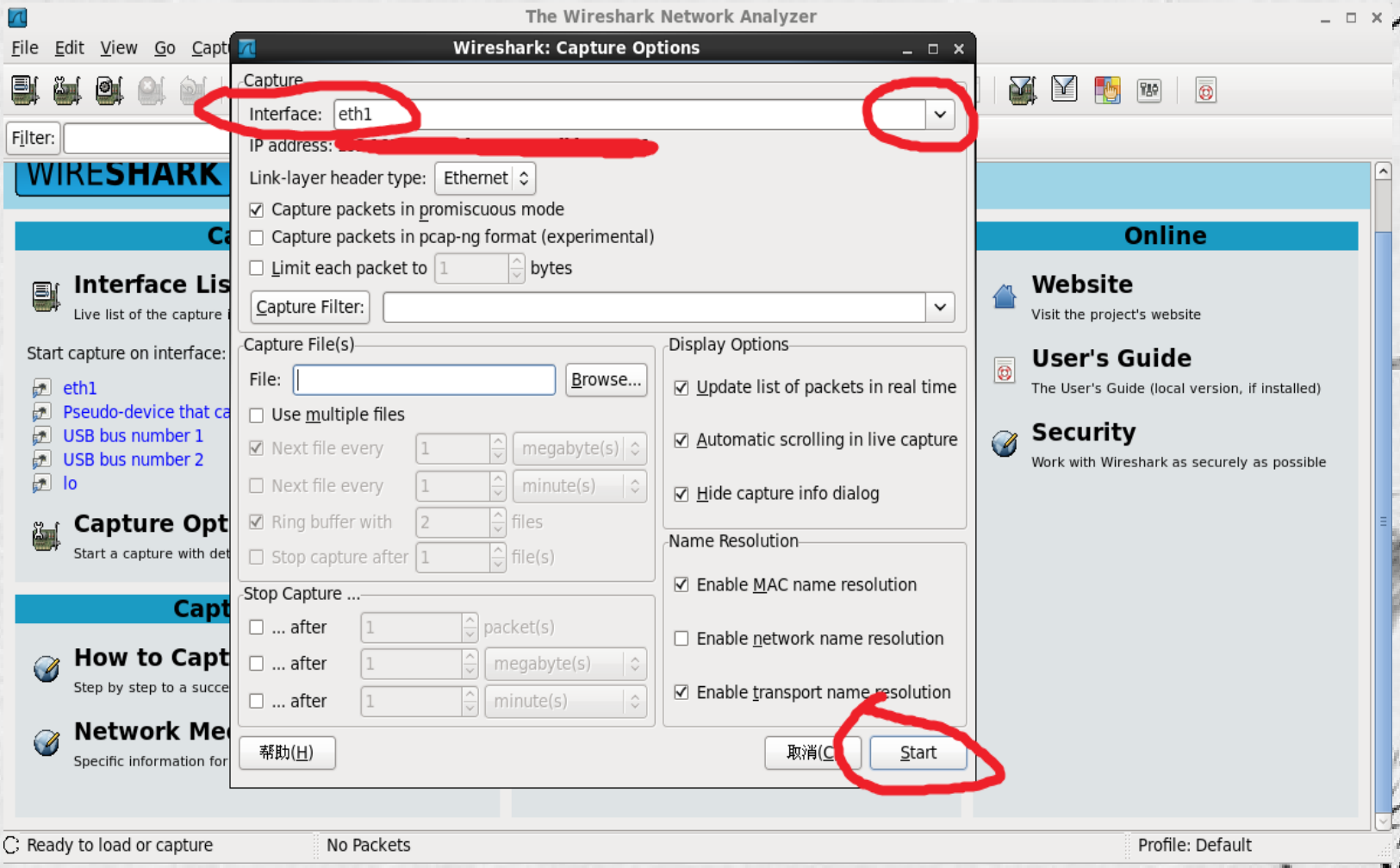
GNU/Linux-Sniffer

启动 wireshark



GNU/Linux-Sniffer

启动 wireshark



GNU/Linux-Sniffer

启动 wireshark