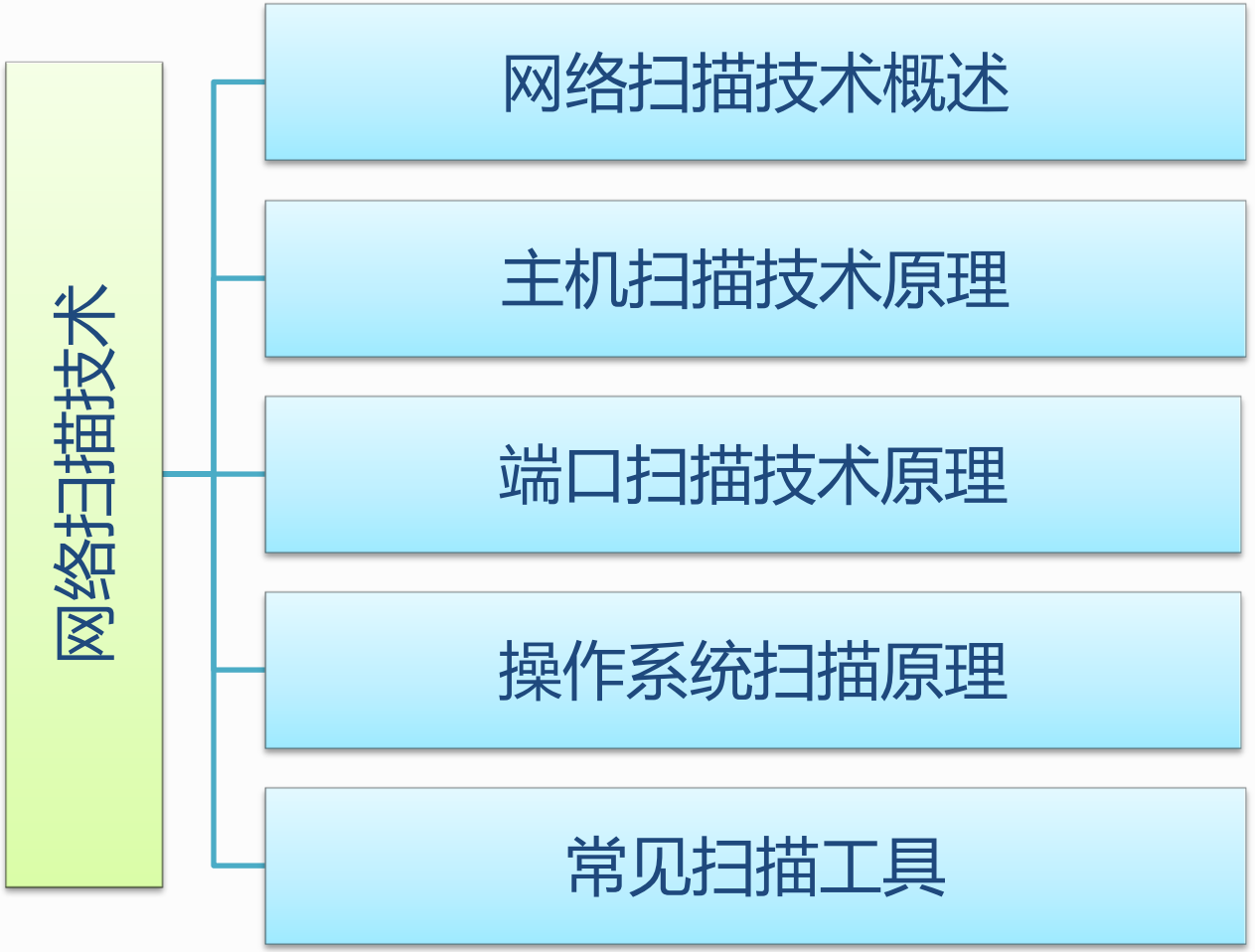
第五讲 网络扫描技术

了解网络扫描的基本概念；理解不同网络扫描技术的原理和特点；掌握简单网络扫描工具的设计与实现。

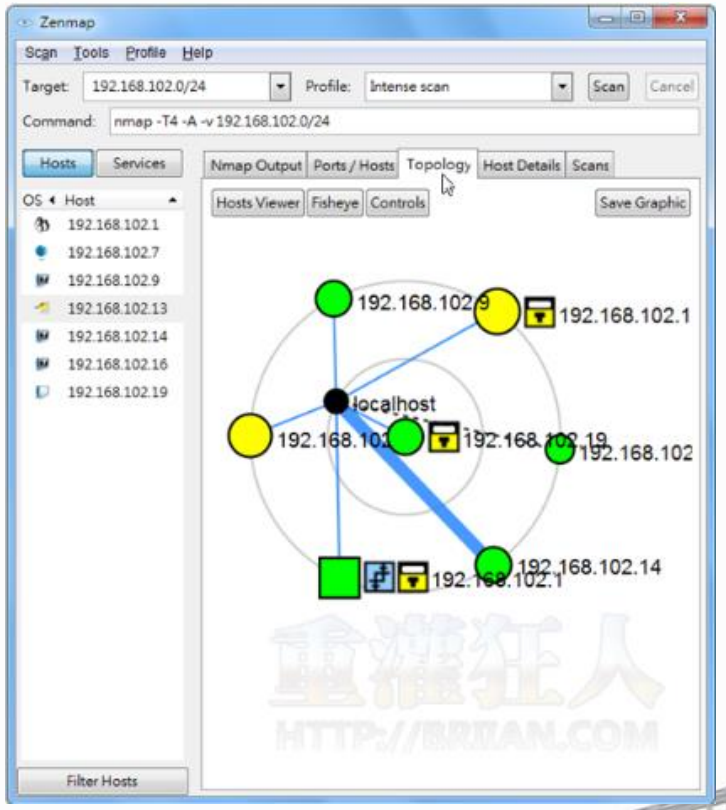
内容安排



第五讲 网络扫描技术

一、网络扫描技术概述

- 什么是网络扫描？
  - 网络扫描技术是一种用于发现Internet远程目标网络或本地主机安全性脆弱点的技术。通过网络扫描，可以获取各种TCP/IP端口的分配、开放的服务、Web服务软件版本和这些服务及软件呈现在Internet上的安全漏洞 。



第五讲 网络扫描技术

一、网络扫描技术概述

- 网络扫描的过程
  - 第1阶段：发现目标主机或网络；
  - 第2阶段：发现目标后进一步搜集目标信息，包括运行的服务、操作系统类型以及服务软件的版本等。如果目标是一个网络，还可以进一步发现该网络的拓扑结构、路由设备以及各主机的信息；
  - 第3阶段：根据搜集到的信息判断或者进一步测试系统是否存在安全漏洞。

一、网络扫描技术概述

网络扫描技术的分类

- 主机扫描：确定在目标网络上的主机是否可达，同时尽可能多映射目标网络的拓扑结构，主要利用ICMP数据包；
- 端口扫描：发现远程主机开放的端口以及服务；
- 操作系统扫描：根据扫描获取的特征判别操作系统；
- 漏洞扫描：检测出目标系统存在的安全漏洞。

二、主机扫描技术原理

- 传统主机扫描技术
  - ICMP Echo ：通过简单地向目标主机发送ICMP Echo Request 数据包，并等待回复的ICMP Echo Reply 包，如Ping；
  - ICMP Sweep（Ping Sweep）：使用ICMP Echo Request一次探测多个目标主机。通常这种探测包会并行发送，以提高探测效率，如pingsweep工具；
  - Broadcast ICMP ：设置ICMP请求包的目标地址为广播地址或网络地址，则可以探测广播域或整个网络范围内的主机，这种情况只适合于UNIX/Linux系统；
  - Non-Echo ICMP：其它ICMP服务类型（13和14、15和16、17和18）也可以用于对主机或网络设备如路由器等的探测。

二、主机扫描技术原理

ICMP报文类型

- 0 Echo Reply
- 3 Destination Unreachable
- 4 Source Quench
- 5 Redirect
- 8 Echo
- 11 Time Exceeded
- 12 Parameter Problem
- 13 Timestamp
- 14 Timestamp Reply
- 15 Information Request（已不再使用）
- 16 Information Reply（已不再使用）
- 17 Address Mask Request（已不再使用）
- 18 Address Mask Reply（已不再使用）

二、主机扫描技术原理

- 操作系统对ICMP消息的处理（非广播）

操作系统	直接的ICMP消息类型（非广播）			
	8	13	15	17
Linux	是	是	否	否
*BSD	是	是	否	否
Solaris	是	是	否	是
HP-UX	是	是	是	否
AIX	是	是	是	否
Ultrix	是	是	是	是
Windows 95\98\me	是	是	否	是
Windows NT 4.0	是	否	否	否
Windows 2000	是	是	否	否
Cisco IOS	是	是	是	否

二、主机扫描技术原理

- 操作系统对ICMP消息的处理（广播）

操作系统	直接的 ICMP 消息类型（广播）			
	8	13	15	17
Linux	是	是	否	否
*BSD	否	否	否	否
Solaris	是	是	否	否
HP-UX	是	是	是	否
AIX	否	否	否	否
Ultrix	否	否	否	否
Windows 95\98\me	否	否	否	否
Windows NT 4.0	否	否	否	否
Windows 2000	否	否	否	否
Cisco IOS	否	否	是	否

二、主机扫描技术原理

- 高级主机扫描技术：利用被探测主机产生的ICMP错误报文来进行复杂的主机探测。
  - 异常的IP包头
    - 向目标主机发送包头错误的IP包，目标主机或过滤设备会反馈ICMP Parameter Problem Error信息。常见的伪造错误字段为Header Length 和IP Options。不同厂家的路由器和操作系统对这些错误的处理方式不同，返回的结果也不同。
  - 在IP头中设置无效的字段值
    - 向目标主机发送的IP包中填充错误的字段值，目标主机或过滤设备会反馈ICMP Destination Unreachable信息。这种方法同样可以探测目标主机和网络设备。

二、主机扫描技术原理

- 通过超长包探测内部路由器
  - 若构造的数据包长度超过目标系统所在路由器的PMTU且设置禁止分片标志，该路由器会反馈 Fragmentation Needed and Don’ t Fragment Bit was Set差错报文。
- 反向映射探测：用于探测被过滤设备或防火墙保护的网络和主机
  - 构造可能的内部IP地址列表，并向这些地址发送数据包。当对方路由器接收到这些数据包时，会进行IP识别并路由，对不在其服务的范围的IP包发送ICMP Host Unreachable或ICMP Time Exceeded 错误报文，没有接收到相应错误报文的IP地址可被认为在该网络中。

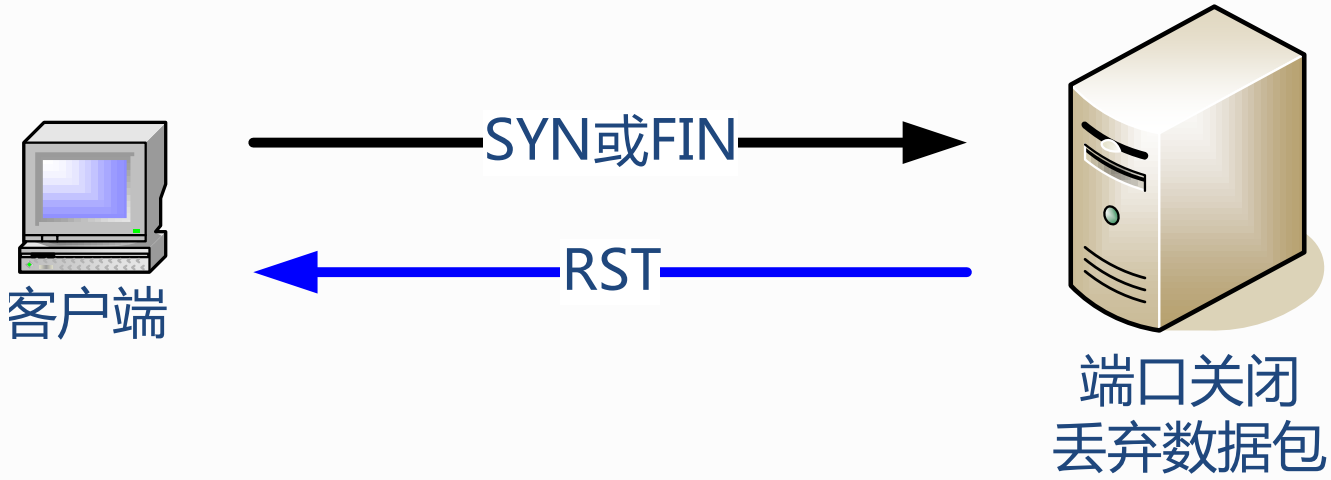
- 测试点 5-1
  - 主机扫描技术是利用ICMP协议来实现，请查阅相关资料，了解ICMP协议的工作原理，并简要说明Ping功能的实现原理。

三、端口扫描技术原理

- 什么是端口扫描？
  - 一个端口就是一个潜在的通信通道，也就是一个入侵通道。对目标计算机进行端口扫描，能得到许多有用的信息，从而发现系统的安全漏洞。它使扫描者了解系统目前向外界提供了哪些服务，从而为系统的入侵提供了一种手段。
- 端口扫描分类
  - 开放/全连接扫描
  - 半开放/半连接扫描
  - 秘密扫描

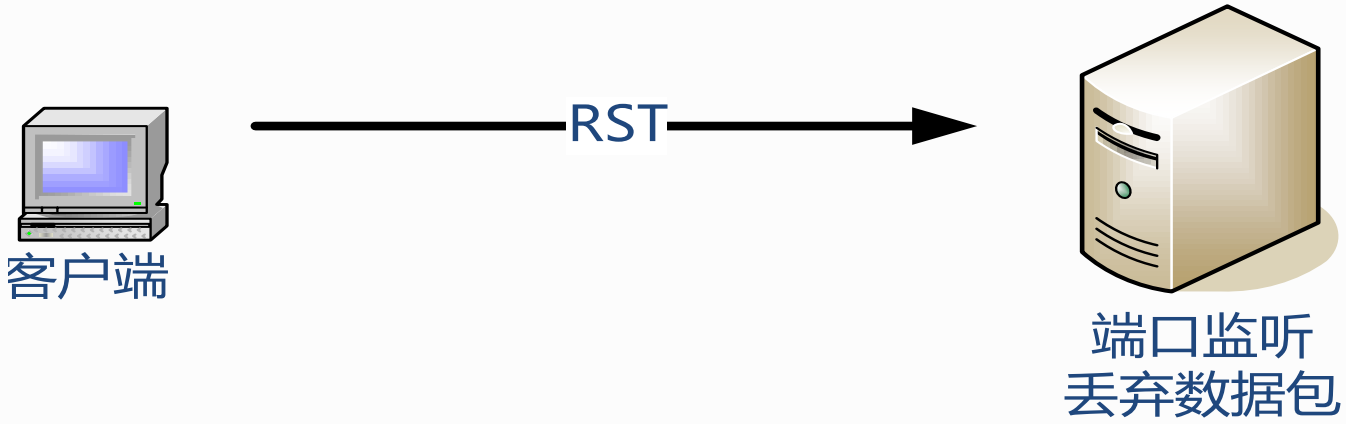
三、端口扫描技术原理

- 多数TCP/IP协议实现遵循的原则（一）
  - 当直接发送一个SYN或者FIN数据包到达一个关闭的端口，TCP丢弃数据包同时发送一个RST数据包。



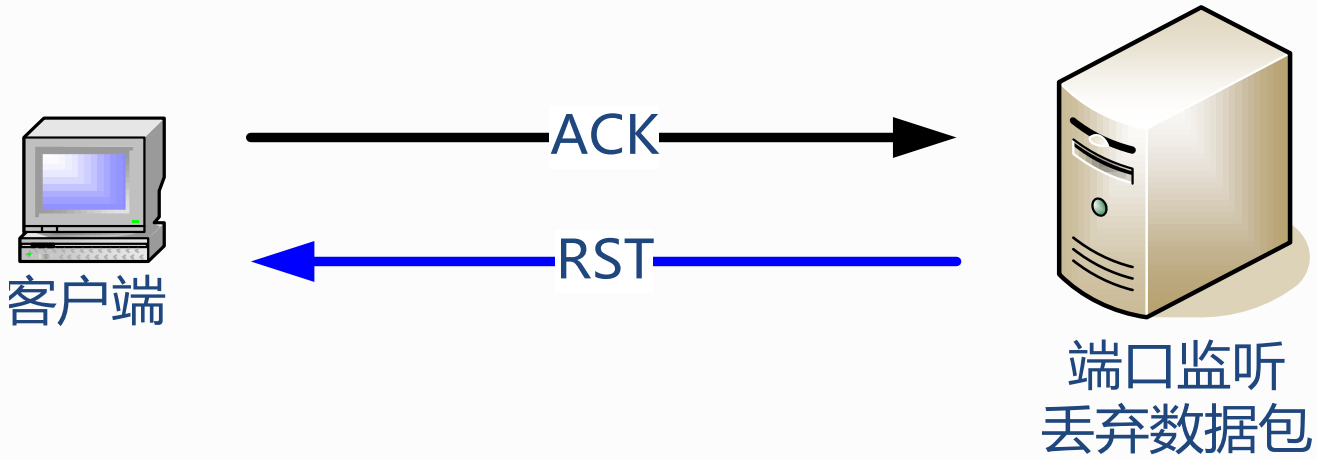
三、端口扫描技术原理

- 多数TCP/IP协议实现遵循的原则（二）
  - 当直接发送一个RST数据包到达一个监听端口，RST被丢弃。



三、端口扫描技术原理

- 多数TCP/IP协议实现遵循的原则（三）
  - 直接发送一个包含ACK的数据包到达一个监听端口时，数据包被丢弃，同时发送一个RST数据包。



三、端口扫描技术原理

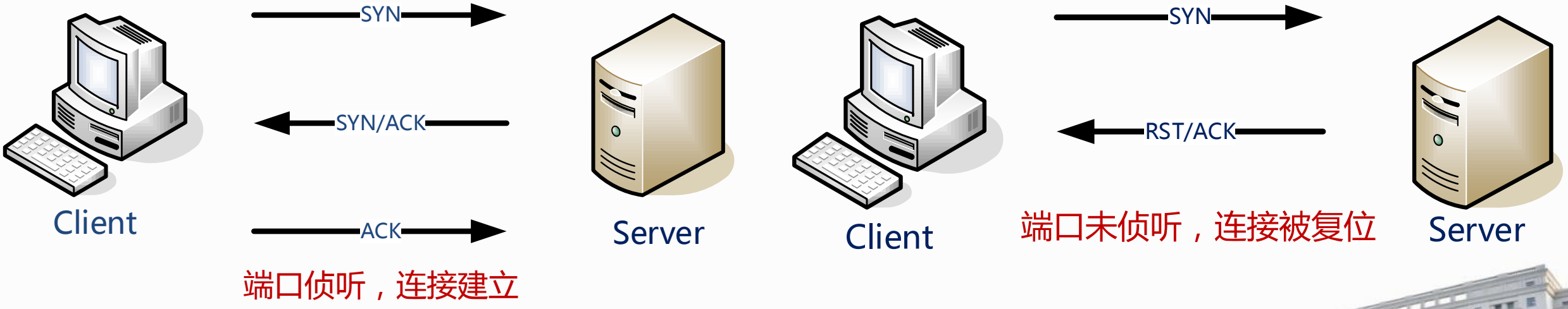
- 多数TCP/IP协议实现遵循的原则（四）
  - 当一个SYN位关闭的数据包到达一个监听端口时，数据包被丢弃。
  - 当一个SYN数据包到达一个监听端口时，正常的三阶段握手继续，回答一个SYN|ACK数据包。

三、端口扫描技术原理

- 多数TCP/IP协议实现遵循的原则（五）
  - 当一个FIN数据包到达一个监听端口时，关闭的端口返回RST，监听端口丢弃包（FIN行为），在URG和PSH标志位置位时同样要发生。所有的URG，PSH和FIN，或者没有任何标记的TCP数据包都会引起"FIN行为"。

三、端口扫描技术原理

- TCP Connect扫描（开放扫描）
  - 向目标系统的目标端口发起连接，能建立连接，则表示端口开启，反之不然。



三、端口扫描技术原理

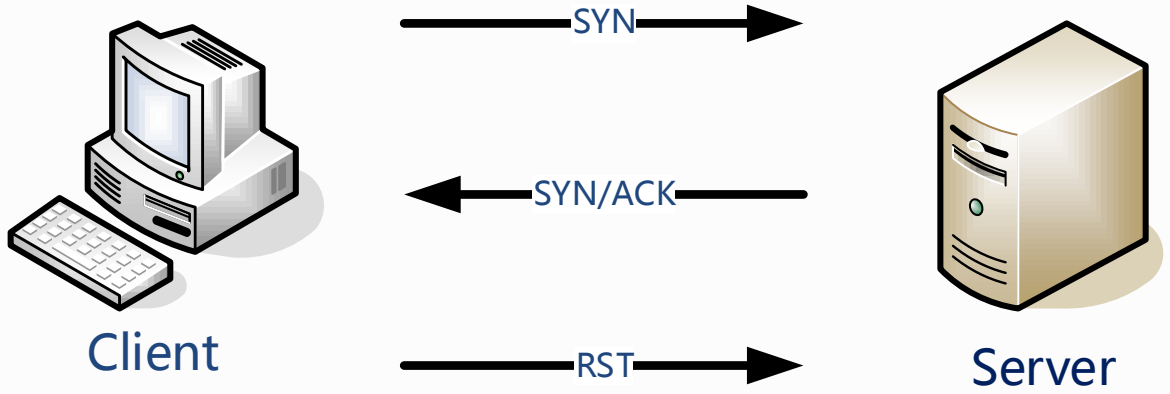
TCP Connect部分代码实现

```
for(int i=m; i<n; i++)
        {
          if((mysocket = socket(AF_INET, SOCK_STREAM,0)) == INVALID_SOCKET)
                exit(1);
              ……
        if(connect(mysocket, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == SOCKET_ERROR)
              {       printf("Port %d - 关闭\n", i);
                closesocket(mysocket);                    }
              else{   pcount++;
                printf("Port %d - 打开\n", i);    }
        }
        printf("%d ports open on host - %s\n", pcount, adr);
        closesocket(mysocket);
        WSACleanup();
       }
```

三、端口扫描技术原理

TCP SYN扫描（半开放扫描）

- 扫描主机向目标主机的选择端口发送SYN数据段。如果应答是RST，那么说明端口是关闭的，按照设定就探听其它端口；如果应答中包含SYN和ACK，说明目标端口处于监听状态，扫描主机传送一个RST给目标机从而停止建立连接。



三、端口扫描技术原理

- TCP SYN扫描的技术实现
  - “控制”TCP首部的标志位
    - 为了能“控制”TCP的首部，按需要修改TCP的标志位，必须使用原始数据协议（RAW SOCKET）来创建套接字。
  - 接收返回的数据包
    - 设置网卡为混杂模式；
    - 绑定套接字到本地网卡；
    - 设置套接字的I/O模式为能够接收所有经过本机网卡的数据包。

三、端口扫描技术原理

- 参考代码—TCP首部定义

```
//tcp 首部结构的定义
struct tcphdr
{
    unsigned short      th_sport;
    unsigned short      th_dport;
    unsigned int        th_seq;
    unsigned int        th_ack;
    unsigned char       th_vl:4,th_off:4;
    unsigned char       th_flags;
    #define  TH_FIN     0x01
    #define  TH_SYN     0x02
    #define  TH_RST     0x04
    #define  TH_PSH     0x08
    #define  TH_ACK     0x10
    #define  TH_URG     0x20
    unsigned short      th_win;
    unsigned short      th_cksum;
    unsigned short      th_urg;
};
```

三、端口扫描技术原理

- 参考代码—接受数据包

```
//设置套接字模式和套接字I/O模型
int recv_open(SOCKET  &sniffersock, sockaddr_in  &src)
{
        int flag = 1;
   if(setsockopt(sniffersock,IPPROTO_IP,IP_HDRINCL,(const char*) &flag,
sizeof(int)) != 0)
        {
                cout<<"setsockopt failed,error# "<<WSAGetLastError()<<endl;
                return -1;
        }
   if(bind(sniffersock,(sockaddr*)&src,sizeof(sockaddr_in))==SOCKET_ERROR)
        {
                cout<<"bind failed,error# "<<WSAGetLastError()<<endl;
                return -1;
        }

        DWORD dwValue = 1;
        if(ioctlsocket(sniffersock,SIO_RCVALL,(unsigned long *) &dwValue) != 0)
        {
                cout<<"set SIO_RCVALL failed, error#
"<<WSAGetLastError()<<endl;
                return -1;
        }
        return 0;
}
```

三、端口扫描技术原理

- 秘密扫描（违反三次握手的扫描）
  - 由于这种技术不包含标准的TCP三次握手协议的任何部分，所以无法被记录下来，从而比SYN扫描隐蔽得多。
  - 秘密扫描分类
    - SYN/ACK扫描：发送方发送SYN/ACK分组，目标端口关闭目标机则发送RESET包进行应答，端口打开则忽略分组
    - FIN扫描：发送FIN,如果端口关闭，则目标机对FIN分组应答RESET包，否则忽略此包
    - XMAS扫描：发送一个各个代码位（URG、ACK、PST、RST、SYN和FIN）全为1的分组。如扫描者收到RESET包则说明端口是关闭的，否则端口是开放的
    - NULL扫描：发送各个代码位全等于0的TCP分组，收到RESET包则端口为关闭，否则端口为开放。

三、端口扫描技术原理

- 端口扫描技术对比

	开放扫描	半开放扫描	秘密扫描
优点	简单，不需要特殊的权限，结果可靠	隐蔽性较强，只有少数系统会记录这样的行为	非正常TCP连接过程，通常不记录，隐蔽性最强
缺点	隐蔽性差，服务器通常会记录下客户的连接行为	需要对网络套接字的原始访问，要求系统权限	要求系统权限，对Windows系统无效

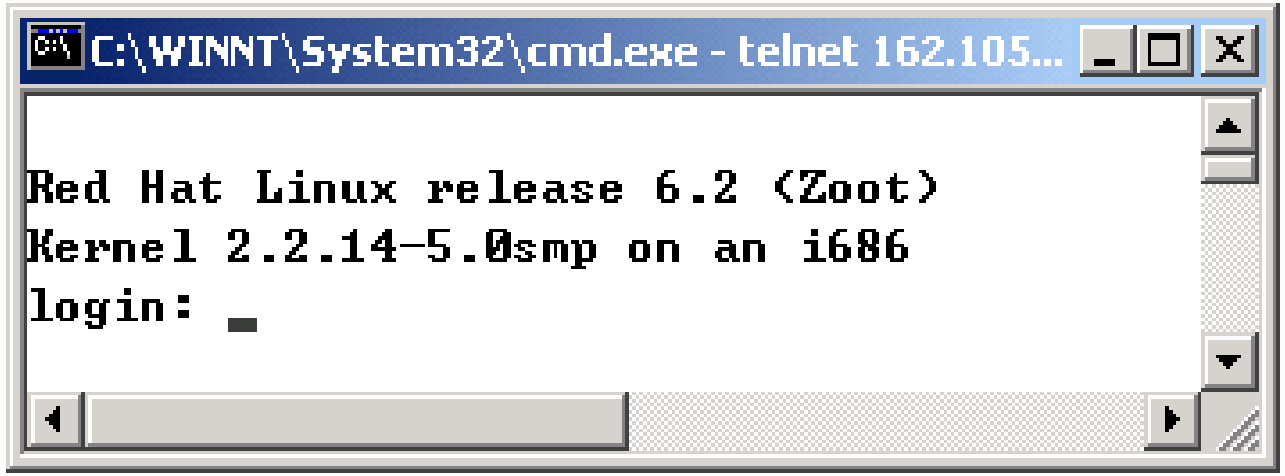
- 测试点5-2
  - 思考题：编制一个端口扫描程序，可以实现对指定IP或指定IP段的主机进行端口扫描。（该思考题为课程实验内容之一，不需要在作业中提交，请查阅资料进行相关的技术准备）

四、操作系统扫描技术原理

- 操作系统扫描的作用
  - 许多安全漏洞都是操作系统特定的，目标主机操作系统类型信息对于入侵举足轻重。
    - 攻击者可以把获取的众多IP地址所提供了TCP和UDP服务信息以及操作系统类型信息保存下来，当发现某操作系统的某安全漏洞或漏洞被发布，可以快速锁定攻击目标
    - 攻击者可以把发现的操作系统漏洞或收集的漏洞保存下来，当发现运行这种操作系统的主机，可以快速锁定攻击目标

四、操作系统扫描技术原理

- 系统探测技术的“手段”（一）
  - 通过系统服务获取标识信息： 在很多探测工具中都使用了此项技术来获得某些服务的标识信息，如Telnet, FTP, HTTP等



四、操作系统扫描技术原理

- 系统探测技术的“手段”（二）
  - 协议栈查询技术：通过测量远程主机的TCP/IP协议栈对不同请求的响应来探测系统。
    - NMAP和QueSO就是基于这种技术的。它们产生一组TCP和UDP请求发送到远程目标主机的开放（未开放）端口。这时，远程主机响应的有用信息就会被探测工具所接收到，然后对其进行分析。

四、操作系统扫描技术原理

- 协议栈查询工作原理
  - TCP/IP规范并不是被严格的执行，每个操作系统中不同的实现将会拥有它们自己的特性，这样就为成功探测带来了可能。
    - 协议规范具有一定的弹性，在某些操作系统中一些选择性的特性被使用，而其他的一些系统则可能没有使用。
    - 协议规范可能被修改，某些对IP协议的自主改进也可能被实现，这就成为了某些操作系统的特性。

四、操作系统扫描技术原理

- 一些特征值在不同操作系统的表现
  - “DF”位：许多操作系统开始在它们发送的包中使用IP”不分片位”以获得好的运行性能；不同操作系统不分片位实现的方式有所不同。
  - TCP初始窗： TCP初始窗只是简单地测试返回包的窗口尺寸。在很多操作系统中是一个常数。例如：AIX是唯一使用0x3F25的操作系统。对于完全重新编写代码的NT 5的TCP堆栈，使用0x402E。
  - ACK值： ACK值不同操作系统具有不同的实现， 如，发送一个FIN|PSH|URG给一个关闭的TCP端口，许多系统将设置ACK等于你的初始序列号，而Windows和某些打印机将发送seq+1；发送SYN|FIN|URG|PSH到打开的端口，不同的Windows系统实现不一致，有些返回seq，有些seq+1，有些返回随机数。

四、操作系统扫描技术原理

- 协议栈查询过程
  - 发送设置了某个选项的数据包，如果目标系统支持，将会进行响应；在返回包中查看哪些选项有值，就可以知道支持什么选项。
    - 通过支持或不支持选项区分不同的系统实现
    - 通过查看不同选项值区分不同的系统实现

四、操作系统扫描技术原理

- 判断主机所用的操作系统一般可以从TCP的一下四个字段着手：
  - TTL
  - Window Size
  - DF
  - TOS
- 通过查看多个特征和组合这些信息可以猜测目标主机操作系统。

四、操作系统扫描技术原理

- 操作系统探测的实例
  - 向目标发送一个信息包，通过嗅探器获取了如下特征:

    04/20-21:41:48.129662 129.142.224.3:659→172.16.1.107:604

    TCP TTL:45 TOS:0x0 ID:56257

    ***F**A*Seq: 0x9DD90553 Ack: 0xE3C65D7 Win:0x7D78

  - TTL: 45 （原始TTL为64，目标系统Linux或FreeBSD系统）
  - Win: 0x7D78（32120 Linux通常使用的默认窗口大小，Linux、FreeBSD、Solaris系统在完整的会话过程中，窗口的大小是维持不变的，部分Cisco路由器和Windows NT窗口是经常改变的）
  - TOS:0x0
  - DF: 设置不分片位（少数系统不使用DF标志）

四、操作系统扫描技术原理

- 但用这种方式识别操作系统类型也有很多的限制，比如：远程系统可以调整这些特征的值来逃避这种检测：
  - 例如用下面的方法来改变TTL值：
    - Solaris: ndd –set /dev/ip ip_def_ttl ‘number’
    - Linux: echo ‘number’ >/proc/sys/net/ipv4/ip_default_ttl
    - NT:HKEY_LOCAL_MACHINE\System\CurentControlSet\Services\Tcpip\Parameters

四、操作系统扫描技术原理

- 系统漏洞扫描
  - 漏洞扫描是用来自动检测远程或本地主机安全漏洞的程序（安全漏洞通常指硬件、软件、协议的具体实现或系统安全策略方面存在的安全缺陷）
  - CVE: Common Vulnerabilities & Exposures
    - CVE是个行业标准，为每个漏洞和暴露确定了惟一的名称和标准化的描述，可以成为评价相应入侵检测和漏洞扫描等工具产品和数据库的基准；
    - CVE类似一个字典表，为广泛认同的信息安全漏洞或者已经暴露出来的弱点给出一个公共的名称。

四、操作系统扫描技术原理

- 漏洞分类：软件、硬件、协议、管理

漏洞类型	漏洞描述
输入验证错误	未对用户的输入进行合法性检查
访问验证错误	访问验证部分本身存在逻辑错误
意外情况处置错误	没有考虑意外情况
设计错误	设计错误造成的
配置错误	系统和应用的配置有误
环境错误	程序在不适当的系统环境下运行造成的

四、操作系统扫描技术原理

- 系统漏洞扫描分类
  - 信息型漏洞探测
    - 大部分网络安全漏洞都与特定的目标状态直接相关，如：目标设备的型号、目标运行的操作系统版本及补丁安装情况、目标的配置情况、运行服务及其服务程序版本等因素。只要对目标的此类信息进行准确探测就可以在很大程序上确定目标存在的安全漏洞。
  - 攻击型漏洞探测
    - 模拟网络入侵的一般过程，对目标系统进行无恶意攻击尝试，若攻击成功则表明相应安全漏洞必然存在。

五、常见扫描工具

- NetCat
- NMAP
- Nessus
- X-Scan
- PortScan

五、常见扫描工具

- Netcat
  - 命令行工具，在Windows和在Linux下的使用方法类似，被设计成一个稳定的后门工具，能够直接由其他程序和脚本轻松驱动。同时，它也是一个功能强大的网络调试和探测工具，能够建立用户需要的几乎所有类型的网络连接。

```
C:\Users\zhaoy>nc -h
[v1.11 NT www.vulnwatch.org/netcat/]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, background mode

        -e prog         inbound program to exec [dangerous!!]
        -g gateway      source-routing hop point[s], up to 8
        -G num          source-routing pointer: 4, 8, 12, ...
        -h              this cruft
        -i secs         delay interval for lines sent, ports scanned
        -l              listen mode, for inbound connects
        -L              listen harder, re-listen on socket close
        -n              numeric-only IP addresses, no DNS
        -o file         hex dump of traffic
        -p port         local port number
        -r              randomize local and remote ports
        -s addr         local source address
        -t              answer TELNET negotiation
        -u              UDP mode
        -v              verbose [use twice to be more verbose]
        -w secs         timeout for connects and final net reads
        -z              zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]
```

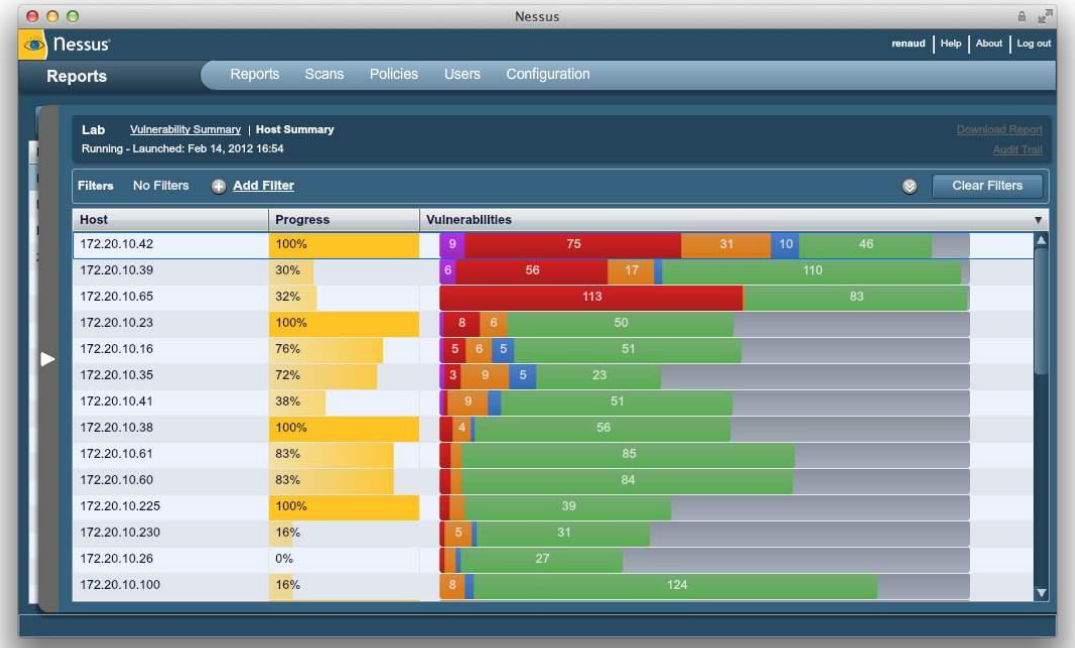
五、常见扫描工具

- NMAP
  - Nmap（Network Mapper，网络映射器）是一款开放源代码的网络探测和安全审核的工具。它的设计目标是快速地扫描大型网络。
    - 发现网络主机
    - 主机提供的服务
    - 服务运行在什么操作系统（包括版本信息）
    - 使用什么类型的报文过滤器/防火墙

五、常见扫描工具

Nessus

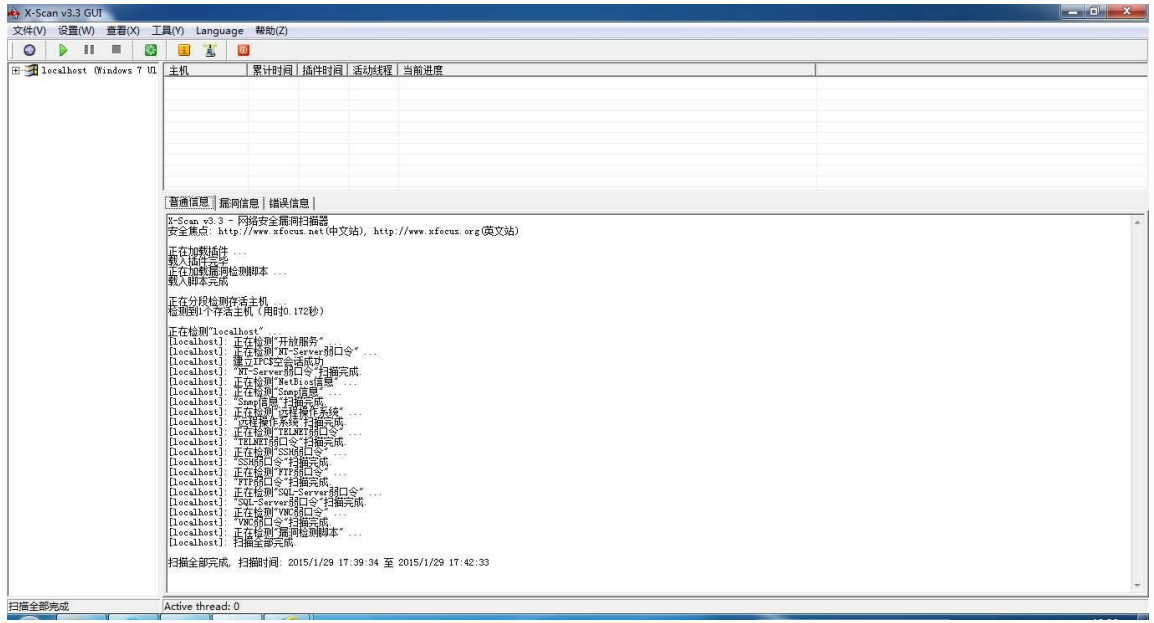
- Nessus是一个功能强大而又易于使用的免费网络漏洞扫描工具。该系统被设计为客户/服务器模式，服务器端负责进行安全扫描，客户端用来配置、管理服务器端，客户端和服务器端之间的通信使用SSL加密。



五、常见扫描工具

X-Scan

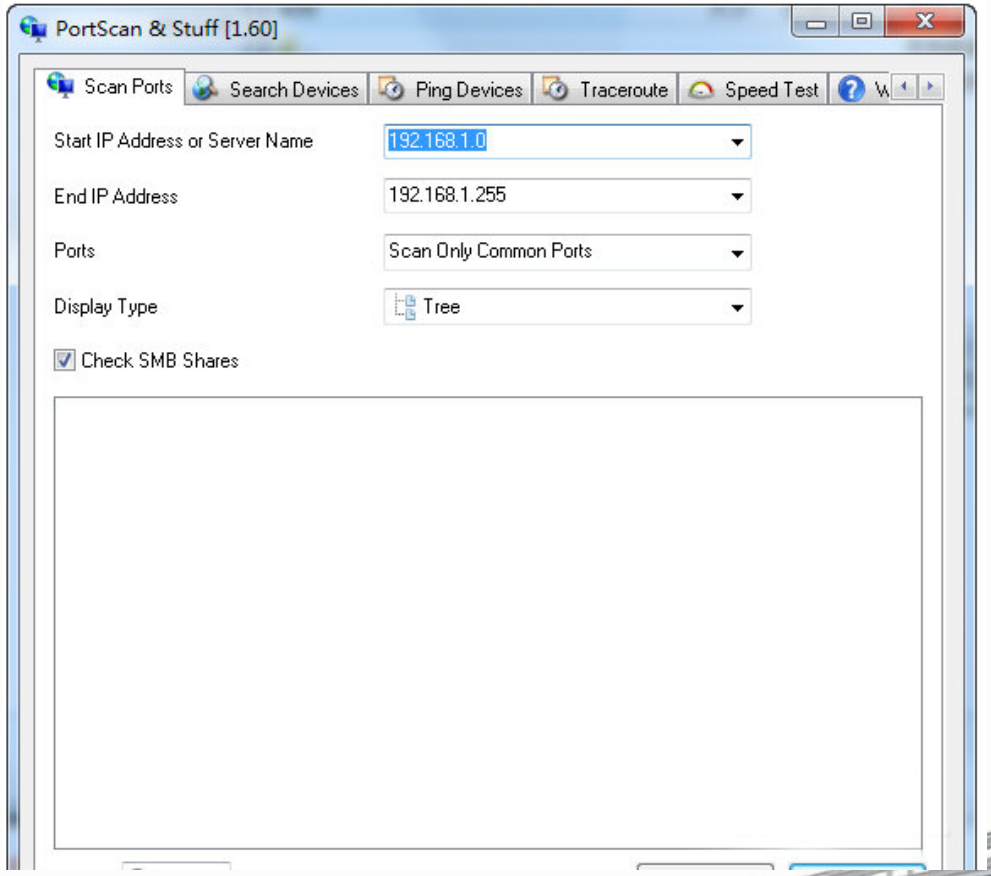
- X-Scan是一款国产免费漏洞扫描工具，运行于Windows操作系统。采用多线程方式对指定IP地址段（或单机）进行安全漏洞检测，具有插件功能，提供了图形界面和命令行两种操作方式。



五、常见扫描工具

PortScan

- PortScan是一款专业的局域网端口扫描器，通过这个端口扫描器可以用于帮助用户扫描目的主机的开放端口，并探测目的主机的操作系统。



结束语

感谢聆听!

特别说明：PPT中所有来自于网络的图片和素材仅用于教学，并保证在未经原作者同意的情况下，不用于任何商业目的。