难度等级：入门级

案例分析挑战内容：

这次案例分析挑战是完全为刚入门的安全分析师准备的，目标是分析由南佛罗里达蜜网项目组成员人为构造的从因特网到一台蜜罐主机的 5 次不同类型端口扫描。需要指出的是，这次案例分析中的端口扫描流量并不是从“野外”捕获的，而是特意构造的，这次入门级的案例分析挑战的目的完全是为了提供学习和训练的机会。

网络入侵检测器-snort 捕获了每次扫描的流量并存入了一个 tcpdump 格式二进制网络日志文件中。这次挑战的任务就是分析这个文件，并回答所列的问题。通过这次挑战，你能够学习到数据包抓取技术的使用方法，以及使用数据包解码工具 tcpdump 或 ethereal(现已改名为 Wireshark)分析网络数据包的技能。

问题：

1. 什么是二进制网络日志文件？这种文件是如何生成的？
2. 什么是 MD5？它提供什么作用？
3. 攻击主机的 IP 地址是什么？
4. 网络扫描的目标 IP 地址是什么？
5. 本次案例中是使用了哪个扫描工具发起这些端口扫描？你是如何确定的？
6. 什么是端口扫描，端口扫描的目的是什么？
7. 本次案例中使用了 5 种不同类型的扫描方式，请将这 5 种方法标识出来，并描述其工作原理。
8. 在蜜罐主机上哪些端口被发现是开放的？
9. 额外奖励问题：攻击主机的操作系统是什么？

待分析二进制文件位置：ftp://222.29.112.10/data/sotm23.tar.gz。

课程 1 提示：

- **建议阅读**
  - **《黑客大曝光》 第二章 扫描**
  - **如无计算机网络知识基础，阅读**
    - **《计算机网络》第 1 章，第 5.6 节，第 6.5 节**
    - **《TCP/IP 详解》协议卷第 1, 3, 6, 7, 17, 18 章**
- **使用工具**
  - **wireshark(ethereal)为主**
  - **snort 可为辅－检测扫描工具的特征**
  - **结合使用命令行与图形界面：高手倾向于使用命令行，但不意味着完全排斥 GUI**
- **分析关键点**
  - **确认发起扫描使用的工具，然后对照它的使用说明和你看到的数据包特征**
  - **快速区分出各次扫描的范围边界（如何区分？注意观察每次扫描开始时刻具有什么特征）－不要累坏你的眼睛**
  - **善用 tcpdump filter 选择性的关注数据包**

分析过程和问题解答：

0. 准备工作

使用"wget"指令下载待分析的二进制记录文件压缩包：

wget http://www.honeynet.org/scans/scan23/sotm23.tar.gz

使用"md5sum"指令验证文件的完整性：

md5sum sotm23.tar.gz

9d28c5ee9ce7b77e3099a07ad303811f sotm23.tar.gz

查看输出结果中的 MD5 值与公布的 MD5 值是否一致，如不一致，说明文件完整性遭破坏。

使用"tar"指令解压缩下载的压缩文件包，并检查解压后的二进制记录文件的完整性

tar zxf sotm23.tar.gz

ll

-rw------- 1 root root 11998444 Aug 27 2002 0826@19-snort.log

-rw-r--r-- 1 root root 52 Aug 27 2002 0826@19-snort.log.md5

md5sum 0826@19-snort.log

0ce142f18c23d9ab00f992a57ad097d4 0826@19-snort.log

cat 0826@19-snort.log.md5

0ce142f18c23d9ab00f992a57ad097d4 0826@19-snort.log

至此，我们已经验证了待分析文件的完整性，现在可以开始面对挑战。

防御策略: *在下载软件或其他文件后，对其进行完整性验证，即核对下载文件的 MD5 值和该文件在权威发布站点上的 MD5 值是否一致，可以避免遭受特洛伊木马的威胁，建议读者能够培养并保持这样的好习惯。*

历史事件: *GMT 时间 2002 年 11 月 11 日 10 点 14 分，tcpdump 工具官方网站 www.tcpdump.org 上提供下载的 tcpdump 和 libpcap 源码包中被黑客植入特洛伊木马，木马程序会在 tcpdump 编译时自动执行，并向攻击者提供远程 Shell。被植入木马的 tcpdump 源码包的 MD5 值为 md5sum 3c410d8434e63fb3931fe77328e4dd88 tcpdump-3.7.1.tar.gz，而官方发布的 tcpdump 源码包 MD5 值为 md5sum 03e5eac68c65b7e6ce8da03b0b0b225e tcpdump-3.7.1.tar.gz。到 11 月 13 日这些包含特洛伊木马的源码包才被 tcpdump 开发团队移除，并由 CERT/CC 发布了 CA-2002-30 安全警告。*

1. 什么是二进制网络日志文件？这种文件是如何生成的？

二进制网络日志文件是对'0'和'1'组成的网络二进制数据进行捕获和记录的文件格式，二进制网络日志通常被称为原始流量捕获，因为我们以流量的最原始存在形式（'0'和'1'）捕获和记录数据包内容。网络流量可以在混杂模式的网络接口、共享式集线器、TAP 分路器、路由器或者交换机的 SPAN 镜像端口进行二进制捕获。在类 Unix 平台上，从网络上捕获流量的底层功能模块是 Libpcap 网络数据包捕获库，作为一个标准库，Libpcap 被 tcpdump、snort、wireshark(ethereal)等众多工具所使用。Libpcap 库的 Windows 平台版本称为 Winpcap，winpcap 使得我们可以在 Windows 环境下运行 snort、wireshark(ethereal)等工具。

使用 snort 工具生成二进制记录文件的指令格式如下：

snort -l /var/log/snort -b

读取二进制记录文件的指令格式为：

snort -vde -r snort.log

使用 tcpdump 工具生成和读取二进制记录文件的指令格式分别为：

tcpdump -w log

tcpdump -r log

使用 wireshark(ethereal)工具生成和读取二进制记录文件的指令格式分别为：

tshark(tethereal) -w log

tshark(tethereal) -r log

你也可以使用 wireshark(ethereal)工具的图形界面生成和读取二进制记录文件。

工具简介：

- *tcpdump (http://www.tcpdump.org/)是类UNIX 平台下经典的网络数据包抓取和分析工具。*
- *snort (http://www.snort.org/)是最著名的开源网络入侵检测系统，由Martin Roesch 从1998 年开始开发，目前已发展到 2.6 版本，成为入侵检测领域的事实标准。Snort 工具也具有基本的网络数据包抓取功能。*
- *wireshark（最流行的开源网络数据包分析工具）：原名 ethereal(http://www.ethereal.com/)，由Gerald Combs 等人开发，后由于2006 年8 月Gerald Combs 离开NIS 公司加入CaceTech 公司，没有与NIS 公司(ethereal 商标的所有者)达成协议，因此将ethereal 改名为wireshark http://www.wireshark.org/。*

2. 什么是 MD5？它提供什么作用？

从 RFC1321 引用：MD5 算法以任意长度的消息作为输入，产生输入消息的一个 128 bit 的"指纹"或称"消息摘要"。根据猜想，产生拥有同样摘要的两个不同消息在计算上是不可行的，同样产生一个具有预先给出摘要的消息在计算上也是不可行的，MD5 算法主要应用于数字签名。MD5 算法的作用是为保证消息、数据、文件的完整性提供了一个安全的单向哈希函数。在类 Unix 平台下提供 MD5 算法计算的指令是"md5sum"。

科研进展：

在 2004 年 8 月 17 日美国加州圣巴巴拉召开的国际密码学会议(Crypto'2004)上，来自中国的王小云教授做了破解 MD5、HAVAL-128、MD4 和 RIPEMD 这四个著名的 hash 算法的报告，在王小云教授仅公布到他们的第三个惊人成果的时候，会场上已经是掌声四起，报告不得不一度中断。报告结束后，所有与会专家对他们的突出工作报以长时的热烈掌声，有些学者甚至起立鼓掌以示他们的祝贺和敬佩。当人们掌声渐息，来学嘉教授又对文章进行了一点颇有趣味的补充说明。由于版本问题，作者在提交会议论文时使用的一组常数和先行标准不同，在会议发现这一问题之后，王小云教授立即改变了那个常数，在很短的时间内就完成了新的数据分析，这段有惊无险的小插曲倒更加证明了他们论文的信服力，攻击方法的有效性，反而凸显了研究工作的成功。会议结束时，很多专家围拢到王小云教授身边，既有简短的探讨，又有由衷的祝贺，褒誉之词不绝。包含公钥密码的主要创始人 R. L. Rivest 和 A. Shamir 在内的世界顶级的密码学专家也上前表示他们的欣喜和祝贺。会议总结报告这样写道："我们该怎么办？MD5 被重创了；它即将从应用中淘汰。SHA-1 仍然活着，但也见到了它的末日。现在就得开始更换 SHA-1 了。"

然而 SHA-1 的好景也不长，在 2005 年的国际密码学会议(Crypto'2005)上，王小云教授又给出了 SHA-1 算法的碰撞产生方法，只需要 2^63 次 hash 操作就可以找出碰撞，这个结果比她提交论文中给出的 2^69 次操作又提高了 2^6 倍，也大大小于枚举破解需要的 2^80 次操作。该论文被评为了 Crypto'2005 的优秀论文。另外王小云教授还给了 2^39 次操作找

出 SHA-0 算法碰撞的方法。

至此，目前在世界上应用最广泛的两大 hash 算法 MD5 和 SHA-1 均被破解。《崩溃！密码学的危机》，美国《新科学家》杂志用这样惊耸的标题概括王小云教授里程碑式的成就。由于王小云教授的出现，美国国家标准与技术研究院宣布，美国政府 5 年内将不再使用 SHA-1，取而代之的是更为先进的新算法，微软、Sun 等知名公司也纷纷发表各自的应对之策。由于在密码分析领域里的杰出贡献，王小云教授荣获 2006 年“中国青年女科学家奖”。

科研内容介绍：

一个安全的 hash 函数 f 应当满足以下三个条件：

(1) 任意 y，找 x，使得 f(x)=y，计算上不可行，该特性称为单向性(One-way)或不可逆性(irreversible)。

(2) 给定 x1，找 x2，使得 f(x1)=f(x2)，计算上不可行，该特性称为抗强碰撞性(strong collision-free)。

(3) 找 x1，x2，使得 f(x1)=f(x2)，计算上不可行，该特性称为抗弱碰撞性(weak collision-free)。

*上面的“计算上不可行”的意思是除了枚举外不可能有别的更快的方法。几乎所有的 hash 函数的破解，都是指的破坏上面的第三条性质，即抗弱碰撞性。（单向性和抗强碰撞性是 hash 函数的基石，一旦被破坏意味着 hash 函数被完全破解）。在密码学上还有一个概念是**理论破解**，指的是提出一个算法，使得可以用低于理论值的枚举次数找到碰撞。*

王小云教授等人在 Crypto’2004 上给出了 MD5 算法任意初始值的碰撞，并同时找出了 HAVAL-128、MD4 和 RIPEMD 算法的碰撞，此外还给出 SHA-0 和 HAVAL-160 的理论破解。在提交 RSA’2005 的论文中，王小云教授等人提出了一些有效查找 SHA-1 算法碰撞的新技术，通过分析得出 SHA-1 碰撞能够在少于 2^69 次 hash 操作内找出，这是针对 SHA-1 算法的第一次理论破解；论文中另外给出了 SHA-0 算法和 58 步 SHA-1 算法的实际碰撞实例。在 2005 年 8 月份的 Cypto’2005 上，王小云教授等人进一步将 SHA-1 算法的碰撞查找复杂度减少到 2^63 次 hash 操作，并找出了一个完整 80 步 SHA-0 算法的碰撞实例。

3. 攻击主机的 IP 地址是什么？
4. 网络扫描的目标 IP 地址是什么？

使用 wireshark(本文使用的是 v0.99.5 版本)打开待分析的二进制记录文件。

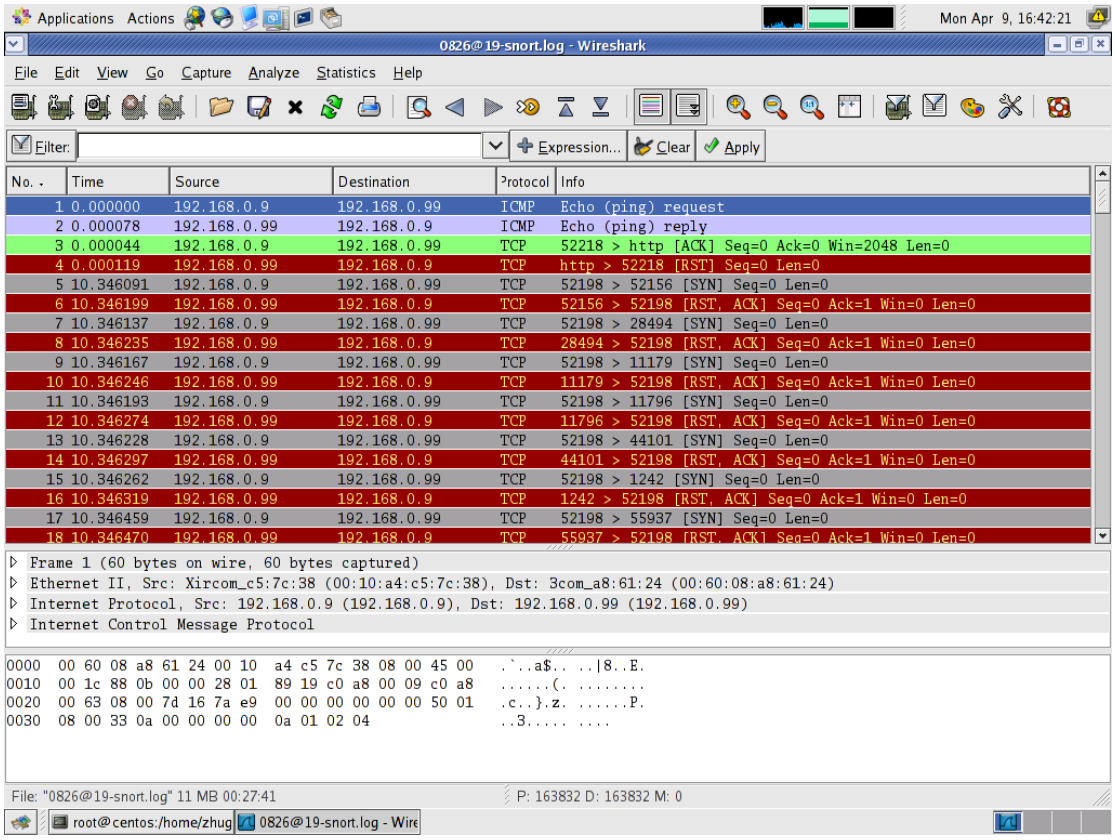

图 1 使用 wireshark 软件打开待分析二进制记录文件

使用 Statistics(统计)菜单项中的 Conversation List(会话列表)，选择 IPv4，得到如下的分析结果：

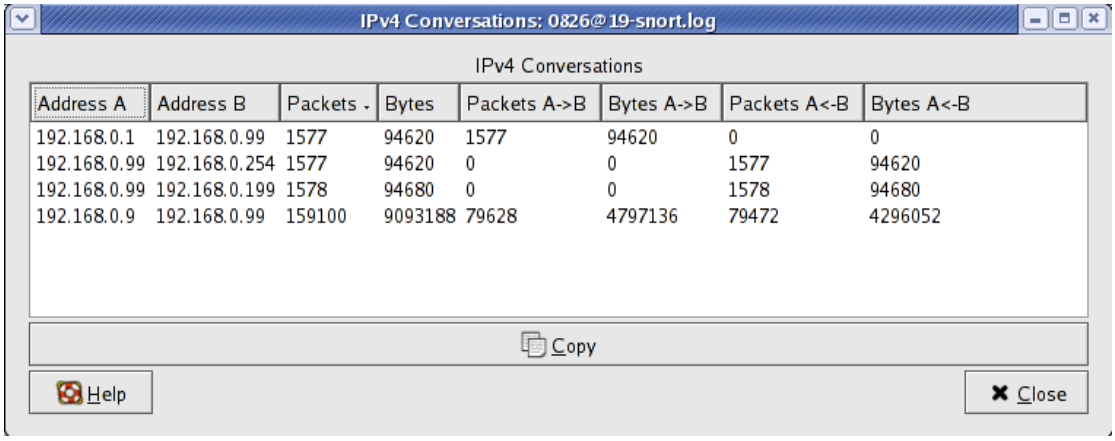

图 2 待分析二进制记录文件进行 IPv4 会话列表的统计结果

从图中我们可以看出，只有 192.168.0.9 和 192.168.0.99 之间有双向的网络数据包，因此可初步确定两者为攻击主机 IP 和目标主机 IP。从数据包内容看，所有的响应数据包（如 TCP RST 包、SYN/ACK 包、ICMP Echo Reply 包等）均是从 192.168.0.99 发出，**可以确定 192.168.0.99 是被扫描的目标主机**。而请求数据包（如 TCP SYN 包、ICMP Echo 包等）则是从 192.168.0.{1, 9, 199, 254}发起。并且从图中的统计看出没有从 192.168.0.99 到 192.168.0.{1, 199, 254}的响应包，因此**可以确定 192.168.0.9 是攻击主机的 IP 地址**。

而从 192.168.0.{1, 199, 254}只有发往 192.168.0.99 的请求包，而没有响应包，可以初步推测从 192.168.0.{1, 199, 254}发出的请求包是伪造源 IP 数据包，这一推测将在问题 5 解答中进行证实。

5. 本次案例中是使用了哪个扫描工具发起这些端口扫描？你是如何确定的？

我们使用 Snort 工具对二进制记录文件进行入侵检测，确定攻击者使用的扫描工具。

Snort 工具的安装

下载获得 snort 最新的源码发行包，在 Snort 网站上注册并获取最新的规则文件包

```
# tar -zxf snort-2.6.1.4.tar.gz
# cd snort-2.6.1.4
# ./configure
# make
# cp ../ snortrules-snapshot-CURRENT.tar.gz .
# tar -zxf snortrules-snapshot-CURRENT.tar.gz
```

```
# snort -r ../0826@19-snort.log -c ./etc/snort.conf
```

报警结果显示如下报警信息，从这些报警信息中可以确认本案例中使用了由 Fyodor 所开发的著名开源网络扫描器 nmap(http://insecure.org/nmap/)。

[**] [1:469:4] ICMP PING NMAP [**]

6. 什么是端口扫描，端口扫描的目的是什么？

端口扫描就是连接目标主机的 TCP 和 UDP 端口，确定哪些服务正在运行即处于监听状态的过程。端口扫描的目的是确定一个目标主机上所开放的网络端口和网络服务，端口扫描目前被防御者和攻击者都广泛使用。对于防御者，使用端口扫描可以让他们更加了解所管理的网络状况，找出没有必要开放的端口并关闭，这是保证业务网络安全的第一步。对于攻击者而言，使用端口扫描的目的是找出可供进一步攻击的网络服务，同时结合操作系统探测技术也可以确定目标主机所安装的操作系统版本。开放网络服务和操作系统版本信息为攻击者提供了破解攻击的目标，使其更容易找出进入目标主机的漏洞路径。

7. 本次案例中使用了 5 种不同类型的扫描方式，请将这 5 种方法标识出来，并描述其工作原理。

- 关键步骤－确定各次扫描的边界

第一次扫描产生了大量的 TCP SYN 包和关闭端口反馈的 RST 包，我们需要找出各次扫描的范围才能进行对后续扫描过程的分析，在 wireshark 中人工地找出各次扫描的起始和终结位置是比较累眼神的，我们还是想些办法出来，以免折磨我们原本就视疲劳的“心灵窗口”。

由于我们已经确定了这些扫描是由 nmap 所发起的，而 nmap 在发起端口扫描之前总是先通过 Ping 扫描和针对 80 端口的探测确定目标主机是否活跃，因此我们可以通过找出 ICMP Echo 包的序号快速地划分出各次扫描的范围。

```
[root@centos scanofthemonth]# tshark -r 0826\@19-snort.log icmp.type == 8
     1    0.000000    192.168.0.9 -> 192.168.0.99 ICMP Echo (ping) request
148007 1274.602300    192.168.0.9 -> 192.168.0.99 ICMP Echo (ping) request
150753 1407.256096    192.168.0.9 -> 192.168.0.99 ICMP Echo (ping) request
155987 1602.084879    192.168.0.1 -> 192.168.0.99 ICMP Echo (ping) request
155988 1602.084912 192.168.0.254 -> 192.168.0.99 ICMP Echo (ping) request
155989 1602.084941 192.168.0.199 -> 192.168.0.99 ICMP Echo (ping) request
155990 1602.084976 192.168.0.199 -> 192.168.0.99 ICMP Echo (ping) request
```

从 ICMP Echo 包结果看，我们已经确定出 4 次扫描的起始点（注：后四个 ICMP Echo 包的序号和时间连续，显然是属于同一次扫描），与案例分析给出的 5 次扫描事件还缺少一个，显然某次扫描屏蔽了端口扫描前的主机活跃探测。结合人为在 wireshark 工具上对各次扫描范围的确认，我们可以确定出这 5 次扫描的各自范围、包数量和持续时间：

扫描事件	起始序号	结束序号	包数量	开始时间	结束时间	持续时间(秒)
第一次扫描	1	148006	148006	0	1245.35	1245.35
第二次扫描	148007	150752	2746	1274.6	1406.67	132.07
第三次扫描	150753	153250	2498	1407.26	1491.47	84.21
第四次扫描	153237	155986	2749	1489.56	1597.67	108.11
第五次扫描	155987	163832	7846	1602.08	1661.98	59.9

注：第三次扫描和第四次扫描的数据包范围有交叉。

- Scan #1: TCP SYN 扫描

使用 wireshark 查看待分析二进制记录文件，首先我们可以看到从攻击主机到目标主机的一对 ICMP Echo 和 ICMP Echo Reply 数据包，显然是在端口扫描前进行 Ping 扫描确定目标主机是否活跃，另外第 3 个发往 80 端口的 ACK 包也是另一种通过查看目标端口 80 端口是否开放确定其是否活跃的方法。然后从第 5 个数据包开始的就是从攻击主机 192.168.0.9 到目标主机 192.168.0.99 的 SYN 包扫描。通过设置过滤规则"frame.number < 148007 and tcp.flags != 0x2 and tcp.flags != 0x14"可以查看除了 SYN 包和非开发端口响应的 RST/ACK 包外的其他数据包，在结果中可以发现攻击主机对目标端口上开放的端口进一步发送 RST 包断开连接，因此我们可以确定第一次扫描属于 TCP SYN 扫描。

对照 nmap 的使用手册，我们可以发现 nmap 的-sS 选项代表了选择 TCP SYN 扫描方式，而第一次扫描过程中涉及的目标端口包括了从 1 到 65535 的全部端口号，因此可以推断发起第一次扫描的指令是"# nmap -sS -p 1-65535 192.168.0.99"。

攻击技术内幕：

Ping 扫描：目的为检查目标主机是否活跃(active)，传统的 ping 扫描是通过向指定的目标主机发送 ICMP 回射请求报文(ICMP echo, icmp.type=8)，并期待活跃主机反馈的 ICMP 回射应答报文(ICMP echo reply, icmp.type=0)，根据是否收到应答报文确定目标主机是否活跃。同时扫描大量的 IP 地址段，以发现某个 IP 地址是否绑定活跃主机的网络扫描称为 Ping 扫射。由于在某些实际环境中，防火墙通常配置为阻断外部进入的 ICMP 报文，因此攻击者也常使用向目标主机的 80 等常用应用协议端口（防火墙通常会放行）发送 ACK 等报文，活跃主机将返回 SYN/ACK 或 RST 包，非活跃主机则无应答，从而可绕过防火墙探测目标主机是否活跃，这种 Ping 扫描方式称为 ACK Ping 扫描。其他的 Ping 扫描方式还包括 SYN Ping 扫描，UDP Ping 扫描等。Nmap 工具-sP 选项为选择 Ping 扫描功能，缺省在每次扫描前都会执行，并行使用 ICMP Ping 扫描和 ACK Ping 扫描两种方式。

端口扫描：端口扫描是连接到目标主机的 TCP 和 UDP 端口上，确定哪些服务正在运行即处于监听状态的过程，标识监听状态端口对于确定目标主机操作系统和应用程序类型至关重要，攻击者通过攻击监听活动服务上的已知安全漏洞或配置不当可远程取得目标主机的访问权。

TCP 连接扫描：这是最基本的 TCP 扫描方式，通过操作系统提供的 connect()系统调用向目标主机上的端口发起网络连接，如果目标端口开放，connect()调用将成功，否则调用失

败。这种扫描方式最大的优势在于发起者并不需要任何特殊权限，在类 Unix 平台下的任意用户都可以使用。但是这种扫描方式是很容易检测的，因为目标主机的日志中会留下一大堆的连接和错误信息。TCP 连接扫描是非特权用户的缺省扫描方式。

TCP SYN 扫描：通常也被称为半开扫描，因为扫描主机不会完成完整的 TCP 三次握手连接：首先像打开一个真正的网络连接一样发送 SYN 请求包；当目标主机反馈一个 SYN|ACK 包，就表明目标端口是开放的，而当目标端口关闭时，则将反馈一个 RST 包；当扫描主机接收到 SYN|ACK 包后，将会立即发送一个 RST 包关闭这个连接（实际上是由扫描主机操作系统内核自动完成的）。半开扫描方式的主要优点在于极少数站点会记录未建立的网络连接，因此也就不会发现被扫描。但不幸的是，发起半开扫描需要根用户权限构建定制的 SYN 包。TCP SYN 扫描是特权用户缺省的扫描方式。

- Scan #2: NULL 扫描+操作系统探测

从第二次扫描开始的数据包中可以发现，攻击主机发出的是没有设置任何标志位的 TCP 包，即 Null 包，对照 nmap 用户手册，可以确认是由-sN 选项所指定的 Null 扫描方式；进一步通过设置过滤器"frame.number > 148006 and frame.number < 150753 and tcp and ip.src != 192.168.0.99"过滤出第二次扫描中发往目标主机的 TCP 包，从中可以发现该次扫描的目标端口列表与/etc/services 中列出的基本一致，而 nmap 工具的-F 选项指定使用快速扫描模式，即只扫描 services 文件中所列出的端口，因此可以确认第二次扫描使用了-F 快速扫描模式；最后我们可以在第二次扫描的最后阶段（数据包序号 150639 至 150752）发现如下表所示的一系列网络探测，主要针对 ssh、TCP 和 UDP 的 1 端口，并使用了大量构造的标志位，以触发不同的响应包，并尝试从响应包中识别目标主机的操作系统信息。

nmap 的-O 选项是用于通过 TCP/IP 协议栈"指纹"识别进行远程主机的操作系统探测，具体实现机制是通过构造一些特殊的数据包发送给目标主机，然后收集反馈包中存在的一些微妙"指纹"（不同操作系统网络协议栈实现上的差异），并与一个已知操作系统"指纹"数据库（nmap-os-fingerprints 文件）进行比较，从而识别目标主机的操作系统版本。对照 nmap 对操作系统版本探测的实现机制，不难确认这些流量是由其-O 选项所触发的。

综合以上分析，发起第二次扫描的指令是"# nmap -sN -F -O 192.168.0.99"

协议	端口	标志位
TCP	ssh 22	SYN, ECN
TCP	ssh 22	Null
TCP	ssh 22	FIN, SYN, PSH, URG
TCP	ssh 22	ACK
TCP	ssh 22	RST
TCP	1	SYN
TCP	1	ACK
TCP	1	FIN, PSH, URG
UDP	1	
TCP	ssh 22	……

- Scan #3: Xmas 扫描+操作系统探测

利用同样的分析方法，可以确定第三次扫描的发起指令"# nmap -sX -p 1-1024 -O 192.168.0.99"。

攻击技术内幕：

隐蔽 FIN、Null、Xmas 扫描方式：在某些场景下 SYN 扫描也并非足够隐蔽，某些防火墙和数据包过滤器会监控发往受限端口的 SYN 包，一些软件如 Synlogger 和 Courtney 可以检测出 SYN 扫描。而这些更加隐蔽的扫描方式，则可能躲避检测，从而完成扫描的目的。FIN 扫描方式使用只带 FIN 标志位的 TCP 包，Null 扫描方式使用所有 TCP 标志位都设置为 0 的的 TCP 包，Xmas 扫描方式则使用带 FIN、PSH、URG 标志位的 TCP 包，这三种标志位组合方式在标准的 TCP 协议规范中并没有定义，而开放的端口接收到伪造的 TCP 包，按照规范应直接丢弃，而不向发送方进行任何反馈，而对于关闭的端口则反馈一个 RST 包，因此根据这两种不同的反馈方式，就可以确定目标端口是否开放。由于 Windows、Cisco、 BSDI、HP/UX、MVS 和 IRIX 系统在协议栈实现上并没有遵从规范，开放端口对这些伪造的 TCP 包也会发送 RST 包，因此这三种扫描方式无法适用于这些系统。

操作系统辨识技术：

操作系统辨识是通过各种不同操作系统类型和版本实现机制上的差异，通过特定方法以确定目标主机所安装的操作系统类型和版本的技术手段，明确操作系统类型和版本是进一步进行安全漏洞发现和渗透攻击的必要前提。各种不同操作系统类型和版本在网络角度上的差异体现在：开放端口的不同、运行的应用服务类型的不同以及协议栈实现细节上的差异。因此对操作系统辨识的方法包括端口扫描、应用服务旗标攫取和协议栈指纹鉴别，而按照获取这些信息方式的不同又分为操作系统主动探测技术和被动操作系统识别技术。

操作系统主动探测技术：

操作系统主动探测技术包括端口扫描、应用服务旗标攫取和主动协议栈指纹鉴别。其中主动协议栈指纹鉴别是目前最有效和最常用的探测方式，其经典代表论文为 Fyodor 于 1998 年在 Pharck 杂志上发表的 Remote OS detection via TCP/IP Stack Finger-Printing。在论文中提出的可用来探测不同操作系统类型和版本的技巧包括：FIN 探测报文、假标志位探测报文、ISN 采样、DF 位监视、初始窗口大小, ACK 值, ICMP 出错消息抑制, ICMP 消息引用, ICMP 出错消息回射完整性, TOS,重叠分片处理, TCP 选项等。实现主动协议栈指纹鉴别的经典工具为 Fyodor 的 queso 和 nmap 工具-O 选项。

- Scan #4: TCP 连接扫描+操作系统探测

我们继续在 wireshark 中查看第四次扫描，与之前三次不同的是，第四次扫描并没有出现 ICMP Echo 数据包，而是仅仅使用了 TCP 80 端口的 ACK 包探测目标端口是否活跃；此外与第一次扫描不同的是，通过设置过滤规则"frame.number > 153237 and frame.number < 155986 and ip.src != 192.168.0.99 and tcp.flags.syn != 1"可发现攻击主机对开放的端口发送了 ACK 包建立三次握手后，又发送了 RST|ACK 包关闭连接，而这种模式是典型的 TCP 连接扫描；通过前面给出的方法分析目标端口列表可进一步确定此次扫描采取快速扫描模式；而最后阶段也进行了操作系统的探测，因此，可以确认第四次扫描的指令是"# nmap -sT -PA -F -O 192.168.0.99"。

- Scan #5: 伪造源 IP 地址的 Xmas 扫描

最后，对第五次扫描数据包的观察可发现扫描方式与第三次相同，为 Xmas 扫描方式，不同的是这次扫描中的源 IP 地址出现了 192.168.0.{1, 9, 199, 254}这四个 IP 地址，除了攻击主机 192.168.0.9 之外，从捕获流量看还有其他三个 IP 地址在进行扫描，因此可以确认这次扫描是使用了 nmap 工具的-D 选项，使用了源 IP 地址欺骗技术，经过进一步分析可以得出

扫描指令是"# nmap -sX -F -D 192.168.0.1, 192.168.0.254, 192.168.0.199 192.168.0.99"。

注：本案例中由于攻击主机、伪装的 IP 地址和目标主机在同一子网内，因此目标主机在发送响应包前会通过 ARP 查询寻找 192.168.0.{1, 199, 254}这些伪装的 IP，可以确定在本案例部署场景中这些 IP 并不活跃，因此目标主机不会向伪装 IP 地址做出响应。在伪装 IP 与目标主机不在同一子网的一般场景下，目标主机将会向伪装 IP 地址发送响应包。

攻击技术内幕：

IP 地址欺骗技术(IP Spoofing)：通过伪造数据包中的源 IP 地址以达到躲避检测和追溯，或者伪装成受信主机进行中间人攻击的技术手段。IP 地址欺骗技术的根源在于 IP 和路由协议设计中只按照目标 IP 地址对数据包进行路由和转发，而没有对源 IP 地址的合法性进行有效验证。通过源 IP 地址欺骗技术进行扫描或拒绝服务攻击较为简单，也非常常见，通过特定工具修改发送数据包的源 IP 地址即可。而通过源 IP 地址欺骗技术进行中间人攻击则需要能够监听目标主机向伪装 IP 地址发送的响应包，并通过分析进行会话劫持，以达成中间人攻击效果。

Nmap 工具的-D 选项将发起一次源 IP 欺骗扫描，即看起来像扫描是从你所指定的源 IP 地址所发起的一样。这样如果目标网络中的 IDS 发现这次扫描事件并进行了报警，那么报警信息将显示从伪造的源IP 地址所发起的扫描，而无法知道真正发起扫描的IP 地址是什么。尽管源 IP 地址欺骗可能会被路由器路径追踪、响应丢弃和其他的一些"主动"机制解决，但在一般情况下仍是一种隐藏自身位置的有效技巧。

8. 在蜜罐主机上哪些端口被发现是开放的？

因为第一次扫描（半开 TCP 扫描）对从 1 到 63335 的全部 TCP 端口进行了探测，因此我们可以以这次扫描的结果确定蜜罐主机上哪些端口是开放的。根据 TCP 协议规范，开放服务将响应 SYN 请求包，反馈 SYN|ACK 包，因此我们通过设置 tcpdump filter "ip.src == 192.168.0.99 and tcp.flags == 0x0012 and frame.number <= 148006"就可以找出开放的端口。

```
# tshark -r 0826\@19-snort.log -nn ip.src == 192.168.0.99 and "tcp.flags == 0x0012 and frame.number < 148007"
```

从输出结果可以得出，蜜罐主机上开放的端口和对应的网络服务如下：

- 22 SSH
- 111 SUNRPC
- 32768 rpc.statd/NFS file locking
- 80 HTTP
- 443 HTTPS
- 53 DNS

9. 额外奖励问题：攻击主机的操作系统是什么？

要确定一台主机的操作系统存在两种常用的方法：一种是主动的操作系统版本探测，常用的工具包括 queso 和 nmap，但在这个离线的案例分析中，我们无法对攻击主机进行探测，即使防御者处于在线分析的场景下，为了不让攻击者警觉对他的调查，一般情况下也不应执行对攻击者的主动探测；另一种是被动的操作系统版本辨识，由于不同的操作系统平台和版本在实现网络协议栈时存在差异，从而导致在网络数据包上存有各种操作系统的"指纹"，根据从抓取的数据包中识别这些"指纹"，我们就可以辨识出数据包发送者的操作系统平台和版本。常用的被动操作系统版本辨识工具为 p0f，它是由 Michael Zalewski 和 Bill Stearns 所开发和维护的

开源项目。

p0f 工具一般不会在各种 Linux 发行版中默认安装，但可以通过 yum 获取和安装。

yum -y install p0f

使用 p0f 的-s 参数可读取待分析的二进制记录文件，从中识别发送方的操作系统信息，由于 p0f 中也包含了针对 nmap 扫描的"指纹"，所以在输出中也显示了大量"192.168.0.9 [xx hops]: NMAP scan (distance inaccurate) "结果，这也验证了问题 6 中确认的攻击主机确实使用了 nmap 作为端口扫描工具。通过 grep -v 排除掉 nmap 的识别结果后，我们得到 "192.168.0.9:xxxxx - Linux 2.4-2.6 (up: 46 hrs) -> 192.168.0.99:xxxxx (distance 0, link: ethernet/modem) "的辨识信息，从中我们可以确认攻击主机安装的是 Linux 操作系统，内核版本为 2.4 系列或 2.6 系列，此外 p0f 还给出了攻击主机的在线时间(46 小时)，与目标主机的距离(TTL 递减值为 0，即攻击主机与目标主机在同一子网)等其他信息。

p0f -s 0826\@19-snort.log | grep -v NMAP | more

192.168.0.9:xxxxx - Linux 2.4-2.6 (up: 46 hrs) -> 192.168.0.99:xxxxx (distance 0, link: ethernet/modem)

攻击技术内幕：

被动操作系统识别技术：

被动操作系统识别技术包括网络监听、被动应用服务识别和被动协议栈指纹鉴别。其中被动协议栈指纹鉴别是目前最有效和最常用的被动操作系统识别技术。

Lance Spitzner 在此领域做了很多研究工作，并写了一篇技术白皮书"Passive Fingerprinting"来讲解他的发现(http://packetstormsecurity.org/papers/unix/finger.htm)，可用来被动识别操作系统类型的四个主要特征为：

- *TTL: 不同的操作系统类型设置数据包的 Time to Live 缺省值有一些差异;*
- *窗口大小: 不同的操作系统类型设置 TCP 数据包的窗口大小字段存在差异;*
- *DF 位: 不同的操作系统类型是否设置 DF 位存在差异;*
- *TOS: 不同的操作系统类型是否设置 TOS 位，设置的值存在差异.*

通过对不同操作系统类型和版本的这些特征变量取值构建指纹数据文件，即可作为被动操作系统识别的依据。实现被动协议栈指纹鉴别的经典工具为 siphon(已过时)和 p0f v2，其使用的指纹数据文件位 p0f.fp，使用了" wwww:ttt:D:ss:OOO...:QQ:OS:Details"的指纹格式，使用特征包括 Window size (WSS)、Initial TTL、Don't fragment flag、overall SYN packet size、option value and order specification、quirks list 等，最后给出识别的 OS genre 和 OS description。

参考文献：

Scan of the Month 23: http://www.honeynet.org/scans/scan23/

Writeup from Richard La Bella of the South Florida Honeynet Project

Top Three Entries

- Nick DeBaggis
- Leon Ward
- Laurent Butti

Ethereal 木马事件：

http://www.cert.org/advisories/CA-2002-30.html

王小云教授：

http://www.view.sdu.edu.cn/news/news/sdyw/2004-09-04/1094261946.html

http://www.ygb.sdu.edu.cn/html/xssk/xszx/083152.html

http://www.gmw.cn/content/2007-03/15/content_569216.htm

http://zhiqiang.org/blog/446.html

Xiaoyun Wang, Collisions for Hash Functions MD4, MD5,HAVAL-128 and RIPEMD,Crypto'04,E-print.

Xiaoyun Wang, Yiqun Yin, Hongbo Yu, Collision Search Attacks on SHA1,2005.

Xiaoyun Wang, Yiqun Yin, Hongbo Yu, Finding Collisions in the Full SHA-1,Crypto'05.

Xiaoyun Wang1, Hongbo Yu, Yiqun Lisa Yin, Efficient Collision Search Attacks on SHA-0,Crypto'05.

基础知识：

《计算机网络》第四版　　IP/TCP/ICMP 协议

黑客大曝光　　网络扫描节

Linux：

Md5sum

Tcpdump

Ethereal

snort

p0f　　yum install p0f

nmap

涉及到的攻击技术

网络扫描：Ping 扫描、端口扫描

操作系统探测&辨识

源 IP 地址欺骗技术