

# 第2章 扫描与防御技术

---

国家计算机网络入侵防范中心

张玉清



# 本章内容安排

---

- **2.1** 扫描技术概述
- **2.2** 常见的扫描技术
- **2.3** 扫描工具赏析
- **2.4** 扫描的防御
- **2.5** 小结



## 2.1 扫描技术概述

---

- 什么是扫描器
- 网络扫描器是一把双刃剑
- 为什么需要网络扫描器
- 扫描的重要性
- 网络扫描器的主要功能
- 网络扫描器与漏洞的关系
- 扫描三步曲
- 一个典型的扫描案例



# 什么是扫描器

---

- 扫描器是一种自动检测远程或本地主机安全性弱点的程序。
  - 集成了常用的各种扫描技术
  - 能自动发送数据包去探测和攻击远端或本地的端口和服务，并自动收集和记录目标主机的反馈信息
  - 从而发现目标主机是否存活、目标网络内所使用的设备类型与软件版本、服务器或主机上各TCP/UDP端口的分配、所开放的服务、所存在的可能被利用的安全漏洞。
  - 据此提供一份可靠的安全性分析报告，报告可能存在的脆弱性。

# 网络扫描器是一把双刃剑

---

## □ 安全评估工具

系统管理员保障系统安全的有效工具

## □ 网络漏洞扫描器

网络入侵者收集信息的重要手段

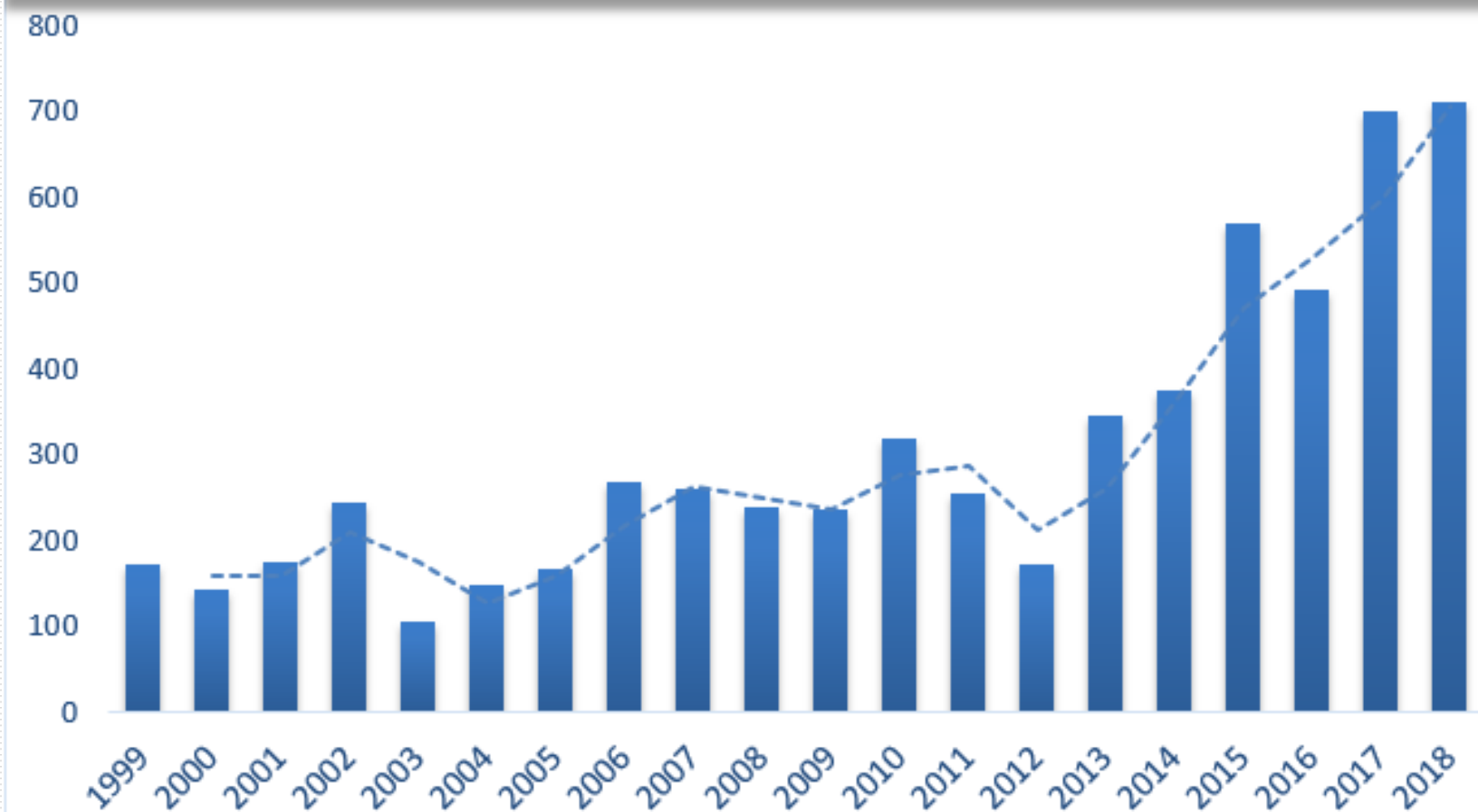
## □ 扫描器是一把“双刃剑”。

# 为什么需要网络扫描器

---

- 由于网络技术的飞速发展，网络规模迅猛增长和计算机系统日益复杂，导致新的系统漏洞层出不穷
  - 例如2018年Windows平台安全漏洞提交数相较过往三年同比上升最高超过40%（见下页图）
- 由于系统管理员的疏忽或缺乏经验，导致旧有的漏洞依然存在
- 许多人出于好奇或别有用心，不停的窥视网上资源

# Windows历年漏洞提交量



数据来源：腾讯安全 <https://s.tencent.com/research/report/641.html>

# 扫描的重要性

---

- 扫描的重要性在于把繁琐的安全检测，通过程序来自动完成。
  - 减轻网络管理员的工作
  - 缩短检测时间
- 同时，也可以认为扫描器是一种网络安全性评估软件，利用扫描器可以快速、深入地为目标网络进行安全评估。
- 网络安全扫描技术与防火墙、安全监控系统互相配合能够为网络提供很高的安全性。



# 网络扫描器的主要功能

---

- ❑ 扫描目标主机识别其工作状态（开/关机）
- ❑ 识别目标主机端口的状态（监听/关闭）
- ❑ 识别目标主机操作系统的类型和版本
- ❑ 识别目标主机服务程序的类型和版本
- ❑ 分析目标主机、目标网络的漏洞（脆弱点）
- ❑ 生成扫描结果报告

# 网络扫描器与漏洞的关系

---

- **网络漏洞**是系统软、硬件存在安全方面的**脆弱性**，安全漏洞的存在导致非法用户入侵系统或未经授权获得访问权限，造成信息篡改、拒绝服务或系统崩溃等问题。
- **网络扫描**可以对计算机网络系统或网络设备进行**安全相关的检测**，以找出安全隐患和可能被黑客利用的漏洞。

# 扫描三步曲

---

- 一个完整的网络安全扫描分为三个阶段：
  - 第一阶段：发现目标主机或网络
  - 第二阶段：发现目标后进一步搜集目标信息
    - 包括操作系统类型、运行的服务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
  - 第三阶段：根据收集到的信息判断或者进一步测试系统是否存在安全漏洞

# 扫描三步曲（续）

- 网络安全扫描技术包括**PING**扫描、操作系统探测、穿透防火墙探测、端口扫描、漏洞扫描等
- **PING**扫描用于扫描**第一阶段**，识别系统是否活动
- **OS**探测、穿透防火墙探测、端口扫描用于扫描**第二阶段**
  - **OS**探测是对目标主机运行的**OS**进行识别
  - 穿透防火墙探测用于获取被防火墙保护的网路资料
  - 端口扫描是通过与目标系统的**TCP/IP**端口连接，并查看该系统处于监听或运行状态的服务
- 漏洞扫描用于安全扫描**第三阶段**，通常是在端口扫描的基础上，进而检测出目标系统存在的安全漏洞



# 一个典型的扫描案例



# 1. Find targets

---

- ❑ 选定目标为: **192.168.1.18**
- ❑ 测试此主机是否处于活动状态, 工具是用操作系统自带的**ping**, 使用命令:  
**ping 192.168.1.18**
- ❑ 结果见下页图。

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Administrator>ping 192.168.1.18

Pinging 192.168.1.18 with 32 bytes of data:

Reply from 192.168.1.18: bytes=32 time<1ms TTL=128
Reply from 192.168.1.18: bytes=32 time<1ms TTL=128
Reply from 192.168.1.18: bytes=32 time<1ms TTL=128
Reply from 192.168.1.18: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.18:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



说明该主机处于活动状态

```
C:\WINDOWS\system32\cmd.exe

Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 192.168.1.18

Pinging 192.168.1.18 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.18:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```



说明该主机处于关机状态或  
数据包被过滤



## 2. Port Scan

---

- 运用扫描工具，检查目标主机开放的端口，判断它运行了哪些服务
- 使用的工具是 Nmap 6.40
- 扫描命令： **nmap** 参数 **IP地址/域名**

```

Nmap scan report for 192.168.209.147
Host is up (0.00069s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.2.2
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x  2 0          0          4096 Feb 19  2013 pub
22/tcp    open  ssh      OpenSSH 5.3 (protocol 2.0)
| ssh-hostkey: 1024 49:79:5e:d0:b7:2b:68:9d:93:f3:45:60:88:bb:c7:a0 (DSA)
|_2048 fc:df:20:91:fd:b5:23:d6:3b:62:6d:8d:df:36:9a:3c (RSA)
23/tcp    open  telnet   Linux telnetd
80/tcp    open  http     Apache httpd 2.2.15 ((CentOS))
| http-methods: GET HEAD POST OPTIONS TRACE
| Potentially risky methods: TRACE
|_See http://nmap.org/nsedoc/scripts/http-methods.html
|_http-title: Apache HTTP Server Test Page powered by CentOS
MAC Address: 00:0C:29:90:43:DE (VMware)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.9
Uptime guess: 0.065 days (since Mon Aug 26 09:13:55 2013)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.69 ms  192.168.209.147

```

端口的开放信息

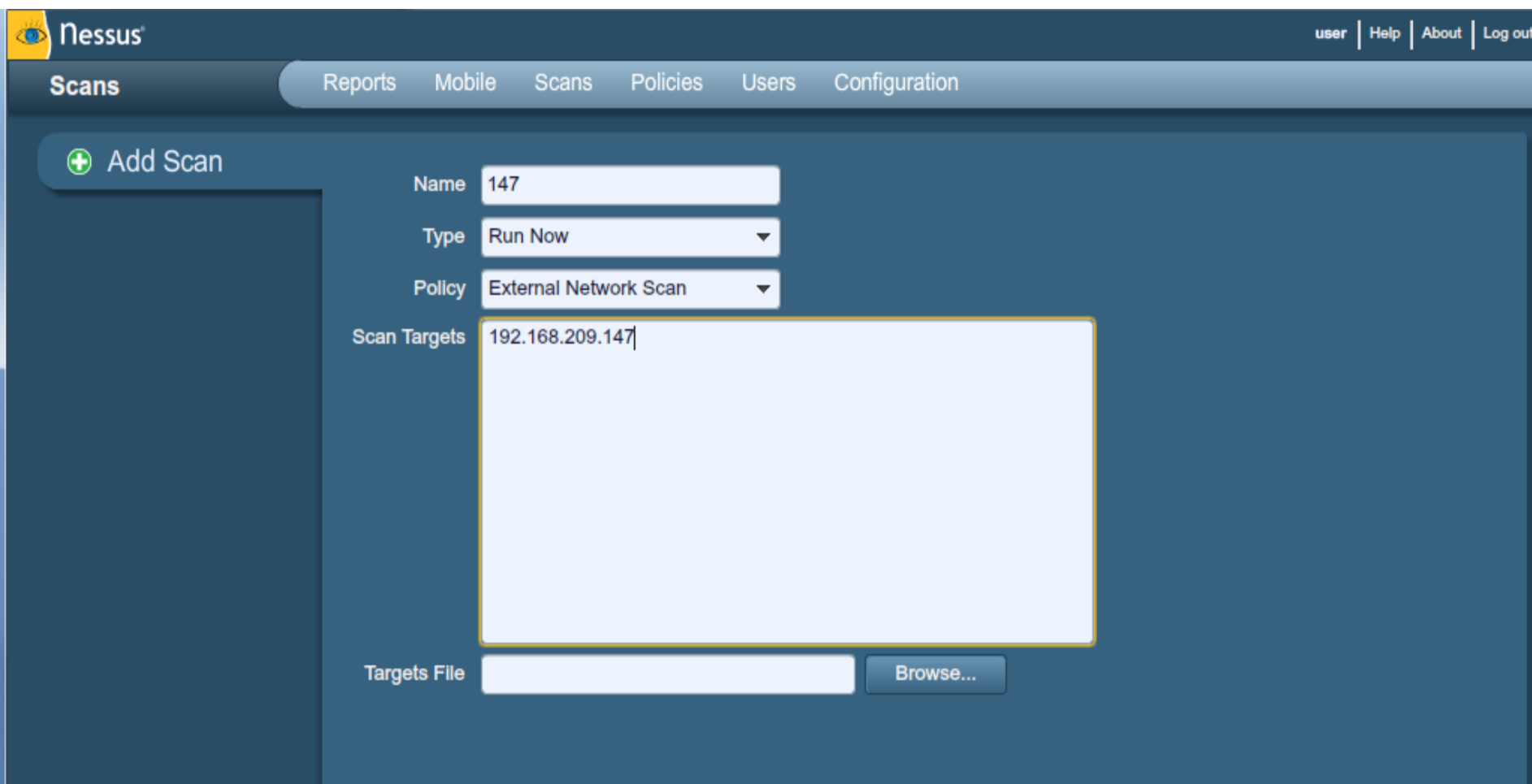
192.168.209.147 的详细信息

# 3. Vulnerability Check

---

- ❑ 检测服务是否存在漏洞
- ❑ 使用漏洞扫描工具 Nessus 5.2.1
- ❑ 扫描过程见下页图

# 扫描目标是192.168.209.147



The image shows the Nessus web interface for configuring a scan. The top navigation bar includes the Nessus logo and links for user, Help, About, and Log out. Below this is a secondary navigation bar with tabs for Reports, Mobile, Scans, Policies, Users, and Configuration. The main content area is titled 'Add Scan' and contains several input fields: 'Name' with the value '147', 'Type' set to 'Run Now', and 'Policy' set to 'External Network Scan'. A large text area for 'Scan Targets' contains the IP address '192.168.209.147'. At the bottom, there is a 'Targets File' input field and a 'Browse...' button.

Nessus

user | Help | About | Log out

Scans | Reports | Mobile | Scans | Policies | Users | Configuration

+ Add Scan

Name: 147

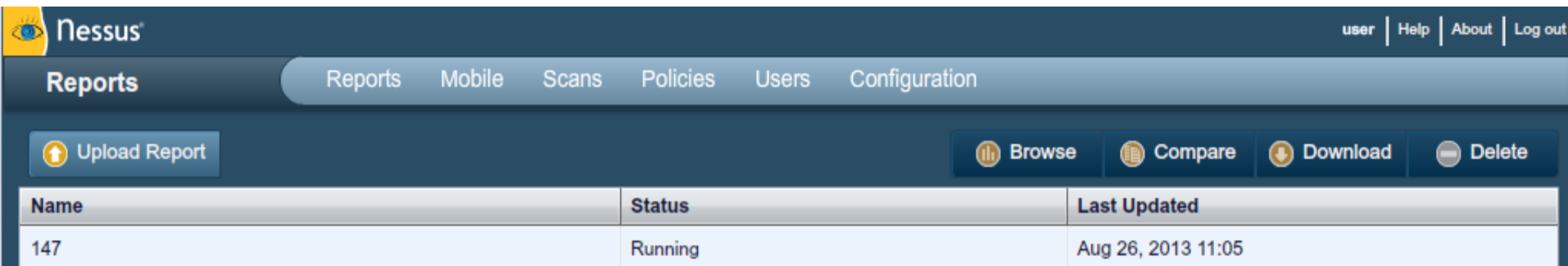
Type: Run Now

Policy: External Network Scan

Scan Targets: 192.168.209.147

Targets File:  Browse...

# Nessus正在进行漏洞扫描



The screenshot displays the Nessus web interface. At the top, the 'Nessus' logo is on the left, and user navigation links ('user', 'Help', 'About', 'Log out') are on the right. Below the header is a navigation bar with tabs for 'Reports', 'Mobile', 'Scans', 'Policies', 'Users', and 'Configuration'. The 'Reports' tab is active. Underneath the navigation bar, there is a toolbar with buttons: 'Upload Report', 'Browse', 'Compare', 'Download', and 'Delete'. The main content area features a table with three columns: 'Name', 'Status', and 'Last Updated'. A single row is visible in the table, representing a scan with ID 147 that is currently 'Running' and was last updated on 'Aug 26, 2013 11:05'.

Name	Status	Last Updated
147	Running	Aug 26, 2013 11:05

## 4. Report

---

- ❑ **Nessus**发现了目标主机的**FTP**服务存在漏洞。
- ❑ 扫描报告中与**FTP**漏洞相关的部分见下页图。

# 漏洞编号: CVE-1999-0497

**147** Vulnerability Summary | [Host Summary](#)  
Completed: Aug 26, 2013 11:05

[Download Report](#)  
[Remove Vulnerability](#) | [Audit Trail](#)

Filters No Filters [+ Add Filter](#) [Clear Filters](#)

Plugin ID	Count	Host	Port
10079	1	192.168.209.147	21 / tcp
11213	1		
34324	1		
42263	1		
11219	4		
22964	4		
10092	1		
10107	1		
10114	1		
10267	1		
10281	1		
10287	1		
10881	1		
11032	1		
11936	1		
18261	1		
19506	1		
20094	1		
24260	1		
25220	1		
39520	1		

**Plugin ID:** 10079 **Port / Service:** ftp (21/tcp) **Severity:** Medium ×  
**Plugin Name:** Anonymous FTP Enabled  
**Synopsis:** Anonymous logins are allowed on the remote FTP server.  
**Description**  
This FTP service allows anonymous logins. Any remote user may connect and authenticate without providing a password or unique credentials. This allows a user to access any files made available on the FTP server.  
**Solution**  
Disable anonymous FTP if it is not required. Routinely check the FTP server to ensure sensitive content is not available.  
**Risk Factor:** Medium  
**CVSS Base Score**  
5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)  
**Plugin Output**  
The contents of the remote FTP root are :  
drwxr-xr-x 2 0 0 4096 Feb 19 2013 pub  
**CVE**  
[CVE-1999-0497](#)  
**Cross-References**  
[OSVDB:69](#)  
**Vulnerability Publication Date:** 1993/07/01  
**Plugin Publication Date:** 1999/06/22  
**Plugin Last Modification Date:** 2013/01/25

## 2.2 常见的扫描技术

- **TCP/IP**相关知识
- 常用网络命令
- 主机扫描
- 端口扫描
  - 全扫描
  - 半扫描
  - 秘密扫描
  - 认证(ident)扫描
  - FTP代理扫描
- 远程主机**OS**指纹识别
- 漏洞扫描

不可不学的扫描技术  
巧妙奇特的天才构思



# TCP/IP相关知识

---

- **TCP**报文格式
- **TCP**通信过程
- **ICMP**协议

# TCP报文格式

源端口（ 16 位 ）								目的端口（ 16 位 ）							
顺序号（ 32 位 ）															
确认号（ 32 位 ）															
TCP 头长 （ 4 位 ）	保留位 （ 6 位 ）	U R G	A C K	P S H	R S T	S Y N	F I N	窗口大小（ 16 位 ）							
校验和（ 16 位 ）								紧急指针（ 16 位 ）							
可选项（ 0 或更多的 32 位字 ）															
数据（可选项）															

# TCP控制位

---

## □ **URG:** 紧急数据标志。

- 如果它为1，表示本数据包中包含紧急数据。此时紧急数据指针有效。

## □ **ACK:** 确认标志位。

- 如果为1，表示包中的确认号是有效的。否则，包中的确认号无效。

## □ **PSH:** 如果置位，接收端应尽快把数据传送给应用层，而不再等到整个缓存都填满了后再向上传送。

# TCP控制位

---

□ **RST:** 用来复位一个连接。

- RST标志置位的数据包称为复位包。一般情况下，如果TCP收到的一个分段明显不是属于该主机上的任何一个连接，则向远端发送一个复位包。

□ **SYN:** 标志位用来建立连接，让连接双方同步序列号。

- 如果SYN=1而ACK=0，则表示该数据包为连接请求
- 如果SYN=1而ACK=1，则表示接受连接。

□ **FIN:** 表示发送端已经没有数据要求传输了，希望释放连接。

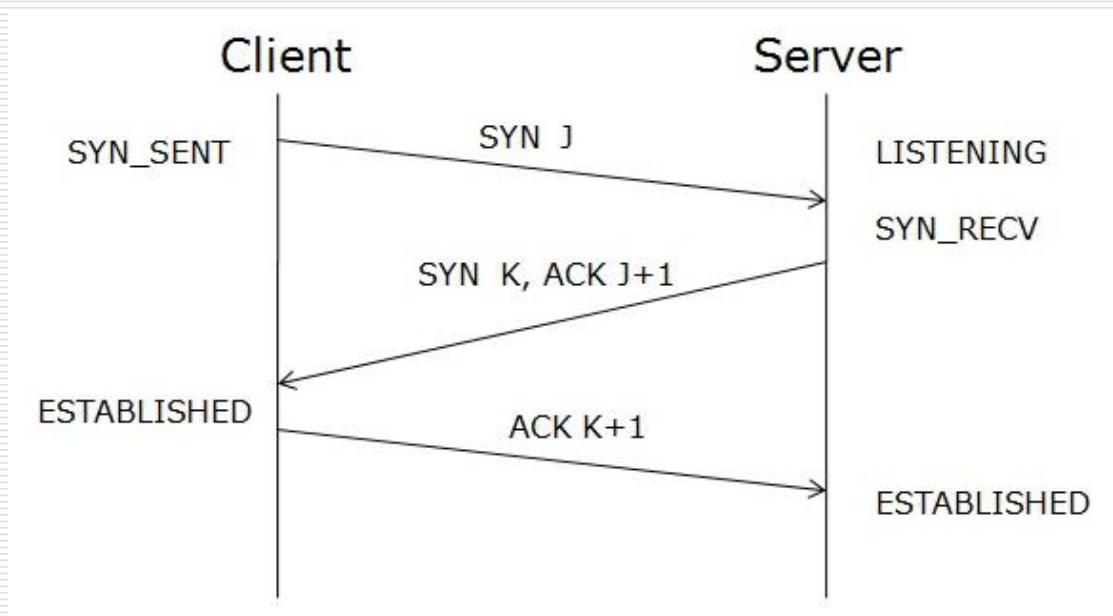
# TCP通信过程

---

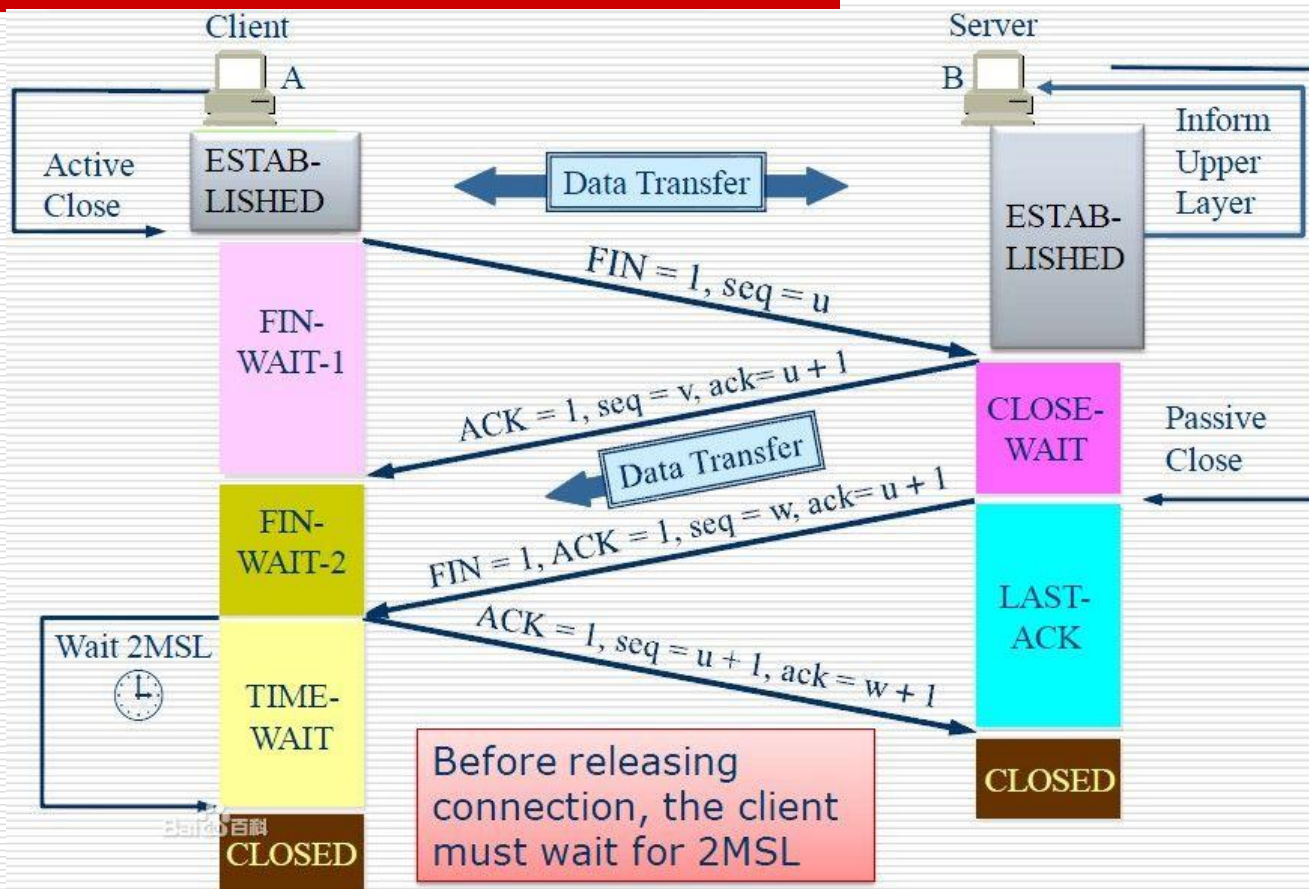
## □ 正常TCP通信过程:

- 建立连接
- (数据传输)
- 断开连接

# 建立TCP连接(三次握手)



# 断开TCP连接



**MSL: 最长报文寿命 (Maximum Segment Lifetime)** 图片来自百度百科

# ICMP协议（1）

---

- **Internet Control Message Protocol**, 是**IP**的一部分，在**IP**协议栈中必须实现。
- 用途：
  - 网关或者目标机器利用**ICMP**与源通讯
  - 当出现问题时，提供反馈信息用于报告错误
- 特点：
  - 其控制能力并不用于保证传输的可靠性
  - 它本身也不是可靠传输的
  - 并不用来反映**ICMP**报文的传输情况



# ICMP协议（2）

---

## 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
- 16 Information Reply
- 17 Address Mask Request
- 18 Address Mask Reply

# 常用网络命令

---

- **Ping**
- **Traceroute、Tracert**
- **Net命令系列**

# 常用网络命令--ping

---

- **Ping**是最基本的扫描技术。
- **ping**命令——主要目的是**检测目标主机是不是可连通**，继而探测一个**IP**范围内的主机是否处于激活状态。

不要小瞧ping  
黑客的攻击往往都是从ping开始的

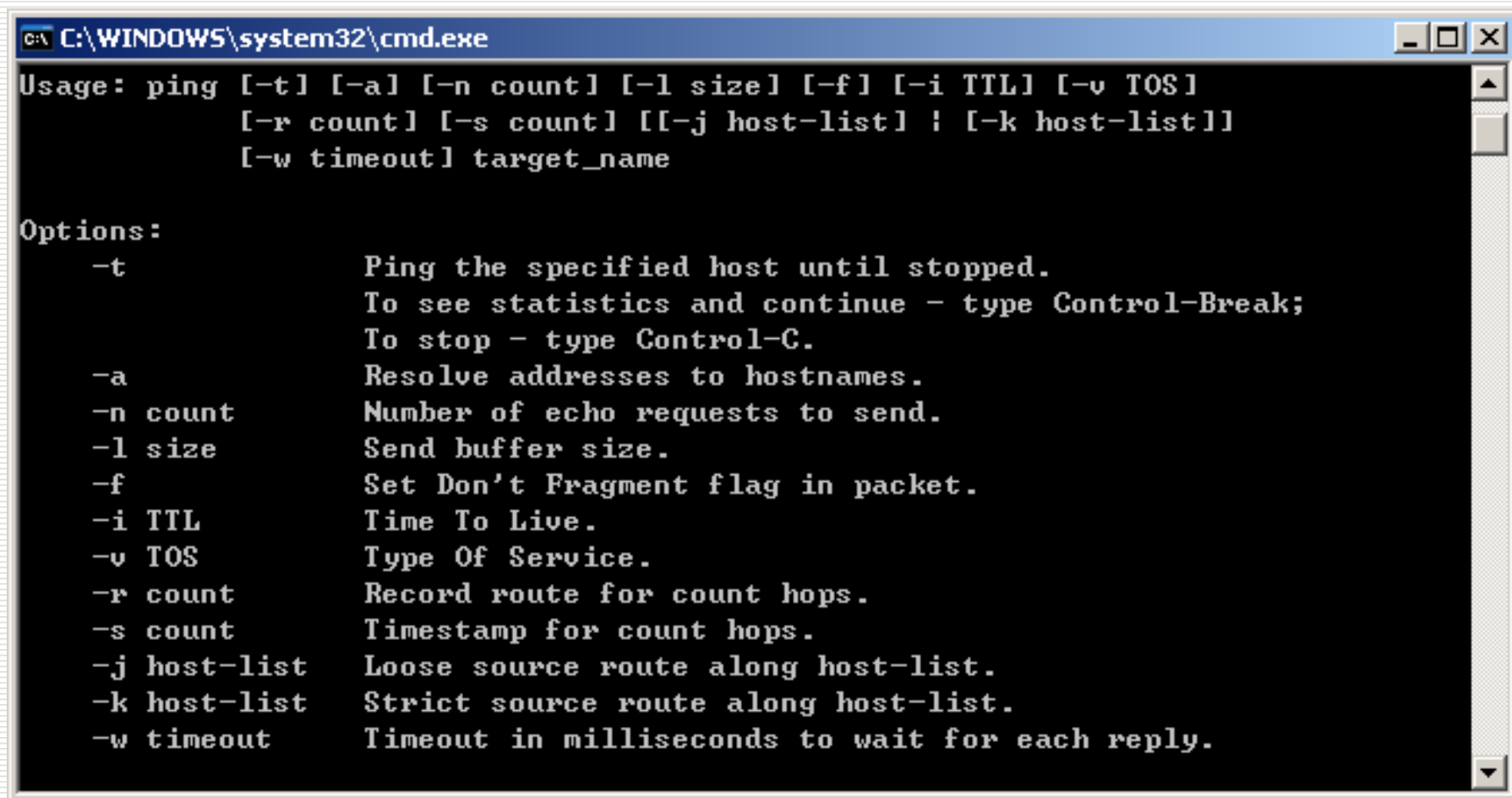
# 常用网络命令--ping的原理

---

- **ping**是一个基本的网络命令，用来确定网络上具有某个特定**IP**地址的主机是否存在以及是否能接收请求。
- **Ping**命令通过向计算机发送**ICMP**回应报文并且监听回应报文的返回，以校验与远程计算机或本地计算机的连接。

# 常用网络命令--ping参数说明

❑ ping在安装了TCP/IP协议后可以使用。



```
C:\WINDOWS\system32\cmd.exe

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] : [-k host-list]]
          [-w timeout] target_name

Options:
    -t                Ping the specified host until stopped.
                     To see statistics and continue - type Control-Break;
                     To stop - type Control-C.
    -a                Resolve addresses to hostnames.
    -n count           Number of echo requests to send.
    -l size            Send buffer size.
    -f                Set Don't Fragment flag in packet.
    -i TTL             Time To Live.
    -v TOS             Type Of Service.
    -r count           Record route for count hops.
    -s count           Timestamp for count hops.
    -j host-list       Loose source route along host-list.
    -k host-list       Strict source route along host-list.
    -w timeout         Timeout in milliseconds to wait for each reply.
```

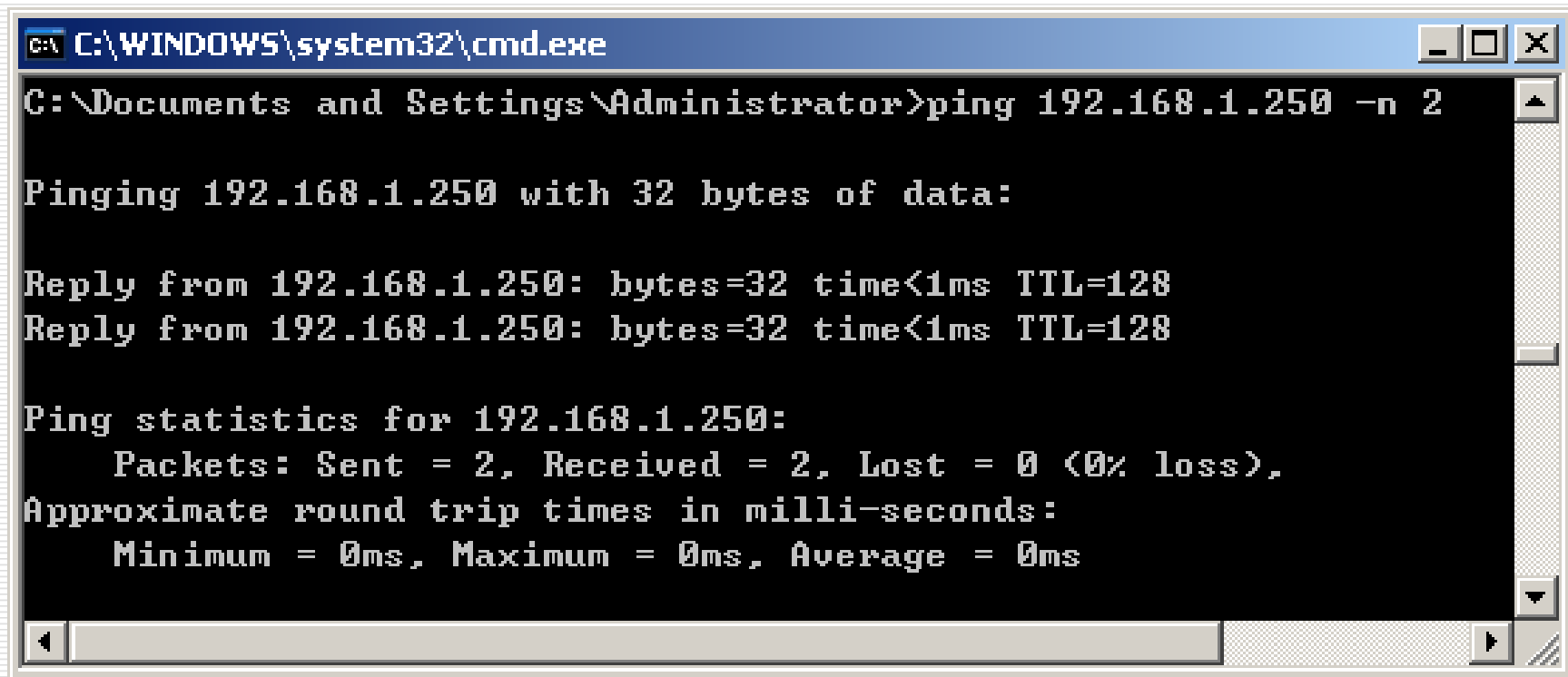
用法: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]  
[-r count] [-s count] [[-j host-list] | [-k host-list]]  
[-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]  
[-4] [-6] target\_name

### 选项:

-t	Ping 指定的主机，直到停止。 若要查看统计信息并继续操作，请键入 Ctrl+Break； 若要停止，请键入 Ctrl+C。
-a	将地址解析为主机名。
-n count	要发送的回显请求数。
-l size	发送缓冲区大小。
-f	在数据包中设置“不分段”标记(仅适用于 IPv4)。
-i TTL	生存时间。
-v TOS	服务类型(仅适用于 IPv4。该设置已被弃用， 对 IP 标头中的服务类型字段没有任何 影响)。
-r count	记录计数跃点的路由(仅适用于 IPv4)。
-s count	计数跃点的时间戳(仅适用于 IPv4)。
-j host-list	与主机列表一起使用的松散源路由(仅适用于 IPv4)。
-k host-list	与主机列表一起使用的严格源路由(仅适用于 IPv4)。
-w timeout	等待每次回复的超时时间(毫秒)。
-R	同样使用路由标头测试反向路由(仅适用于 IPv6)。 根据 RFC 5095，已弃用此路由标头。 如果使用此标头，某些系统可能丢弃 回显请求。
-S srcaddr	要使用的源地址。
-c compartment	路由隔离舱标识符。
-p	Ping Hyper-V 网络虚拟化提供程序地址。
-4	强制使用 IPv4。

# 常用网络命令—ping命令使用

- **-n**: 发送**ICMP**回应报文的个数



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>ping 192.168.1.250 -n 2

Pinging 192.168.1.250 with 32 bytes of data:

Reply from 192.168.1.250: bytes=32 time<1ms TTL=128
Reply from 192.168.1.250: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.250:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

# 常用网络命令-- Traceroute

- ❑ **Traceroute**——跟踪两台机器之间的路径，显示中间的每一个节点的信息。这个工具可以用来确定某个主机的位置。
- ❑ **traceroute** 命令旨在用于网络测试、评估和管理。它应主要用于手动故障隔离。
- ❑ 语法：

操作	命令
TraceRoute	<code>traceroute [-f first_TTL] [-m max_TTL] [-p port] [-q nqueries] [-w timeout] host</code>



# 常用网络命令– Traceroute说明

---

- ❑ **-f** **-f**后指定一个初始**TTL**，它的范围是大于**0**小于最大**TTL**，缺省为**1**。
- ❑ **-m** **-m**后指定一个最大**TTL**，它的范围是大于初始**TTL**，缺省为**30**。
- ❑ **-p** **-p**后可以指定一个整数，该整数是目的主机的端口号，它的缺省为**33434**，用户一般无须更改此选项。
- ❑ **-q** **-q**后可以指定一个整数，该整数是每次发送的探测数据包的个数，它的范围是大于**0**，缺省为**3**。
- ❑ **-w** **-w**后可以指定一个整数，该整数指明**IP**包的超时时间，它的范围是大于**0**，缺省为**5秒**。
- ❑ **host** 目的主机的**IP**地址

# 常用网络命令– Traceroute示例

---

```
Quid # traceroute 35.1.1.48
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 56byte packet
1 helios.ee.lbl.gov (128.3.112.1) 19 ms 19 ms 0 ms
2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
3 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39ms 40ms 39ms
4 ccn-nerif22.Berkeley.EDU (128.32.168.22) 39 ms 39 ms 39 ms
5 128.32.197.4 (128.32.197.4) 40 ms 59 ms 59 ms
6 131.119.2.5 (131.119.2.5) 59 ms 59 ms 59 ms
7 129.140.70.13 (129.140.70.13) 99 ms 99 ms 80 ms
8 129.140.71.6 (129.140.71.6) 139 ms 239 ms 319 ms
9 129.140.81.7 (129.140.81.7) 220 ms 199 ms 199 ms
10 nic.merit.edu (35.1.1.48) 239 ms 239 ms 239 ms
```

可以看出从源主机到目的地都经过了哪些网关，这对于网络分析是非常有用的。

# 常用的网络命令—Tracert、x-firewalk

---

- ❑ **Windows**下用**tracert**命令可以查看路由信息，但是如今的路由器大部分都对**tracert**命令做了限制，此命令已经没有任何效果。
- ❑ 有黑客开发出**x-firewalk.exe**可用于在**Windows**环境下查看路由信息，非常实用。

# 常用的网络命令—x-firewalk示例

- ❑ 命令: **x-firewalk www.163.com**
- ❑ 可以看到本地到达**www.163.com**都经过了哪些路由器

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator\桌面\x-firewalk>x-firewalk www.163.com

Tracing route to www.163.com [202.108.9.38] by ICMP
over No.0 netWork interface with ip [192.168.1.34]
over a maximum of 32 hops, timeout in 3000 milliseconds:

 1      <10 ms   16 ms   <10 ms   192.168.1.254      [局域网 对方和您在同一内部网]
 2      31 ms   31 ms   16 ms    61.148.123.41     [北京市 网通ADSL]
 3      31 ms   32 ms   15 ms    61.148.7.157      [北京市 网通]
 4      31 ms   16 ms   31 ms    61.148.3.81       [北京市 网通]
 5      31 ms   16 ms   31 ms    202.106.192.226   [北京市 路由器]
 6      15 ms   31 ms   16 ms    61.148.143.26     [北京市 网通ADSL]
 7      31 ms   32 ms   15 ms    210.74.176.194    [北京市 CZ88.NET]
 8      31 ms   31 ms   32 ms    202.108.9.38      [北京市 网通]

Trace complete.
```

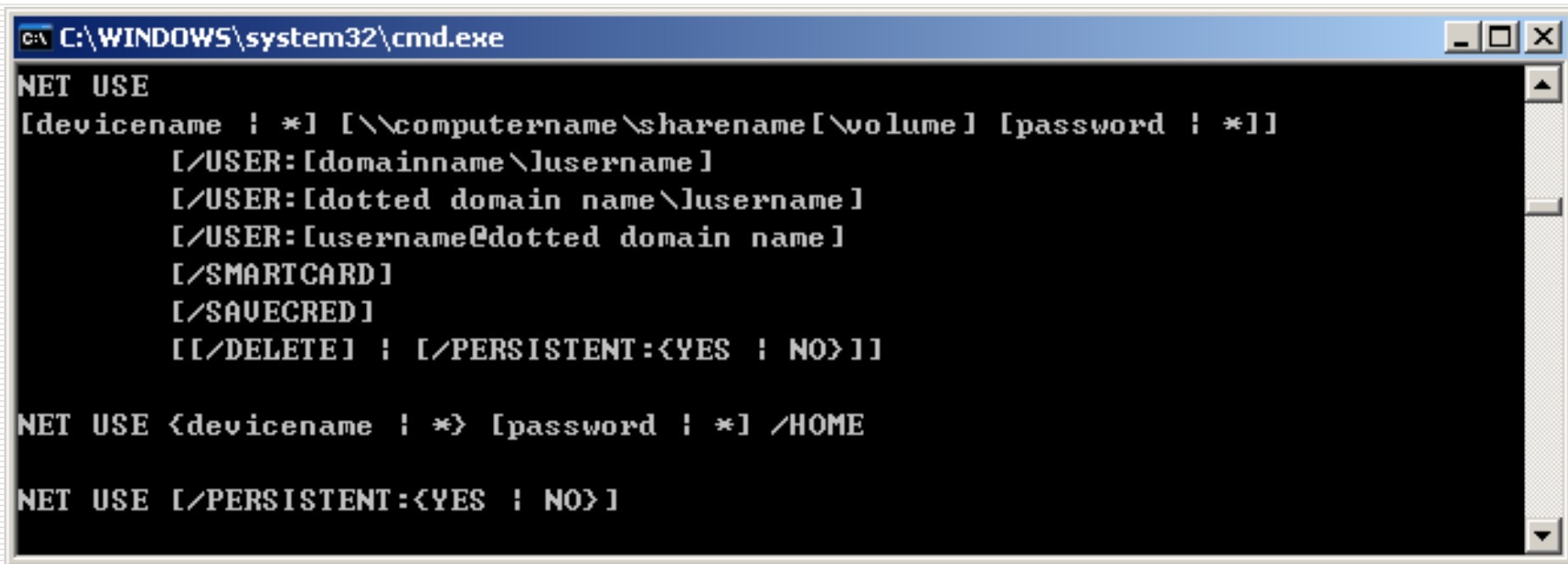
# 常用网络命令--net

---

- **Net命令系列**——很多的**Windows**的网络命令都是**net**开头的。利用**net**开头的命令，可以实现很多的网络管理功能.....
  - 比如用net start server，可以启动服务器；
  - net use 用于将计算机与共享的资源相连接，或者切断计算机与共享资源的连接，当不带选项使用本命令时，它会列出计算机的连接。
- 以下以**Windows NT**下的**Net USE**命令为例。

# 常用网络命令—net use示例

## net use命令及参数使用



```
C:\WINDOWS\system32\cmd.exe

NET USE
[devicename : *] [\computername\sharename[\volume] [password : *]]
    [/USER:[domainname\username]
    [/USER:[dotted domain name\username]
    [/USER:[username@dotted domain name]
    [/SMARTCARD]
    [/SAVECRED]
    [[/DELETE] : [/PERSISTENT:<YES | NO>]]

NET USE <devicename : *> [password : *] /HOME

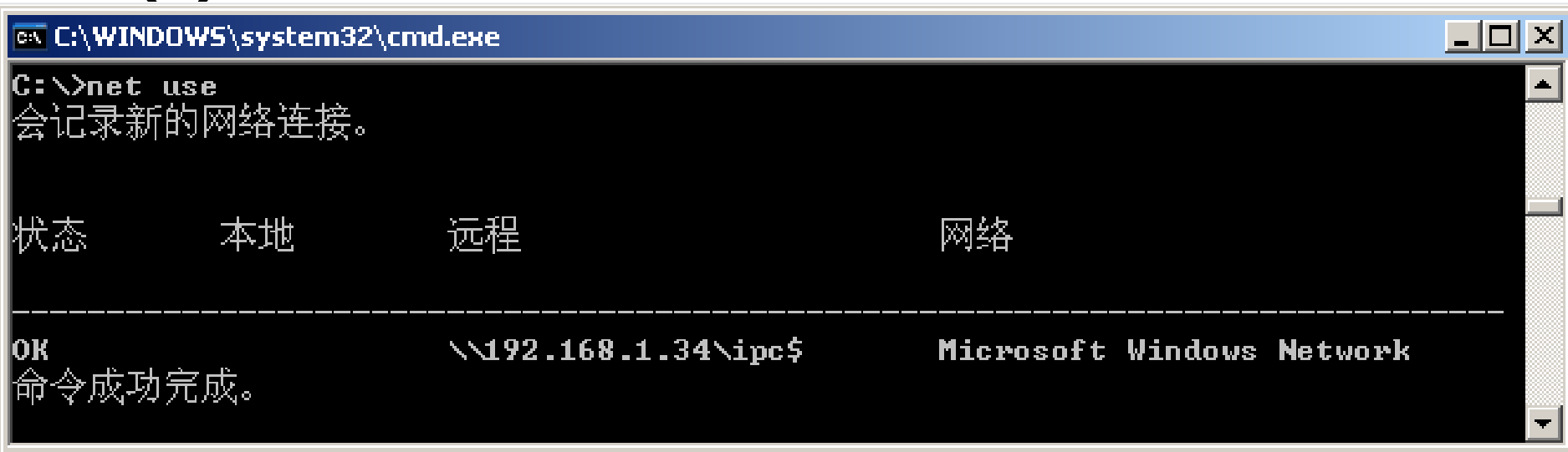
NET USE [/PERSISTENT:<YES | NO>]
```

# 常用网络命令—net use示例

(1) 用户名为**Administrator**，密码为**longmang**与远程计算机**192.168.1.34**进行**IPC\$**连接，如图所示。

```
C:\>net use \\192.168.1.34\ipc$ longmang /user:Administrator
命令成功完成。
```

(2) 查看与远程计算机建立的连接，如图所示。



```
C:\WINDOWS\system32\cmd.exe
C:\>net use
会记录新的网络连接。

状态          本地          远程          网络
-----
OK            \\192.168.1.34\ipc$  Microsoft Windows Network
命令成功完成。
```

# 常用网络命令—net use示例

**(3)**将远程计算机的c盘映射到本地o盘，如图所示。



```
C:\WINDOWS\system32\cmd.exe
C:\>net use o: \\192.168.1.34\c$
命令成功完成。

C:\>o:

O:\>dir
驱动器 o 中的卷没有标签。
卷的序列号是 F4E7-4C6E

O:\ 的目录

2006-08-20  18:00                0 AUTOEXEC.BAT
2007-06-29  11:30             <DIR>          chenh2
2006-08-20  18:00                0 CONFIG.SYS
2006-08-20  17:40             <DIR>          Documents and Settings
2007-08-23  15:33             <DIR>          Downloads
2007-06-22  09:34             <DIR>          Drivers
2007-08-10  15:46             <DIR>          Program Files
2007-08-27  15:42             <DIR>          WINDOWS
                2 个文件                0 字节
                6 个目录 5,914,083,328 可用字节
```



# 常用网络命令—net use示例

## (4)删除一个IPC\$连接



```
C:\WINDOWS\system32\cmd.exe

C:\>net use \\192.168.1.34\ipc$ /delete
\\192.168.1.34\ipc$ 已经删除。
```

## (5)删除共享映射



```
C:\WINDOWS\system32\cmd.exe

C:\>net use o: /delete
o: 已经删除。
```

# 主机扫描技术

---

- 传统技术
- 高级技术

# 主机扫描技术—传统技术

---

- 主机扫描的目的是**确定在目标网络上的主机是否可达**。这是信息收集的初级阶段，其效果直接影响到后续的扫描。
- 常用的传统扫描手段有：
  - ICMP Echo扫描
  - ICMP Sweep扫描
  - Broadcast ICMP扫描
  - Non-Echo ICMP扫描

# ICMP echo扫描

---

- ❑ 实现原理：**Ping**的实现机制，在判断在一个网络上主机是否开机时非常有用。
  - 向目标主机发送ICMP Echo Request (type 8)数据包，等待回复的ICMP Echo Reply 包(type 0)。
  - 如果能收到，则表明目标系统可达，否则表明目标系统已经不可达或发送的包被对方的设备过滤掉。
- ❑ 优点：简单，系统支持
- ❑ 缺点：很容易被防火墙限制
- ❑ 可以通过并行发送，同时探测多个目标主机，以提高探测效率（**ICMP Sweep**扫描）。

# ICMP sweep扫描

---

- ❑ 使用**ICMP ECHO**轮询多个主机称为**ICMP SWEEP(或者Ping Sweep)**。
- ❑ 对于小的或者中等网络使用这种方法来探测主机是一种比较可接受的行为，但对于一些大的网络如**CLASS A、B**，这种方法就显的比较慢，原因是**Ping**在处理下一个之前将会等待正在探测主机的回应。
- ❑ 扫描工具**Nmap**实现了**ICMP sweep**的功能。

# Broadcast ICMP扫描

---

- ❑ 实现原理：将**ICMP**请求包的目标地址设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。
- ❑ 缺点：
  - 只适合于UNIX/Linux系统，Windows 会忽略这种请求包；
  - 这种扫描方式容易引起广播风暴

# Non-Echo ICMP扫描

---

- 一些其它**ICMP**类型包也可以用于对主机或网络设备的探测，如：
  - Stamp Request (Type 13)
  - Reply (Type 14)
  - Information Request (Type 15)
  - Reply (Type 16)
  - Address Mask Request (Type 17)
  - Reply (Type 18)

# 主机扫描技术—高级技术

---

- 防火墙和网络过滤设备常常导致传统的探测手段变得无效。为了突破这种限制，必须采用一些非常规的手段，利用**ICMP**协议提供网络间传送错误信息的手段，往往可以更有效的达到目的：
  - 异常的IP包头
  - 在IP头中设置无效的字段值
  - 错误的数据分片
  - 通过超长包探测内部路由器
  - 反向映射探测



# 端口扫描技术

---

- **TCP/IP**协议提出的**端口**是网络通信进程与外界通讯交流的出口，可被命名和寻址，可以认为是**网络通信进程的一种标识符**。
  - 进程通过系统调用与某端口建立连接**绑定**后，便会**监听**这个端口，传输层传给该端口的数据都被相应进程所接收，而相应进程发给传输层的数据都从该端口输出。
  - 互联网上的通信双方不仅需要知道对方的**IP**地址，也需要知道通信程序的端口号。
-

# 端口扫描技术

---

- 目前**IPv4**协议支持**16**位的端口，端口号范围是**0**～**65535**。
    - 0～1023号端口称为熟知端口，被提供给特定的服务使用
    - 1024～49151号端口称为注册端口，由IANA记录和追踪
    - 49152～65535号端口称为动态端口或专用端口，提供给专用应用程序。
  - 许多常用的服务使用的是标准的端口，只要扫描到相应的端口，就能知道目标主机上运行着什么服务。端口扫描技术就是利用这一点向目标系统的**TCP/UDP**端口发送探测数据包，记录目标系统的响应，通过分析响应来查看该系统处于监听或运行状态的服务。
-

# 端口扫描技术

---

- 当确定了目标主机可达后，就可以使用端口扫描技术，发现目标主机的开放端口，包括网络协议和各种应用监听的端口。端口扫描技术包括以下几种：
- **全扫描**
  - 会产生大量的审计数据，容易被对方发现，但其可靠性高。
- **半扫描**
  - 隐蔽性和可靠性介于全扫描和秘密扫描之间。
- **秘密扫描**
  - 能有效的避免对方入侵检测系统和防火墙的检测，但使用的数据包在通过网络时容易被丢弃从而产生错误的探测信息。
- **认证(ident)扫描**
  - 需要先建立一个完整的TCP连接。
- **FTP代理扫描**
  - 隐蔽性好，难以追踪。但受到服务器设置的限制。

# 全扫描

---

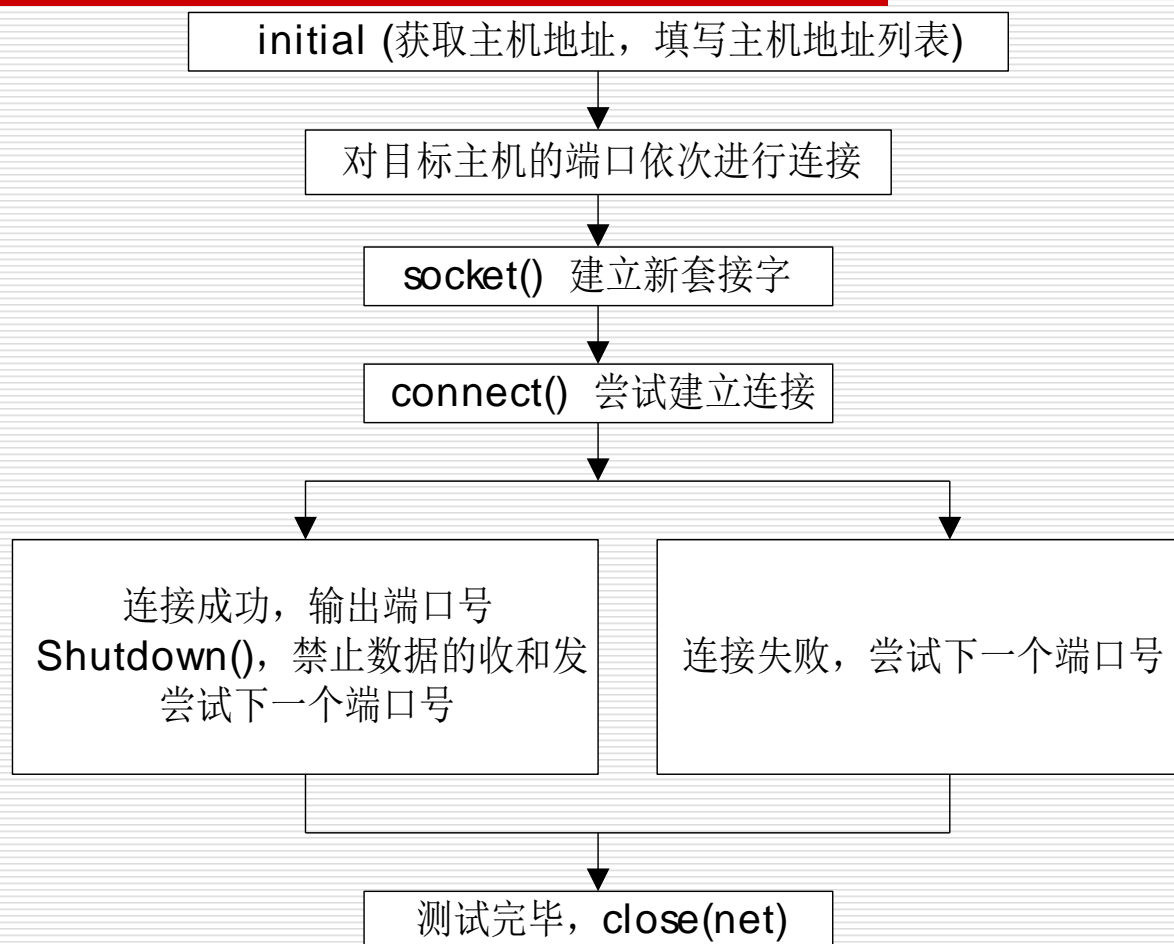
- 全扫描原理
- 全扫描过程
- 全扫描特点

# 全扫描——全扫描原理

---

- 全**TCP**连接是**TCP**端口扫描的基础。
- 扫描主机尝试（使用三次握手）与目标主机的某个端口建立正规的连接。
- 连接由系统调用**connect()**开始。
  - 如果端口**开放**，则连接将建立成功；
  - 否则，返回**-1**，则表示端口**关闭**。

# 全扫描——全扫描过程（流程图）



# 全扫描——全扫描过程（成功）

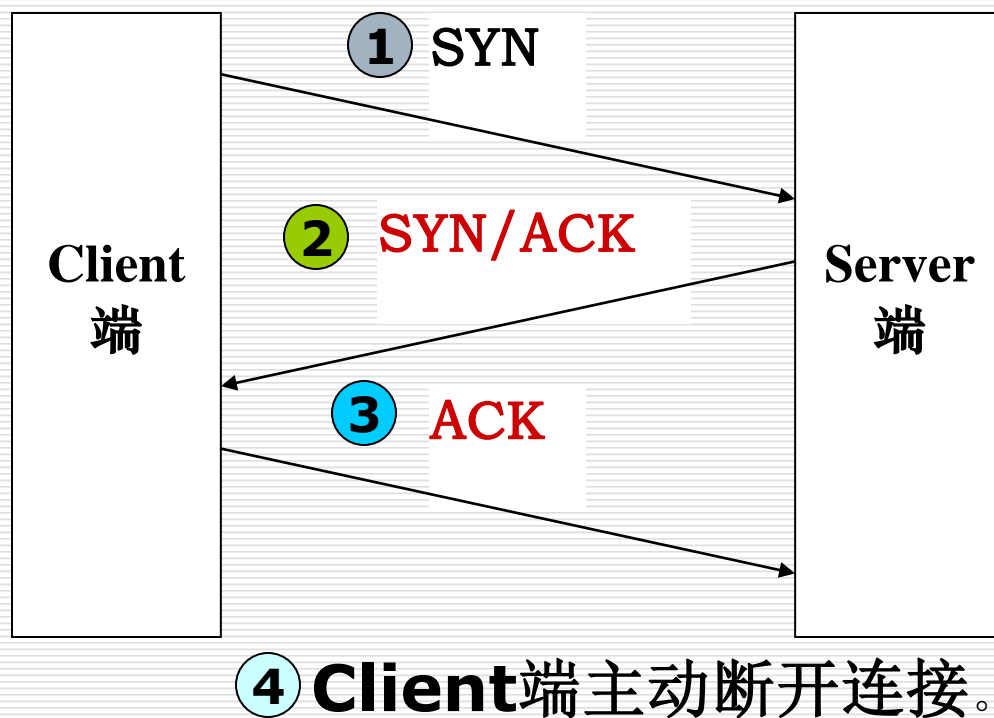
---

□ **TCP Connect**端口扫描服务端与客户端建立连接成功（目标端口开放）的过程：

- （1）**Client**端发送**SYN**；
- （2）**Server**端返回**SYN/ACK**，表明端口开放；
- （3）**Client**端返回**ACK**，表明连接已建立；
- （4）**Client**端主动断开连接。

# 全扫描——全扫描过程（成功）

□ 建立连接成功（目标端口开放）如图所示：





# 全扫描——全扫描过程（未成功）

---

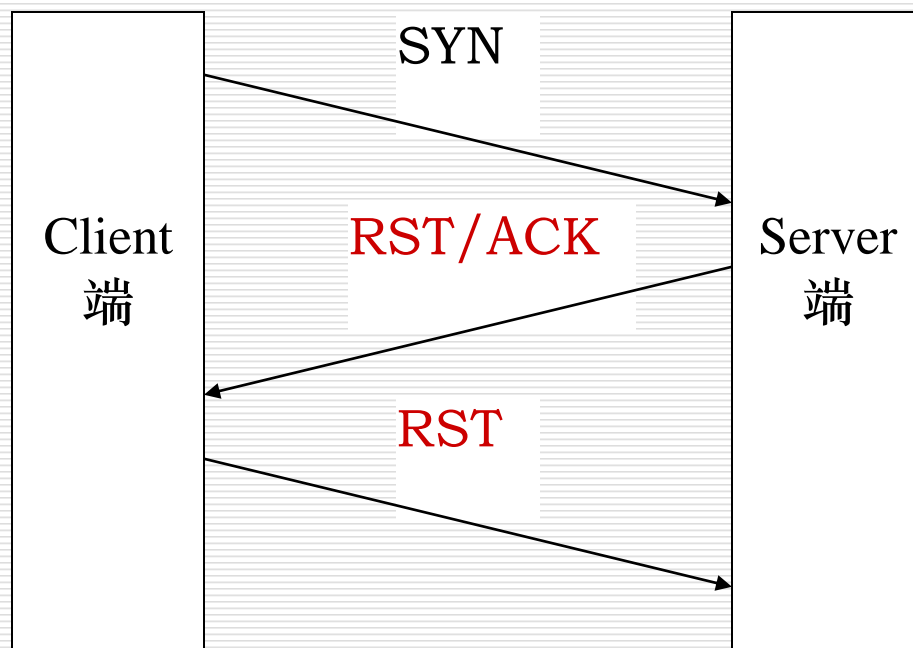
□ **TCP Connect**端口扫描服务端与客户端未建立连接成功（目标端口关闭）过程：

（1）**Client**端发送**SYN**；

（2）**Server**端返回**RST/ACK**，表明端口未开放。

# 全扫描——全扫描过程（未成功）

□ 未建立连接成功(目标端口关闭)如图所示：



# 全扫描——全扫描特点（优点）

---

- 优点是**实现简单**，对操作者的权限没有严格要求（有些类型的端口扫描需要操作者具有**root**权限），系统中的任何用户都有权力使用这个调用。
- 另一优点是**扫描速度快**。如果对每个目标端口以线性的方式，使用单独的**connect()**调用，可以通过同时打开多个套接字，从而加速扫描。

# 全扫描——全扫描特点（缺点）

---

- 扫描方式不隐蔽
- 这种扫描方法很容易被检测出来，在日志文件中会有大量密集的连接和错误记录
- 容易被防火墙发现和屏蔽

# 半扫描

---

- **TCP SYN**扫描的原理
- **TCP SYN**扫描的过程
- **TCP SYN**扫描的特点

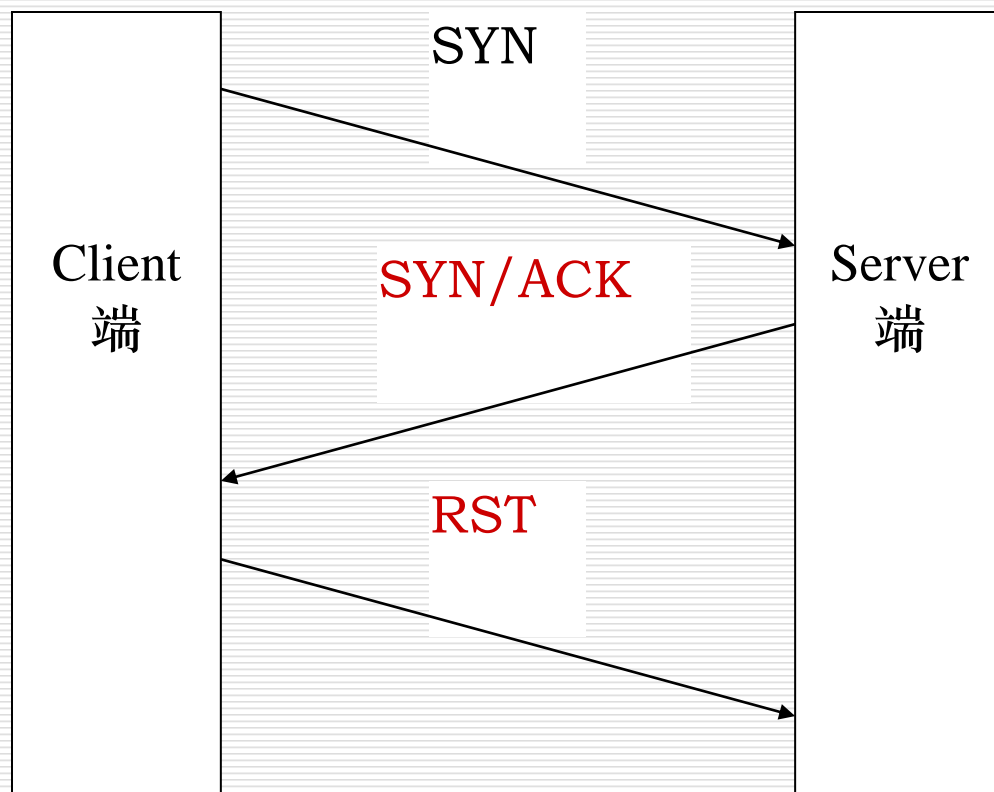
# TCP SYN扫描——原理

---

- 在这种技术中，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，按照设定继续探听其他端口；
  - 如果应答中包含**SYN**和**ACK**，说明目标端口处于监听状态。
- 由于**SYN**扫描时，全连接尚未建立，所以，这种技术通常被称为“**半连接**”扫描。

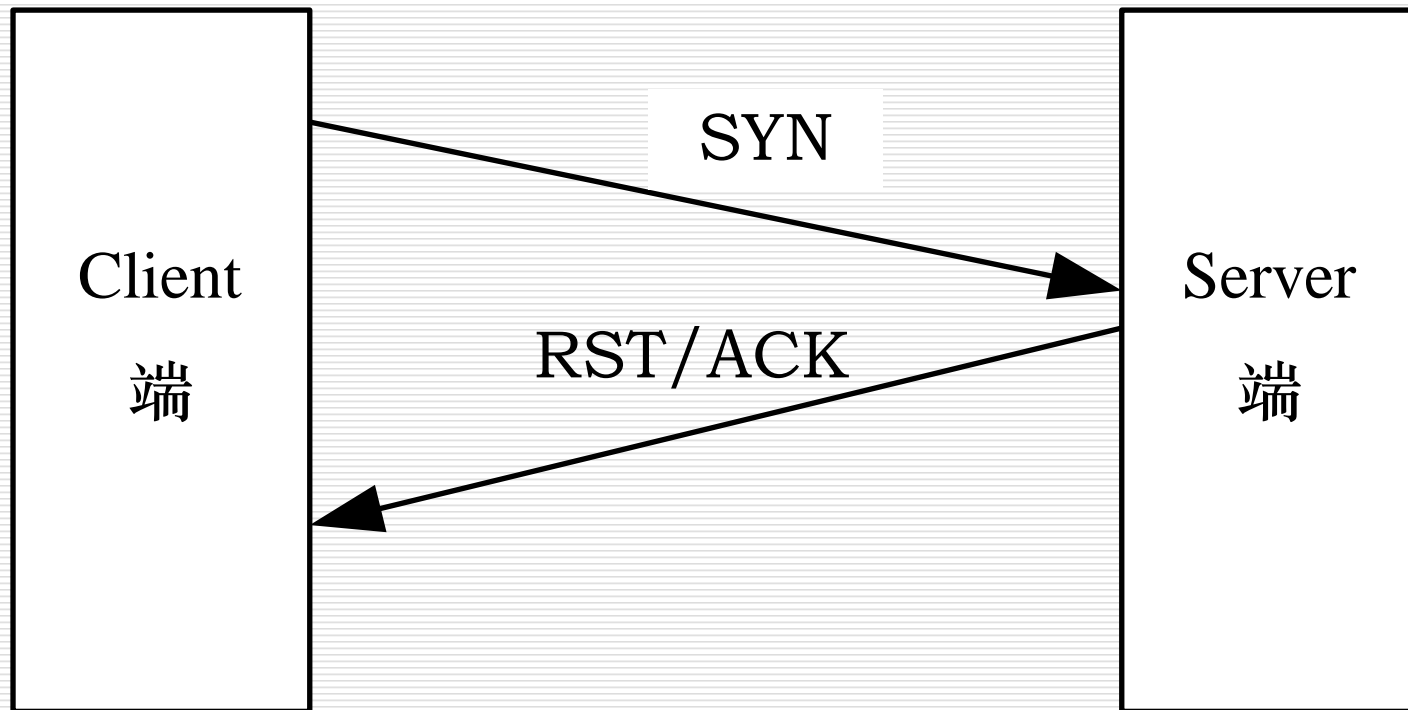
# TCP SYN扫描——过程（成功）

---



# TCP SYN扫描——过程（未成功）

---





# TCP SYN扫描——特点

---

- **SYN扫描的优点**在于即使日志中对于扫描有所记录，但是尝试进行连接的记录也要比全扫描的记录少的多。
- **SYN扫描缺点**是在大部分操作系统中，发送主机需要构造适用于这种扫描的**IP**包，通常情况下，构造**SYN**数据包**需要超级用户**或者得到授权的用户，才能访问专门的系统调用。

# 秘密扫描

---

- 秘密扫描总述
- **TCP FIN**扫描
  - 原理，过程，特点
  - 两个变种
    - Null扫描
    - Xmas扫描
- **SYN|ACK** 扫描
- **ACK**扫描

# 秘密扫描总述

---

- 秘密扫描是一种不被审计工具所检测的扫描技术
- 通常用于在通过普通的防火墙或路由器的筛选（**filtering**）时隐藏自己
- 优点：
  - 能躲避IDS、防火墙、包过滤器和日志审计，从而获取目标端口的开放或关闭的信息
- 缺点：
  - 扫描结果的不可靠性会增加，而且扫描主机也需要自己构造IP包

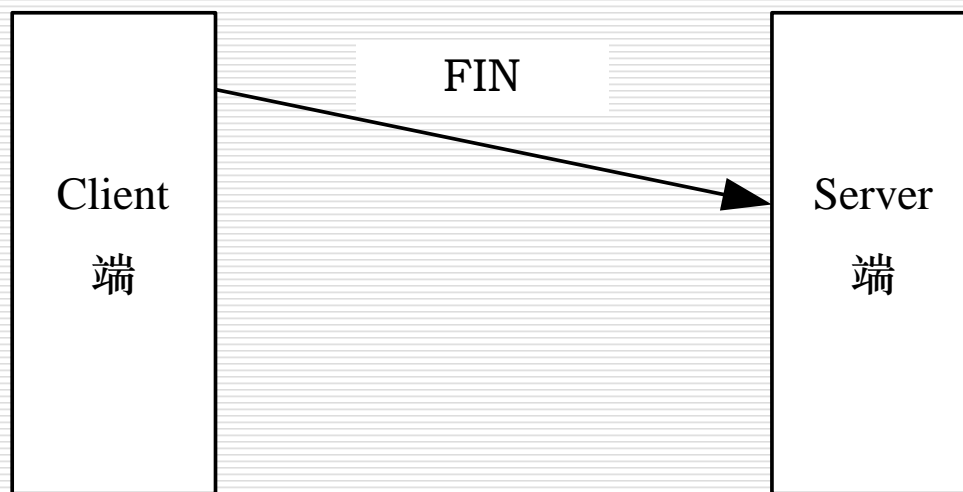
# TCP FIN扫描——原理

---

- **TCP FIN扫描技术使用FIN数据包探测端口：**
  - 当一个**FIN**数据包到达一个关闭的端口，数据包会被丢掉，且返回一个**RST**数据包。
  - 当一个**FIN**数据包到达一个打开的端口，数据包只是简单丢掉（不返回**RST**数据包）。
- 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**扫描隐蔽的多。
- **FIN**数据包能通过监测**SYN**包的包过滤器——**TCP FIN扫描**又称作**秘密扫描**。

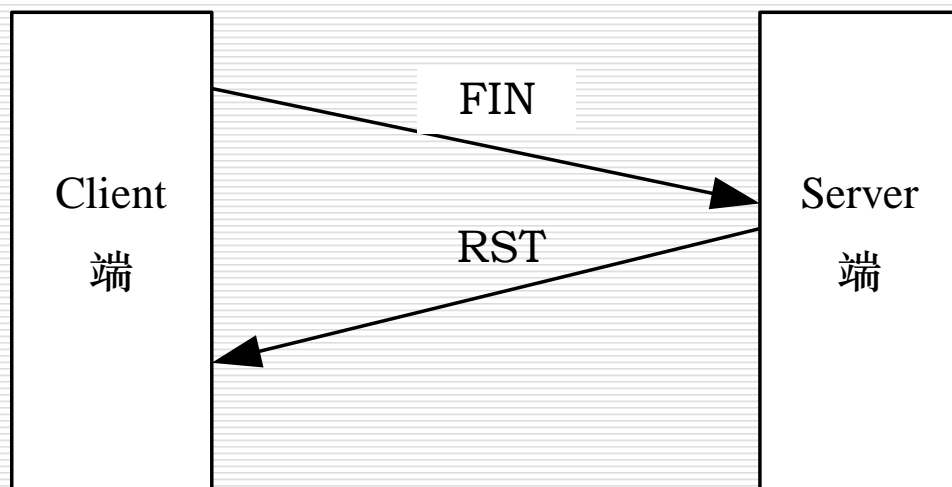
# TCP FIN扫描——过程（成功）

- 扫描主机向目标主机发送**FIN**数据包来探测端口，若**FIN**数据包到达的是一个打开的端口，数据包则被简单的丢掉，并不返回任何信息，如图所示：



# TCP FIN扫描——过程（未成功）

- 当**FIN**数据包到达一个关闭的端口，**TCP**会把它判断成是错误，数据包会被丢掉，并且回返回一个**RST**数据包，如图所示：



# TCP FIN扫描——特点

---

- ❑ **TCP FIN**扫描通常适用于**UNIX**目标主机。
- ❑ 在**Windows NT**环境下，该方法无效，因为不论目标端口是否打开，操作系统都发送**RST**。
- ❑ 但这一点在区分**UNIX**和**NT**两种不同操作系统时，却是十分有用的。

# TCP FIN扫描的两个变种

---

- **Xmas**和**Null**扫描是秘密扫描的两个变种
  - Xmas扫描打开FIN、URG和PSH标记
  - 而Null扫描关闭所有标记
- 使用这些组合的目的是为了通过所谓的**FIN**标记监测器的过滤。



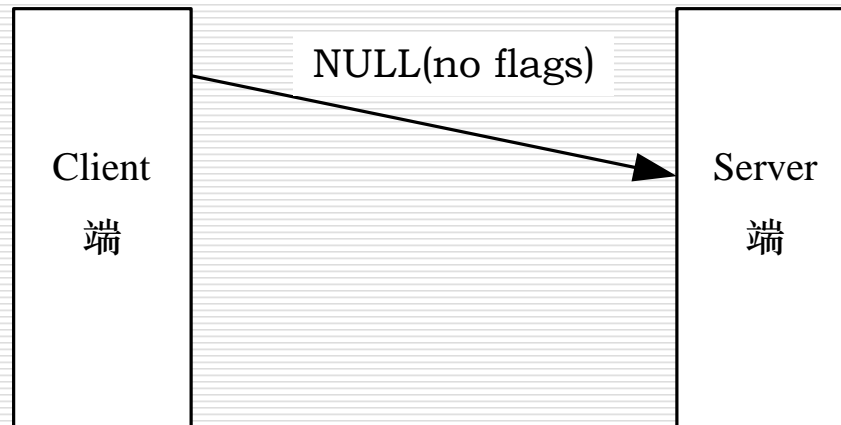
# TCP FIN扫描的变种——Null扫描

---

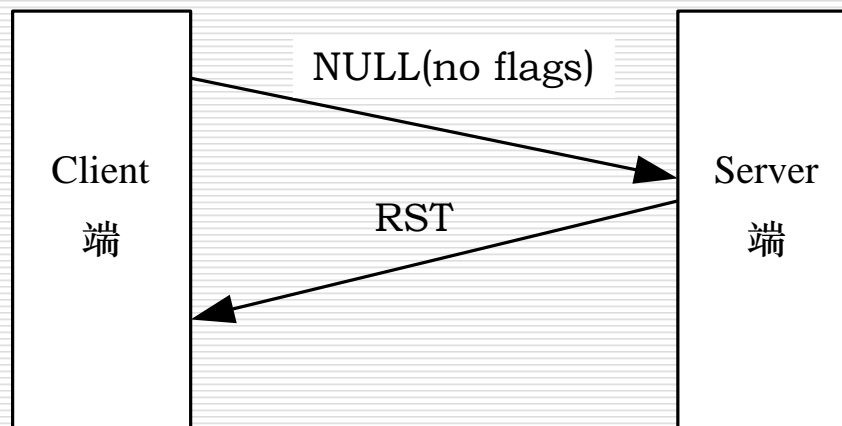
- 扫描主机将**TCP**数据包中的**ACK**（确认）、**FIN**（结束连接）、**RST**（重新设定连接）、**SYN**（连接同步化要求）、**URG**（紧急）、**PSH**(接收端将数据转由应用处理)标志位置空后发送给目标主机。

# TCP FIN扫描的变种——Null扫描

- 若目标端口开放，目标主机将不返回任何信息，如图所示：



- 若目标主机返回 **RST** 信息，则表示端口关闭，如图所示：



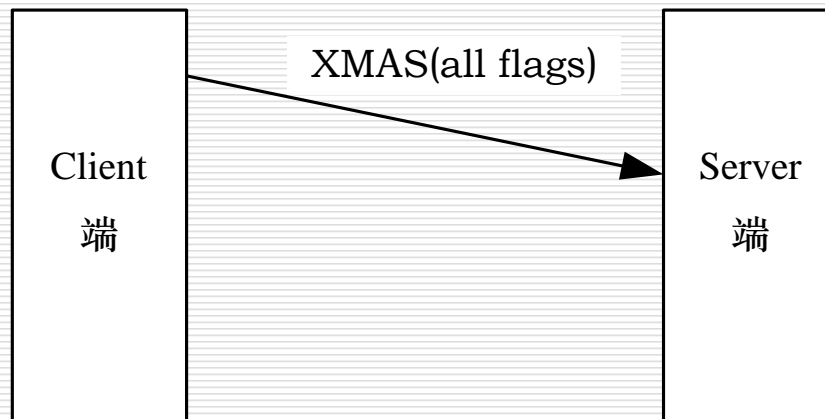
# TCP FIN扫描的变种—XMAS扫描

---

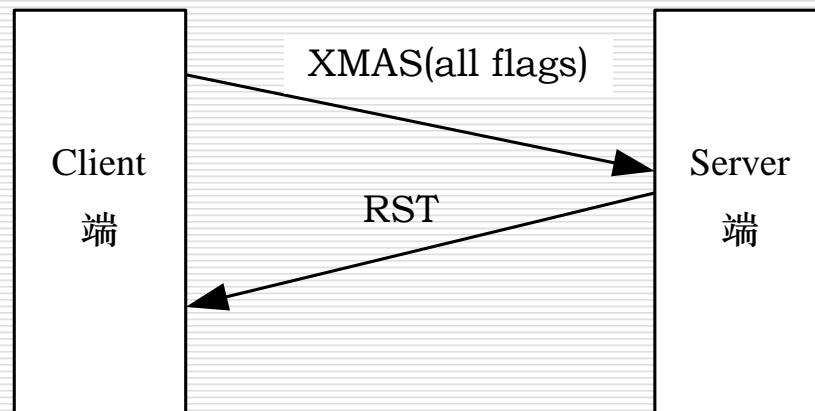
- **XMAS**扫描原理和**NULL**扫描的类似，将**TCP**数据包中的**ACK**、**FIN**、**RST**、**SYN**、**URG**、**PSH**标志位置**1**后发送给目标主机。

# TCP FIN扫描的变种—XMAS扫描

- 若目标端口开放，目标主机将不返回任何信息，如图所示：



- 若目标主机返回**RST**信息，则表示端口关闭，如图所示：



# SYN|ACK 扫描

---

- ❑ 扫描主机发送 **SYN+ACK**给目标主机
- ❑ 结果：
  - 如果端口关闭，则返回RST
  - 如果端口开放，不回复，因为TCP协议需要SYN标志才能启动连接
- ❑ 此扫描可能会生成一定数量的误报
  - 由过滤设备丢弃的数据包、网络流量、超时等可能会错误推断出端口开放

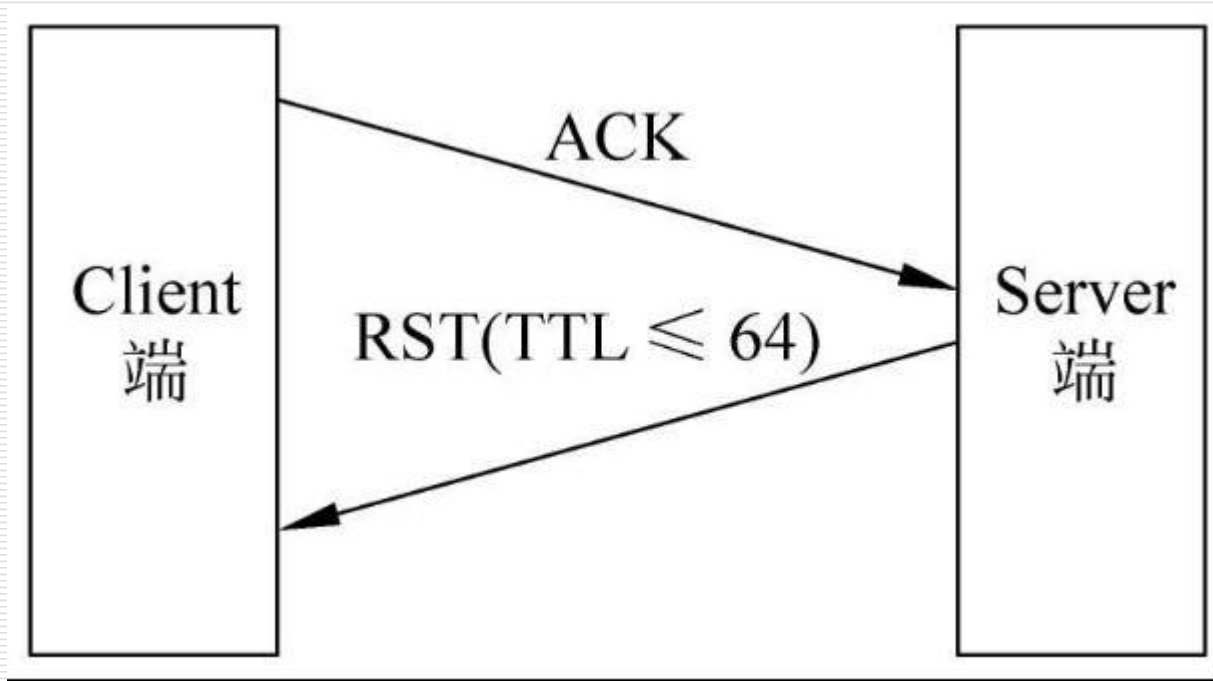
# ACK扫描

---

- ❑ 扫描主机向目标主机发送**ACK**数据包。
- ❑ 根据返回的**RST**数据包有两种方法可以得到端口的信息。

# ACK扫描

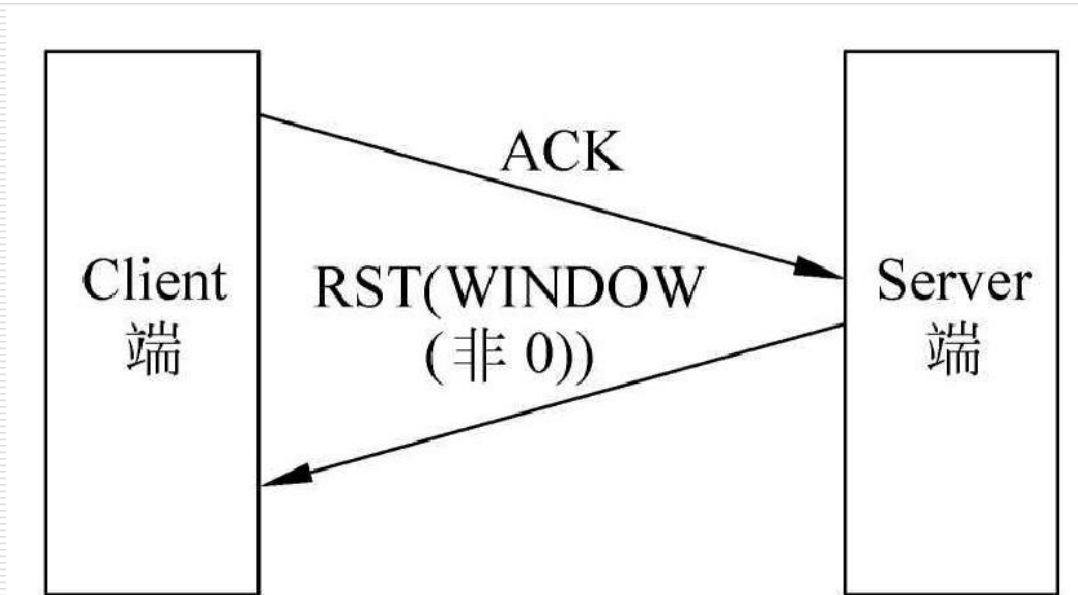
- 方法一： 若返回的**RST**数据包的**TTL**值小于或等于**64**，则端口开放，反之端口关闭



# ACK扫描

---

- 方法二：若返回的**RST**数据包的**WINDOW**值非零，则端口开放，反之端口关闭





# 认证(ident)扫描

- ❑ 认证（**ident**）协议一般用于网络连接过程中服务器验证客户端身份。监听**TCP 113**端口的**ident**服务安装在客户端，并由该**TCP**连接的服务端向客户端的**113**号端口发起认证连接。
- ❑ 连接过程：当客户端向服务器发送某个连接请求后，服务器便先向客户端的**TCP 113**端口发起连接，询问客户端该进程的拥有者名称，也就是问问“现在要连上我的这个家伙，在你那儿是什么身份”。服务器获取这一信息并认证成功后，记录下“某年某月某日谁连接到我的机器上”，再建立服务连接进行通信。

# 认证(ident)扫描

---

- ❑ 在端口扫描中，利用这一协议，扫描端先主动尝试与目标主机建立起一个**TCP**连接（第一个**TCP**连接）。
  - ❑ 第一个**TCP**连接成功之后，扫描端向目标主机的**TCP 113**端口建立另一连接（第二个**TCP**连接）。
  - ❑ 扫描端然后通过第二个**TCP**连接向目标主机的**ident**服务发送第一个**TCP**连接所对应的目标主机的端口号。如果目标主机安装并运行了**ident**服务，那么目标主机的**ident**服务将向扫描端返回相关联的进程的用户属性等信息。
  - ❑ 在此过程中，扫描端先以客户方身份与目标主机建立连接，后又以服务方身份对目标主机进行认证，因此这种扫描方式也被称为反向认证扫描。
-

# FTP代理扫描

---

- 文件传输协议（**FTP**）允许数据连接与控制连接位于不同的机器上，并支持代理**FTP**连接。
  - **FTP**代理扫描正是利用了这个缺陷，使用支持代理的**FTP**服务器来扫描**TCP**端口。这种扫描方式又被称为**FTP反弹扫描（FTP Bounce Attack）**。
  - 扫描端先在本地与一个支持代理的**FTP**服务器来建立控制连接，然后使用**PORT**命令向**FTP**服务器说明欲扫描的目标机器的**IP**地址和端口号，其中**IP**地址为代理传输的目的地址，而端口号则为传输时所需的被动端口，并发送**LIST**命令。
  - 这时，**FTP**服务器会尝试向目标主机指定端口发起数据连接请求。
-

# FTP代理扫描

---

- ❑ 如果目标主机对应端口确实处于监听状态，**FTP**服务器就会向扫描端返回成功信息，返回码为**150**和**226**。否则返回类似这样的错误信息：“**425 Can't build data connection: Connection refused**”。
  - ❑ 扫描端持续使用**PORT**和**LIST**命令，直到目标机器上所有的选择端口扫描完毕。
  - ❑ 这种方式隐藏性很好，难以跟踪，能轻而易举的绕过防火墙。不过对所有需扫描的端口都要逐一进行上述步骤，速度比较慢。而且，现在许多**FTP**服务器都禁止了代理这一特性。
-

# 远程主机OS指纹识别

---

- 基本原理
- 主动协议栈指纹识别
- 被动协议栈指纹识别

# 远程主机OS指纹识别的基本原理

---

- ❑ 操作系统（**Operating System**，简称**OS**）识别是入侵或安全检测需要收集的重要信息，是分析漏洞和各种安全隐患的基础。
- ❑ 只有确定了远程主机的操作系统类型、版本，才能对其安全状况作进一步的评估。
- ❑ 利用**TCP/IP**堆栈作为特殊的“指纹”，以确定系统的技术——远程主机**OS**指纹识别。
  - 主动协议栈指纹识别
  - 被动协议栈指纹识别

# 主动协议栈指纹识别

---

- 由于**TCP/IP**协议栈技术只是在**RFC**文档中描述，各个公司在编写应用于自己的**OS**的**TCP/IP**协议栈的时候，对**RFC**文档做出了不尽相同的诠释。
- 造成了各个**OS**在**TCP/IP**协议的实现上的不同。
- 通过对不同的**OS**的**TCP/IP**协议栈存在的些微差异的鉴别来判定**OS**类型。

# 主动协议栈指纹识别（2）

---

## □ 主要技术有：

- FIN探测
- ISN采样探测
- Don't Fragment位探测
- TCP初始窗口的大小检测
- ACK值探测
- ICMP出错消息抑制
- ICMP出错消息回射完整性
- TOS服务类型
- 片断处理



# FIN探测

- 通过向目标主机上的一个打开的端口发送一个**FIN**分组，然后等待回应；许多系统如：**WINNT、CISCO IOS、HP/UX、IRIX**的**TCP/IP**协议栈实现将返回一个**Reset**。

	FIN 探测 RESET	ISN 采样探测
WINNT	✓	
CISCO IOS	✓	
HP/UX	✓	
IRIX	✓	
Readhat	✗	
.....		

# ISN采样探测

---

- 这是寻找初始化序列长度模板与特定的**OS**匹配的方法，这样可以区分一些**OS**。
- 如早些的**UNIX**系统是**64 K**长度，而一些新的**UNIX**系统则是随机增加长度，如**Solaris**、**IRIX**、**FreeBSD**、**Digital Unix**、**Cray**等。

# Don't Fragment位探测

---

- 一些操作系统会设置IP头部“**Don't Fragment**位”（不分片位）以改善性能，监视这个位就可以判定区分远程**OS**。

# TCP初始窗口大小探测

---

- 简单检查返回的包里包含的窗口大小。某些**OS**在**TCP/IP**协议栈的实现中，这个值是独特的。如**AIX**是**0x3F25**，**NT**和**BSD**是**0x402E**，可以增加指纹鉴别的准确度。

# ACK值探测

---

- ❑ 不同的**OS**对**TCP/IP**协议栈实现在**ACK**包的序列号的值的选择上存在差异。
- ❑ 有些**OS**发回所确认的**TCP**包的序列号，另外一些则发回所确认的**TCP**包的序列号加**1**。

# ICMP出错信息抑制

---

- 有些**OS**限制**ICMP**出错消息的速率，通过某个随机选定的高端口发送**UDP**包，可能统计出在某个给定时间段内接受的不可达出错消息的数目。

# ICMP出错消息回射完整性

---

- ❑ 某些OS对TCP/IP协议栈的实现，在返回ICMP出错消息的时候会修改所引用的IP头，检测对IP头的改动的类型可以粗略判断OS。

# TOS服务类型

---

- ❑ 检测**ICMP**端口不可到达消息的**TOS**字段，多数**OS**会是**0**，而另一些则不是。



# 片断处理

---

- ❑ 不同的**TCP/IP**协议栈实现对重叠的片断处理上有差异。
- ❑ 有些在重组时会用到后到达的新数据覆盖旧数据，有些则相反。

# 被动协议栈指纹识别

---

- ❑ 主动协议栈指纹识别由于需要主动往目标发送数据包，但这些数据包在网络流量中比较惹人注意，因为正常使用网络不会按这样的顺序出现包，因此比较容易被**IDS**扑获。
- ❑ 为了**隐秘**的识别远程**OS**，需要使用被动协议栈指纹识别。

# 被动协议栈指纹识别（2）

---

- 被动协议栈指纹识别在原理上和主动协议栈指纹识别相似，但是它从不主动发送数据包，只是被动的捕获远程主机返回的包来分析其**OS**类型版本，一般可以从**4**个反面着手：
  - **TTL值**：这个数据是操作系统对出站的信息包设置的存活时间。
  - **Windows Size**：操作系统设置的TCP窗口大小，这个窗口大小是在发送**FIN**信息包时包含的选项。
  - **DF**：可以查看操作系统是否设置了不准分片位。
  - **TOS**：操作系统是否设置了服务类型。

# 漏洞扫描

---

- 漏洞扫描技术的原理
- 漏洞扫描技术的分类和实现方法
- 漏洞扫描的问题

# 漏洞扫描——原理

---

- 漏洞扫描主要通过以下两种方法来检查目标主机是否存在漏洞：
  - 基于漏洞库的特征匹配：通过端口扫描得知目标主机开启的端口以及端口上的网络服务后，将这些相关信息与网络漏洞扫描系统提供的漏洞库进行匹配，查看是否有满足匹配条件的漏洞存在；
  - 基于模拟攻击：通过模拟黑客的攻击手段，编写攻击模块，对目标主机系统进行攻击性的安全漏洞扫描，如测试弱势口令等，若模拟攻击成功，则表明目标主机系统存在安全漏洞。

# 漏洞扫描——分类和实现方法

---

- 基于网络系统漏洞库，漏洞扫描大体包括**CGI**漏洞扫描、**POP3**漏洞扫描、**FTP**漏洞扫描、**SSH**漏洞扫描、**HTTP**漏洞扫描等。这些漏洞扫描是基于漏洞库，将扫描结果与漏洞库相关数据匹配比较得到漏洞信息；
- 漏洞扫描还包括没有相应漏洞库的各种扫描，比如**Unicode**遍历目录漏洞探测、**FTP**弱势密码探测、**OPENRelay**邮件转发漏洞探测等，这些扫描通过使用插件（功能模块技术）进行模拟攻击，测试出目标主机的漏洞信息。

# 漏洞扫描——分类和实现方法

---

## □ 基于漏洞库的规则匹配

- 基于网络系统漏洞库的漏洞扫描的**关键部分就是它所使用的漏洞库**。通过采用基于规则的匹配技术，即根据安全专家对网络系统安全漏洞、黑客攻击案例的分析和系统管理员对网络系统安全配置的实际经验，可以形成一套标准的网络系统漏洞库，然后再在此基础上构成相应的匹配规则，由扫描程序自动的进行漏洞扫描的工作。
- 这样，**漏洞库信息的完整性和时效性决定了漏洞扫描系统的性能**，漏洞库的修订和更新的性能也会影响漏洞扫描系统运行的时间。因此，漏洞库的编制不仅要每个存在安全隐患的网络服务建立对应的漏洞库文件，而且应当能满足前面所提出的性能要求。

# 漏洞扫描——分类和实现方法

---

## □ 基于模拟攻击

- 将模拟攻击的模块做成插件的形式，插件是由脚本语言编写的子程序，扫描程序可以通过调用它来执行漏洞扫描，检测出系统中存在的一个或多个漏洞。添加新的插件就可以使漏洞扫描软件增加新的功能，扫描出更多的漏洞。插件编写规范化后，甚至用户自己都可以用perl、c或自行设计的脚本语言编写的插件来扩充漏洞扫描软件的功能。
- 这种技术使漏洞扫描软件的升级维护变得相对简单，而专用脚本语言的使用也简化了编写新插件的编程工作，使漏洞扫描软件具有强的扩展性。



# 漏洞扫描——问题

---

## □ 系统配置规则库问题

- 如果规则库设计的不准确，预报的准确度就无从谈起；
- 它是根据已知的安全漏洞进行安排和策划的，而对网络系统的很多危险的威胁却是来自未知的漏洞，这样，如果规则库更新不及时，预报准确度也会逐渐降低。

# 漏洞扫描——问题

---

## □ 漏洞库信息要求

- 漏洞库信息是**基于网络系统漏洞库的漏洞扫描**的主要判断依据。如果漏洞库信息不全面或得不到即时的更新，不但不能发挥漏洞扫描的作用，还会给系统管理员以错误的引导，从而对系统的安全隐患不能采取有效措施并及时的消除。

## 2.3 扫描工具赏析

---

- 扫描工具概述
- 如何获取扫描工具
- 常用扫描工具
- 常用扫描工具比较
- 其它扫描工具



# 扫描工具概述

---

- 如果扫描范围具有一定的规模，比如要在一个较大的范围内对网络系统进行安全评估，那就需要使用一些多功能的综合性工具。
- 一般来说，这些多功能的综合性扫描工具，都可以对大段的网络**IP**进行扫描，其扫描内容非常广泛，基本上包含了各种专项扫描工具的各个方面。

# 如何获取扫描工具

---

- ❑ 各种工具的官方主页
  - <https://zh-cn.tenable.com/products/nessus>
  - <https://www.openvas.org/>
  - <https://nmap.org/>
- ❑ 有些工具是系统自带的，比如**windows**和**linux**中的**ping**，**linux**中的**nmap**
- ❑ **Kali Linux** 集成了很多渗透测试的工具
- ❑ **Github**上开源工具
  - 2019年Github上开源的安全渗透攻击类工具集合  
<https://zhuanlan.zhihu.com/p/53112370>

# 常用扫描工具

---

- ❑ **SATAN** 古老的经典
- ❑ **Nmap** 扫描技术集大成者
- ❑ **Nessus** 黑客的血滴子，网管的百宝箱
- ❑ **OpenVAS** 原理与 **Nessus** 类似且免费
- ❑ **X-scan** 国内黑客的最爱

# SATAN

---

## □ SATAN概述

## □ SATAN的分级

- 轻度扫描
- 标准扫描
- 重度扫描
- 攻入系统

## □ SATAN的特点

# SATAN概述

---

- **SATAN**是**Security Administrator Tool for Analyzing Networks**（用于分析网络的安全管理员工具）的缩写，作者是**Dan Farmer**和**Wietse Venema**。
- **1995年4月5日**，**SATAN**的发布引起了轩然大波。但是引起争论的更重要的原因，主要是**SATAN**带来了关于网络安全的全新观念。



# SATAN概述(2)

---

- 在**SATAN**发表之前，关于网络安全的防护上，模糊安全论已经有很长一段时间的讨论。
- **模糊安全**的概念——是大多数软件、操作系统厂商喜欢处理安全漏洞的一种方法，他们认为**安全漏洞应该隐藏**，不要在文档中公布，因为很少人会发现这些漏洞，即使有人发现这些漏洞，也不会去研究和利用漏洞。

# SATAN概述(3)

---

- 实际上随着软件规模的日益增大，尽管程序员一般不会故意在程序中留下漏洞，软件中出现安全漏洞也是不可避免的。
- 及时地公布安全漏洞和补丁，让网络管理员及时进行补救，才是正确的方法。
- 而**SATAN**的公布，的确促使所有的操作系统厂商意识到应该及时修正他们的系统中的漏洞。

# SATAN概述(4)

---

- **SATAN**的出现，带来了网络安全方面的全新的观念：以黑客的方式来思考网络安全的问题。
- 这个观念体现在**Farmer**和**Venema** 于**1993**年发表的文章《**通过攻入你的网站来提高安全**》（**Improving the Security of Your Site by Breaking Into It**）中。

# SATAN的分级

---

- ☐ 轻度扫描
- ☐ 标准扫描
- ☐ 重度扫描
- ☐ 攻入系统

# 轻度扫描

---

- 轻度扫描包含最少的入侵扫描。**SATAN**从域名系统（**Domain Name System**，简称**DNS**）收集信息，看看主机提供哪些远程过程调用，通过网络提供哪些文件共享。根据这些信息，**SATAN**就得到主机的一般信息。

# 标准扫描

---

- 在这个层次上，**SATAN**探测常见的网络服务，包括**finger**、**remote login**、**ftp**、**www**、**gopher**、**email**等。根据这些信息，**SATAN**能判断主机的类型，是**Windows**服务器，还是**HP-UNIX**，**Solaris**还是**Linux**，甚至还能探测服务器软件的版本。

# 重度扫描

---

- 当知道目标主机提供什么服务之后，**SATAN**进行更深入的扫描。在这个扫描级别上，**SATAN**探测匿名服务器的目录是不是可写，**X Windows**服务器是不是关闭了访问控制，**/etc/hosts.equiv**文件中是否有“\*”号，等等。

# 攻入系统

---

- 本级别还包括了对系统进行的攻击、清扫攻击时留下的痕迹、隐藏自己等，不过**SATAN**只提出了这一级别的思想，但并没有实现。



# SATAN的特点

---

- **SATAN**作为最早的并且是最典型的扫描工具，具备以下特点：
  - 扫描指定的主机系统
  - 扫描常见的弱点
  - 给数据分析提供帮助
- 总之，**SATAN**能够自动扫描本地和远程系统的弱点，为系统的安全或远程攻击提供帮助。

# NMAP

---

- **NMAP**简介
- **NMAP**基本功能
- **NMAP**功能架构图
- **Zenmap**简介
- **NMAP**使用说明
- **NMAP**使用示例
- **NMAP**特点

# NMAP简介

---

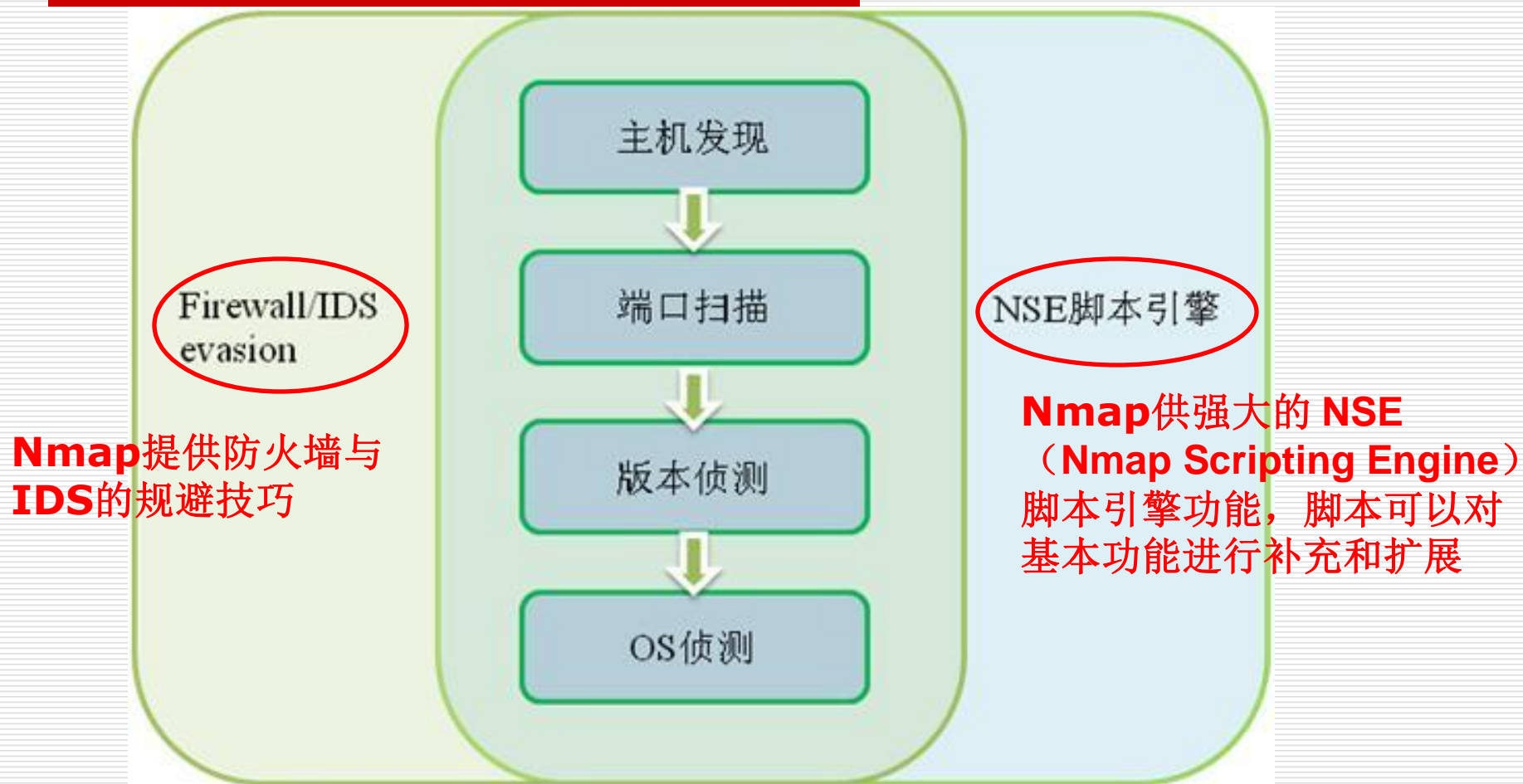
- **Nmap(Network Mapper)**，是由**Fyodor**制作的开源免费的网络发现(**Network Discovery**)和安全审计(**Security Auditing**)工具。
  - 它除了提供基本的**TCP**和**UDP**端口扫描功能外，还集成了众多扫描技术。现在的端口扫描技术很大程度上是根据**Nmap**的功能设置来划分的。
  - **Nmap**还有一个卓越的功能，那就是采用一种叫做“**TCP**栈指纹鉴别(**stack fingerprinting**)”的技术来探测目标主机的操作系统类型。
-

# Nmap的基本功能

---

- ❑ 主机发现 (**Host Discovery**)
- ❑ 端口扫描 (**Port Scanning**)
- ❑ 版本侦测 (**Version Detection**)
- ❑ 操作系统侦测 (**Operating System Detection**)

# Nmap功能架构图



# Zenmap简介

---

- **Nmap**官方提供的开源免费的图形界面，能够运行在不同操作系统平台上（**Windows /Linux /Unix /Mac OS**等）
- **Zenmap**为**nmap**提供更加简单的操作方式
  - 简单常用的操作命令可以保存成为profile，用户扫描时选择profile即可
  - 可以方便地比较不同的扫描结果
  - 提供网络拓扑结构(Network Topology)的图形显示功能

Zenmap

Scan Tools Profile Help

Target: 192.168.0.1/24 扫描目标 Profile: Intense scan 用于选择“Zenmap默认提供的”

Command: nmap -T4 -A -v 192.168.0.1/24 显示选择Profile对应的命令或者“目标器创建的Profile”

Hosts Services

OS Host

localhost (192.168.0.1)  
localhost (192.168.0.88)  
localhost (192.168.0.104)  
localhost (192.168.0.105)  
localhost (192.168.0.106)  
localhost (192.168.0.110)  
localhost (192.168.0.111)

扫描目标中发  
现的主机列表

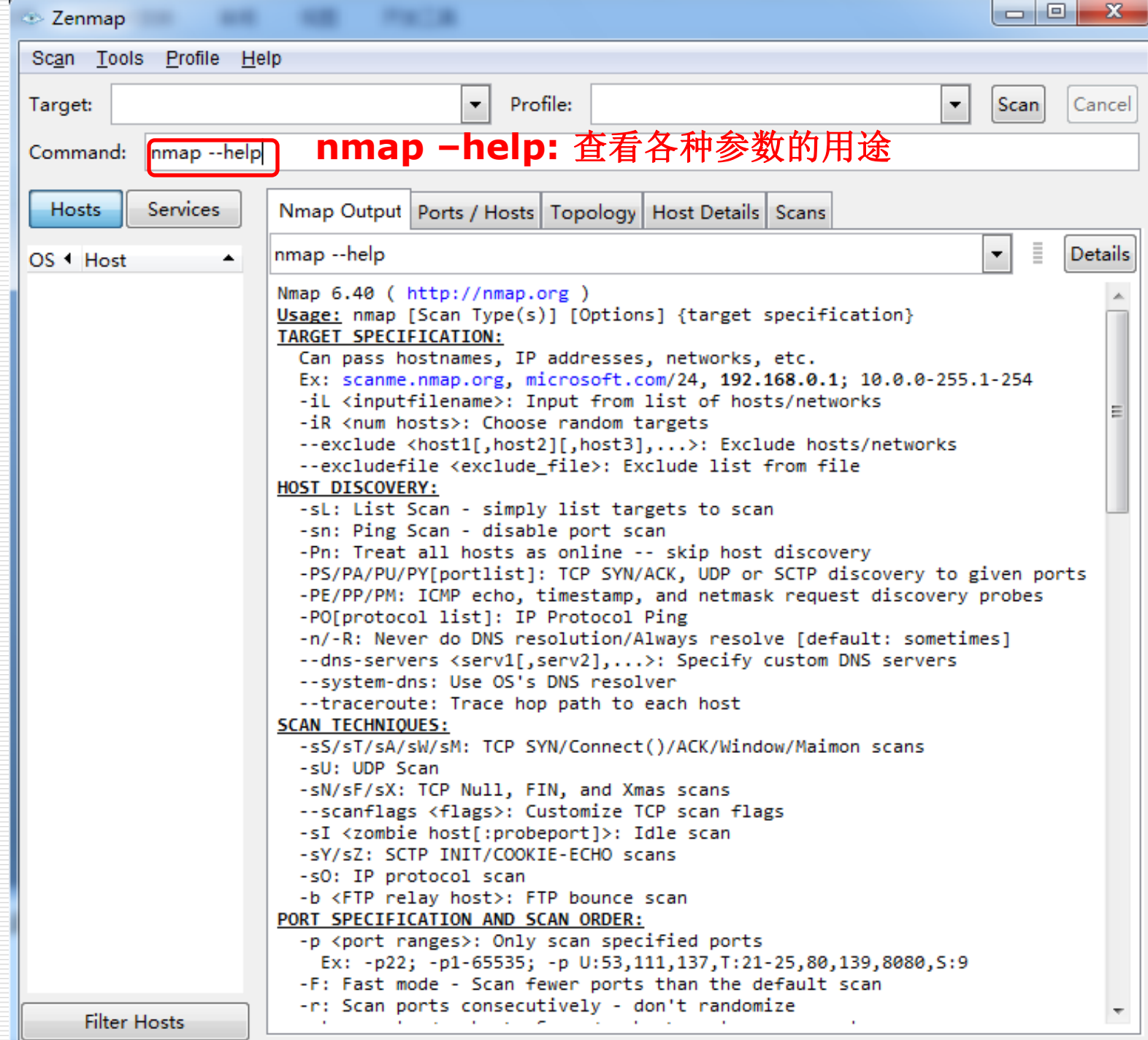
Nmap Output Ports / Hosts Topology Host Details Scans

nmap -T4 -A -v 192.168.0.1/24

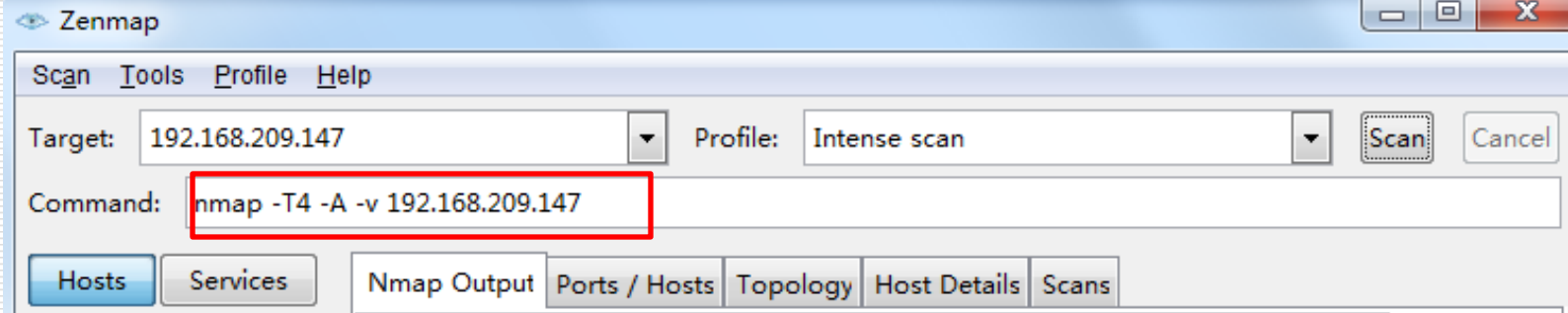
Completed Parallel DNS resolution of 1 host. at 23:54, 0.01s elapsed  
Initiating SYN Stealth Scan at 23:54  
Scanning 6 hosts [1000 ports/host]  
Discovered open port 80/tcp on 192.168.0.111  
Discovered open port 80/tcp on 192.168.0.1  
Discovered open port 3389/tcp on 192.168.0.105  
Discovered open port 22/tcp on 192.168.0.111  
Discovered open port 21/tcp on 192.168.0.111  
Discovered open port 8080/tcp on 192.168.0.104  
Discovered open port 49152/tcp on 192.168.0.104  
Discovered open port 32768/tcp on 192.168.0.111  
Discovered open port 912/tcp on 192.168.0.105  
Completed SYN Stealth Scan against 192.168.0.1 in 5.71s (5 hosts left)  
Completed SYN Stealth Scan against 192.168.0.111 in 6.53s (4 hosts left)  
Completed SYN Stealth Scan against 192.168.0.106 in 8.84s (3 hosts left)  
Completed SYN Stealth Scan against 192.168.0.88 in 16.39s (2 hosts left)  
Completed SYN Stealth Scan against 192.168.0.104 in 20.31s (1 host left)  
Discovered open port 6001/tcp on 192.168.0.105  
Completed SYN Stealth Scan at 23:54, 23.08s elapsed (6000 total ports)  
Initiating Service scan at 23:54  
Scanning 10 services on 6 hosts  
Completed Service scan at 23:56, 110.10s elapsed (10 services on 6 hosts)  
Initiating OS detection (try #1) against 6 hosts  
Retrying OS detection (try #2) against 4 hosts  
WARNING: OS didn't match until try #2  
NSE: Script scanning 6 hosts.  
Initiating NSE at 23:56  
Completed NSE at 23:57, 30.55s elapsed  
Nmap scan report for localhost (192.168.0.1)  
Host is up (0.0022s latency).  
Not shown: 999 closed ports  
PORT STATE SERVICE VERSION  
80/tcp open http GoAhead-Webs embedded httpd  
|\_http-methods: No Allow or Public header in OPTIONS response (status  
code 400)

扫描发现的每台主机  
的1000个常见端口  
的开放情况

Filter Hosts







## TIMING AND PERFORMANCE:

Options which take <time> are in seconds, or append 'ms' (milliseconds), 's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).

**-T<0-5>: Set timing template (higher is faster)**

**-T**: 级别越高，扫描速度越快，但也容易被防火墙或IDS检测并屏蔽掉，在网络通讯状况良好的情况推荐使用T4

**-6**: Enable IPv6 scanning

**-A**: Enable OS detection, version detection, script scanning, and traceroute

**-A**: 用于使用进攻性（**Aggressive**）方式扫描

**--privileged**: Assume that the user is fully privileged

**--unprivileged**: Assume the user lacks raw socket privileges

## OUTPUT:

**-oN/-oX/-oS/-oG <file>**: Output scan in normal, XML, s|<rIpt kIddi3, and Grepable format, respectively, to the given filename.

**-oA <basename>**: Output in the three major formats at once

**-v**: Increase verbosity level (use -vv or more for greater effect)

**-v**: 显示冗余（**verbosity**）信息，在扫描过程中显示扫描的细节，从而让用户了解当前的扫描状态

# NMAP使用说明

---

- **Ping扫描：**了解哪些机器是**up**的
  - `nmap -sP 202.38.64.1`
  - 缺省时同时使用发送icmp和对80端口发送ack来探测
  - 可以用`nmap -sP -P0` 不发送icmp消息
- **TCP connect扫描：**
  - `nmap -sT 202.38.64.1`
- **TCP SYN扫描**
  - `nmap -sS 202.38.64.1`
- **TCP FIN, XMAS, NULL扫描：**
  - `nmap -sF 202.38.64.1`
  - `nmap -sX 202.38.64.1`
  - `nmap -sN 202.38.64.1`
- **UDP扫描：**
  - `nmap -sU 202.38.64.1`

# NMAP使用示例(ping扫描)

Command: `nmap -sn 124.16.84.0/22`

Hosts

Services

OS Host

124.16.84.25

124.16.84.32

124.16.84.137

124.16.84.138

124.16.84.147

124.16.87.249

124.16.87.250

124.16.87.254

Nmap Output

Ports / Hosts

Topology

Host Details

Scans

nmap -sn 124.16.84.0/22

Details

Starting Nmap 6.40 ( <http://nmap.org> ) at 2013-09-08 14:06 中国标准时间

Nmap scan report for 124.16.84.25

Host is up (0.013s latency).

MAC Address: 00:24:8C:A5:F2:A6 (Asustek Computer)

Nmap scan report for 124.16.84.32

Host is up (8.9s latency).

MAC Address: F0:BF:97:63:8B:59 (Sony)

Nmap scan report for 124.16.84.137

Host is up (0.0040s latency).

MAC Address: 20:89:84:E9:DB:EE (Compal Information (kunshan) CO.)

Nmap scan report for 124.16.84.138

Host is up (0.0088s latency).

MAC Address: B8:70:F4:2E:36:77 (Compal Information (kunshan) CO.)

Nmap scan report for 124.16.87.249

Host is up (0.0031s latency).

MAC Address: A0:64:11:48:01:02 (Unknown)

Nmap scan report for 124.16.87.250

Host is up (0.0040s latency).

MAC Address: A0:64:11:48:05:72 (Unknown)

Nmap scan report for 124.16.87.254

Host is up (0.0056s latency).

MAC Address: 74:25:8A:0E:B2:AF (Hangzhou H3C Technologies Co., Limited)

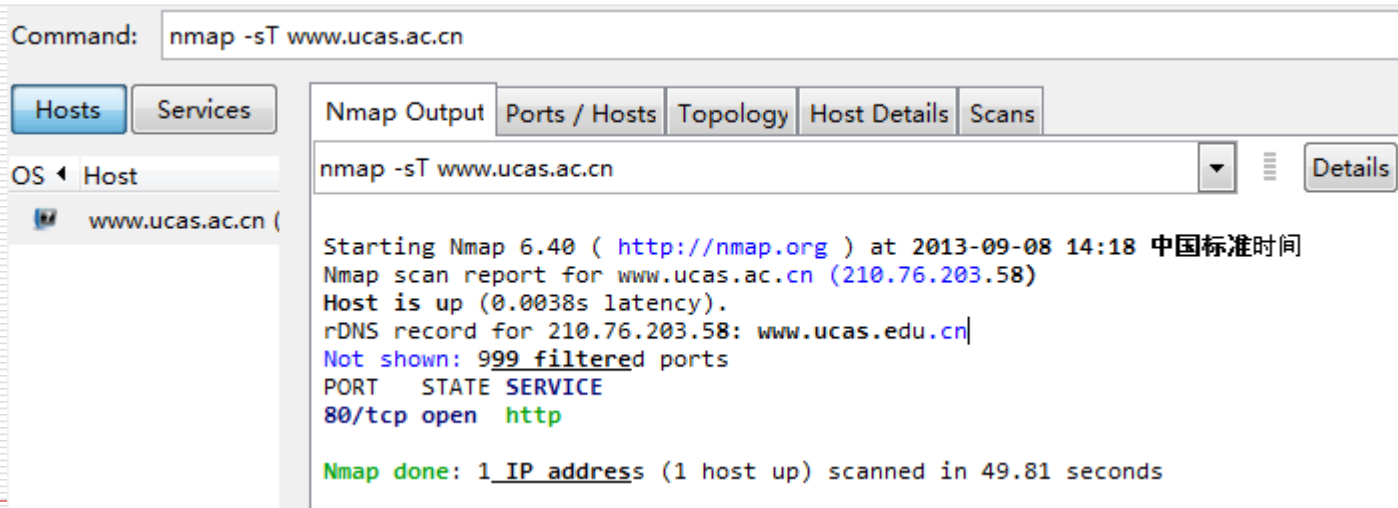
Nmap scan report for 124.16.84.147

Host is up.

Nmap done: 1024 IP addresses (8 hosts up) scanned in 97.38 seconds

# NMAP使用示例(TCP connect扫描)

- 使用**-sT**选项指定进行**TCP connect**端口扫描（全扫描），如果不指定端口号，缺省情况下**Nmap**会扫描**1-1024**和**nmap-services**文件(在**Nmap**下载包中)中列出的服务端口号。
- 下图是利用**-sT**对**www.ucas.ac.cn** 主机进行**TCP connect**扫描的情况，最终结果显示**80**端口是开放的。



```
Command: nmap -sT www.ucas.ac.cn

Hosts Services
OS Host
www.ucas.ac.cn (

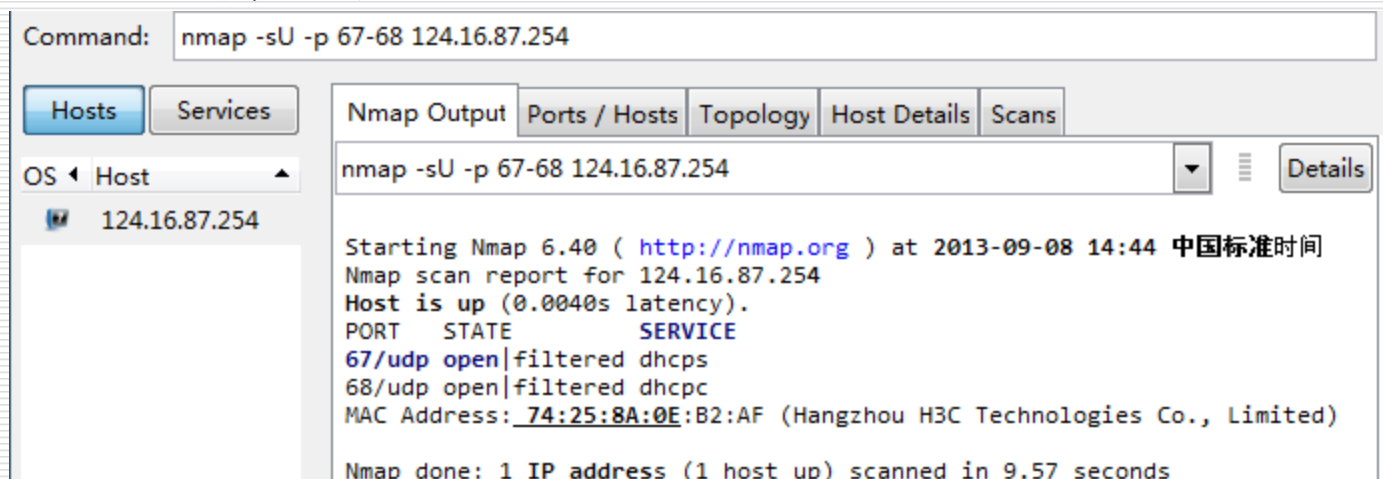
Nmap Output Ports / Hosts Topology Host Details Scans
nmap -sT www.ucas.ac.cn Details

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-08 14:18 中国标准时间
Nmap scan report for www.ucas.ac.cn (210.76.203.58)
Host is up (0.0038s latency).
rDNS record for 210.76.203.58: www.ucas.edu.cn|
Not shown: 999 filtered ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 49.81 seconds
```

## NMAP使用示例(UDP端口扫描)

- ❑ Nmap也可以用于进行UDP端口扫描，只需要指定-sU选项，还可以指定扫描的目标端口
- ❑ 下图是对网关124.16.87.254进行指定的UDP端口扫描



# NMAP使用示例(操作系统类型探测)

The screenshot displays the Nmap GUI interface. At the top, the command bar shows 'nmap -sT -O -v -Pn 210.77.16.29'. Below this, there are tabs for 'Hosts', 'Services', 'Nmap Output', 'Ports / Hosts', 'Topology', 'Host Details', and 'Scans'. The 'Hosts' tab is active, showing a list of hosts with 'auth.gucas.ac.cn' selected. The 'Nmap Output' tab is also visible, showing the scan results. The output text is as follows:

```
nmap -sT -O -v -Pn 210.77.16.29

Completed Connect Scan at 15:00, 225.34s elapsed (1000 total ports)
Initiating OS detection (try #1) against auth.gucas.ac.cn (210.77.16.29)
Nmap scan report for auth.gucas.ac.cn (210.77.16.29)
Host is up (0.14s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
3333/tcp  open  dec-notes
8080/tcp  open  http-proxy
8800/tcp  open  sunwebadmin
8899/tcp  open  ospf-lite
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux kernel:2.6
OS details: Linux 2.6.24 - 2.6.36
Uptime guess: 19.811 days (since Mon Aug 19 19:33:13 2013)
Network Distance: 3 hops
TCP Sequence Prediction: Difficulty=201 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: D:\Program Files\Nmap
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 234.26 seconds
Raw packets sent: 19 (1.598KB) | Rcvd: 23 (1.778KB)
```

The 'OS details' line, 'Linux 2.6.24 - 2.6.36', is highlighted with a red box in the original image.

# NMAP特点

---

- ❑ **NMAP** 是一款开源的扫描工具, 用于系统管理员查看一个大型的网络有哪些主机以及其上运行何种服务。
- ❑ 它支持多种协议多种形式的扫描技术, 还提供一些实用功能如通过**TCP/IP**来鉴别操作系统类型、秘密扫描、动态延迟和重发、平行扫描、通过并行的**PING**鉴别下属的主机、欺骗扫描、端口过滤探测、直接的**RPC**扫描、分布扫描、灵活的目标选择以及端口的描述。
- ❑ **NMAP**主要的特色就是多种扫描模式以及指纹识别技术。

# Nessus

---

- **Nessus**简介
- **Nessus**的特点
- **Nessus**的功能与所用的技术
- **Nessus**的结构
- **Nessus**的扫描过程
- **Nessus**的安装



# Nessus简介

---

- ❑ **Nessus**是一个功能强大而又简单易用的网络安全扫描工具，对网络管理员来说，它是不可多得的审核堵漏工具。
- ❑ **2000年、2003年、2006年**，**Nmap**官方在**Nmap**用户中间分别发起“**Top 50 Security Tools**”、“**Top 75 Security Tools**”、“**Top 100 Security Tools**”的评选活动，**Nessus**“战胜”众多的商业化漏洞扫描工具而三次夺魁。
- ❑ **Nessus**在第三版发布时收回了**Nessus**的版权和程序源代码
- ❑ **Nessus**被誉为**黑客的血滴子**，**网管的百宝箱**。

# Nessus简介(2)

---

- ❑ **Nessus**采用基于插件的技术。
- ❑ 工作原理是通过插件模拟黑客的攻击，对目标主机系统进行攻击性的安全漏洞扫描，如测试弱势口令等，若模拟攻击成功，则表明目标主机系统存在安全漏洞。
- ❑ **Nessus**可以完成多项安全工作，如扫描选定范围内的主机的端口开放情况、提供的服务、是否存在安全漏洞等等。

# Nessus的特点

---

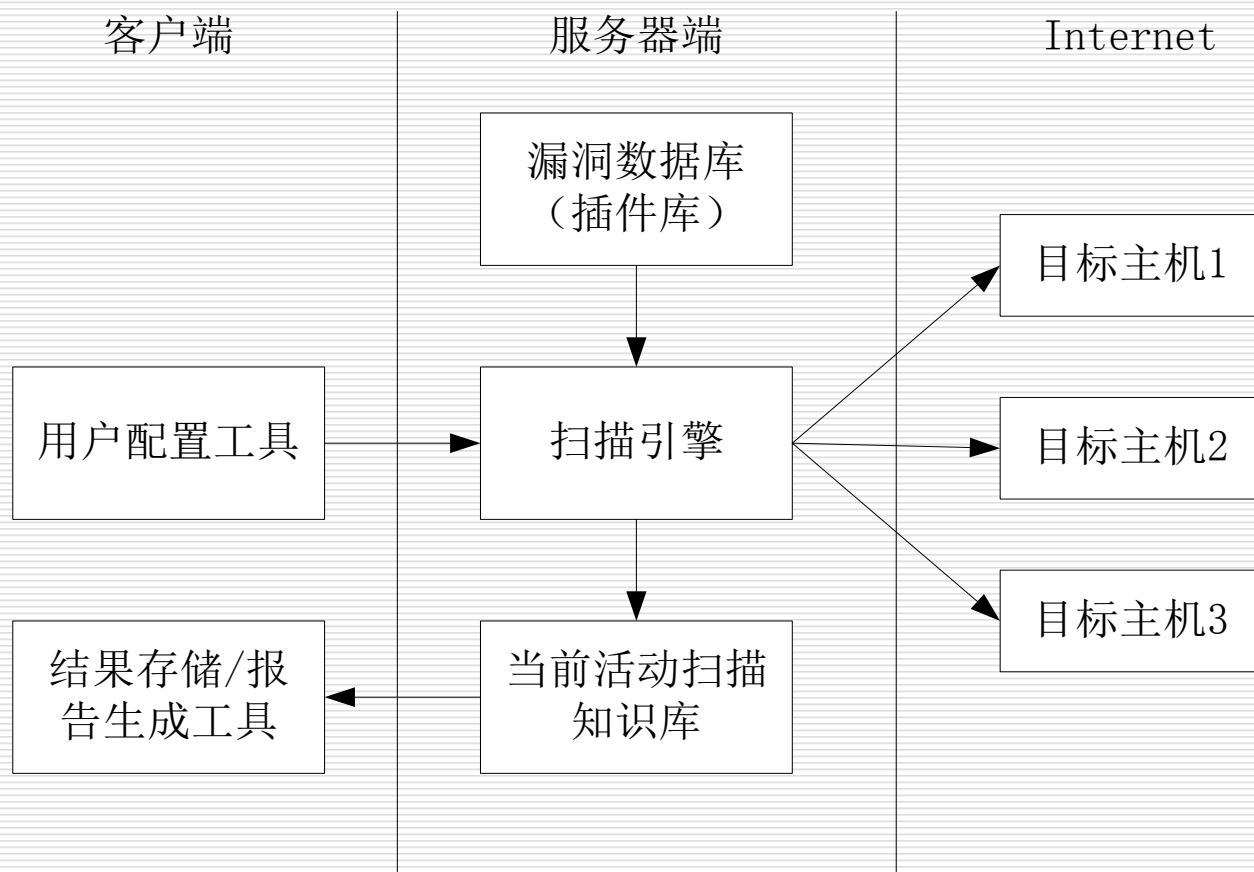
- ❑ 采用了基于多种安全漏洞的扫描，避免了扫描不完整的情况。
- ❑ **Nessus**基于插件体制，扩展性强，支持及时的在线升级，可以扫描自定义漏洞或者最新安全漏洞。
- ❑ **Nessus**采用客户端/服务端机制，容易使用、功能强大。

# Nessus的功能与所用的技术

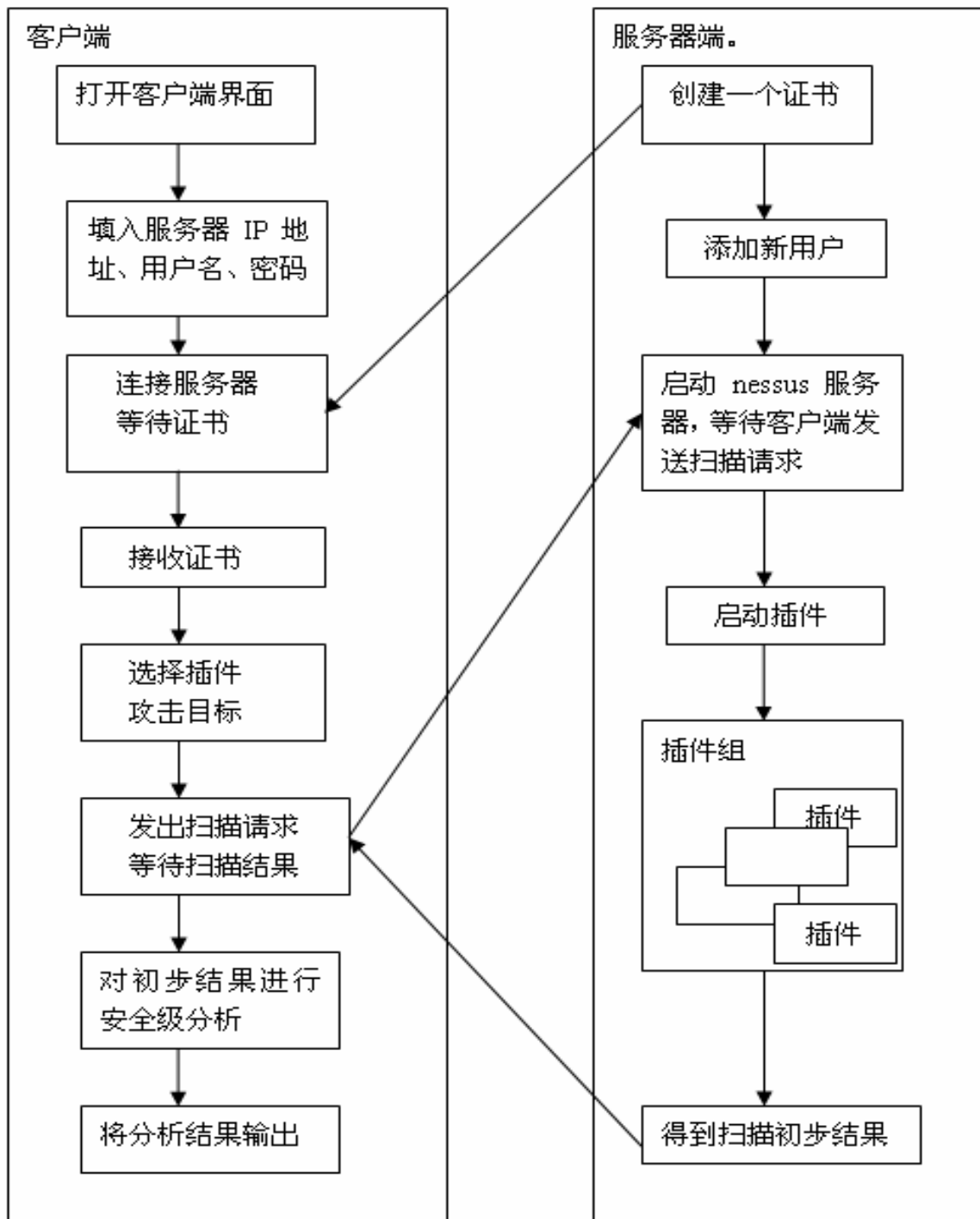
---

- 主机扫描技术
- 端口扫描技术
- 远程主机**OS**识别
- 漏洞扫描技术
  - 这是比Nmap多的功能
  - Nessus自带的上万个扫描插件是其最引人注目的功能

# Nessus的结构



# Nessus的扫描过程



# Nessus的安装环境

---

## □ Nessus 在类Unix环境下安装

- RedHat环境下：下载[Nessus-5.2.1-es6.i386.rpm](#)软件包，然后使用 `rpm -ivh Nessus-5.2.1-es6.i386.rpm` 命令安装

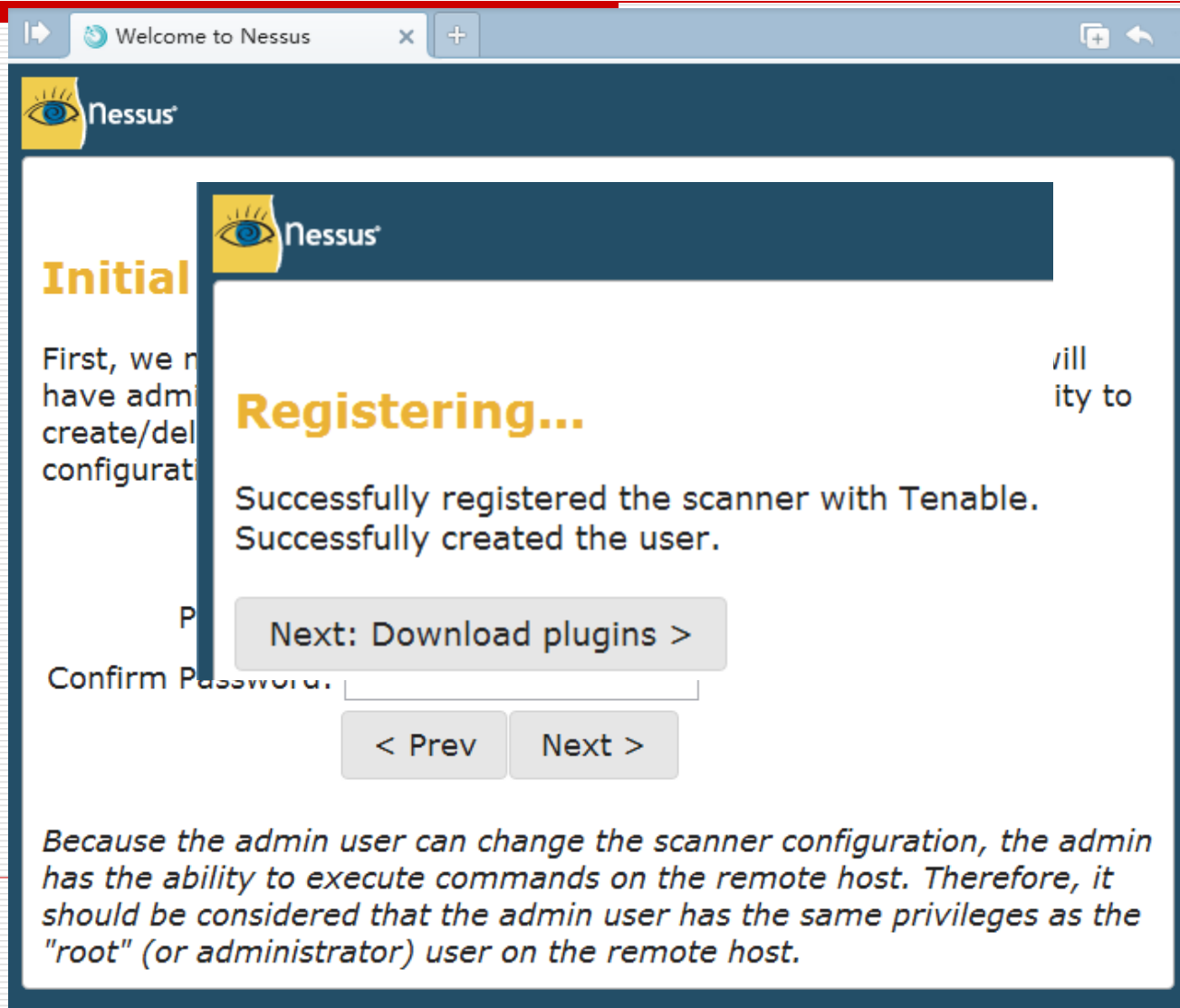
## □ Nessus 在windows下的安装

- 下载[Nessus-5.2.1-Win32.msi](#)，然后直接安装这个安装包即可
- 下面是在win7下面的安装过程

## □ Nessus 是在线安装的

# Nessus 5.2.1安装

注册用户





# Nessus的安装



**Nessus is fetching the newest plugin set**



正在下载安装插件集到本地

**Nessus is initializing**

Please wait...



process. Once the plugins are downloaded and processed, subsequent startups will be much faster.

Since this operation is taking some time, here are some useful links:

- Documentation: This page contains all of the manuals that you'll need to get the most out of Nessus and its features.

[返回](#)

# OpenVAS

---

- ❑ **OpenVAS**简介
- ❑ **OpenVAS**的特点
- ❑ **OpenVAS**的功能与所用的技术
- ❑ **OpenVAS**的结构

# OpenVAS简介

---

- ❑ **Tenable**的开源或商业混合的工具**Nessus**已经有超过**10**年的历史了，但是目前该工具已经不再是免费的了，并且从**3.0**版开始，不再开源。
- ❑ 为了适应**Nessus**的商业化和开源代码的不开源化，开发了开放式漏洞评估系统**OpenVAS (Open Vulnerability Assessment System)**。最开始，其只是**Nessus**的一个纯**GPL**性质的克隆产物。但现在已经开始了进一步的开发，并扩展了**Nessus**项目所没有的能力和函数。

# OpenVAS的特点

---

- ❑ **OpenVAS**是一套开源的漏洞/弱点扫描系统，可以强力替代著名的**Nessus**漏洞检测系统（已经非开源）。
- ❑ **OpenVAS**集成了多个服务、组件，既提供大量的免费扫描插件（**NVT**, **Network Vulnerability Tests**），也提供商业化的增强扫描插件（**GSF**, **Greenbone Security Feed**）。

# OpenVAS的特点

---

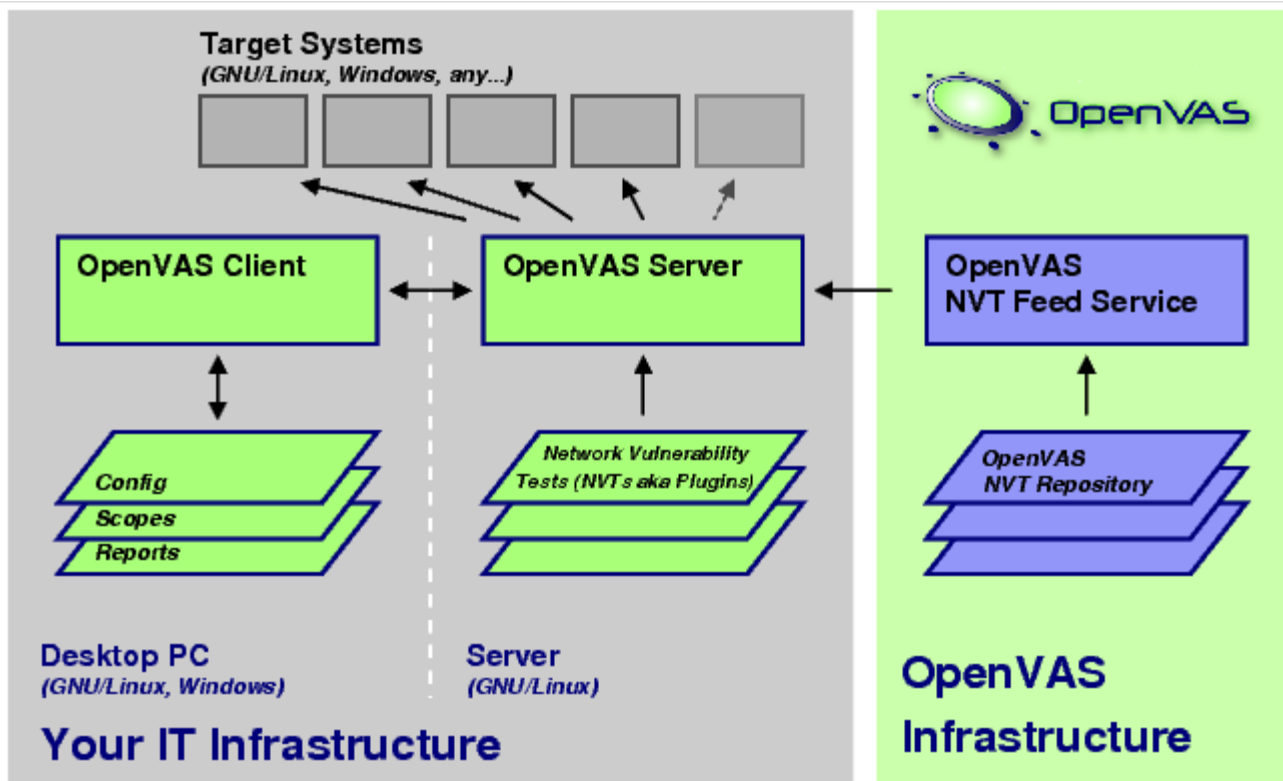
- ❑ 用户需要一种自动测试的方法，并确保正在运行一种最恰当的测试。
- ❑ **OpenVAS**包括一个中央服务器和一个图形化的前端。这个服务器准许用户运行 几种不同的网络漏洞测试（以**Nessus**攻击脚本语言编写），而且**OpenVAS**可以经常对其进行更新。**OpenVAS**所有的代码都符合**GPL**规范。

# OpenVAS的功能与所用的技术

---

- 主机扫描技术
- 端口扫描技术
- 远程主机**OS**识别
- 漏洞扫描技术
  - 这是比Nmap多的功能
  - OpenVAS自带的上万个扫描插件是其最引人注目的功能

# OpenVAS的结构



# X-scan

---

- **X-Scan**概述
- **X-scan**功能简介
- **X-scan**图形主界面
- 案例：用**X-Scan**扫描一个网段的主机



# X-Scan概述

---

- **X-Scan**是**国内最著名的综合扫描器**，它完全免费，是不需要安装的绿色软件、界面支持中文和英文两种语言、包括图形界面和命令行方式。
- 主要由国内著名的网络安全组织“安全焦点”完成，“冰河木马”的作者是其核心作者之一。
- 从**2000**年的内部测试版**X-Scan V0.2**到目前的**最新版本X-Scan V3.3**（发布日期：**07/18/2005**）都凝聚了国内众多专家的心血。
- **X-Scan**把扫描报告对扫描到的每个漏洞进行“**风险等级**”评估，并提供漏洞描述、漏洞溢出程序，方便网管测试、修补漏洞。

# X-scan功能简介

---

- 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，支持插件功能。
- **3.0**及后续版本提供了简单的插件开发包，便于有编程基础的朋友自己编写或将其他调试通过的代码修改为**X-Scan**插件。

# X-scan功能简介(2)

---

- 扫描内容包括：远程服务类型、操作系统类型及版本，各种弱口令漏洞、后门、应用服务漏洞、网络设备漏洞、拒绝服务漏洞等**20**多个大类。
- 对于多数已知漏洞，给出了相应的漏洞描述、解决方案及详细描述链接。
- 因此，**X-scan**与**Nessus**一样，也完成了**主机扫描、端口扫描、远程主机OS识别和漏洞扫描**四大功能。

# X-scan图形主界面



# 案例：用**X-Scan**扫描一个网段的主机

---

- 步骤**1** 设置扫描参数
- 步骤**2** 开始扫描
- 步骤**3** 查看扫描报告

# 1 设置扫描参数



# 1 设置扫描参数——检测范围

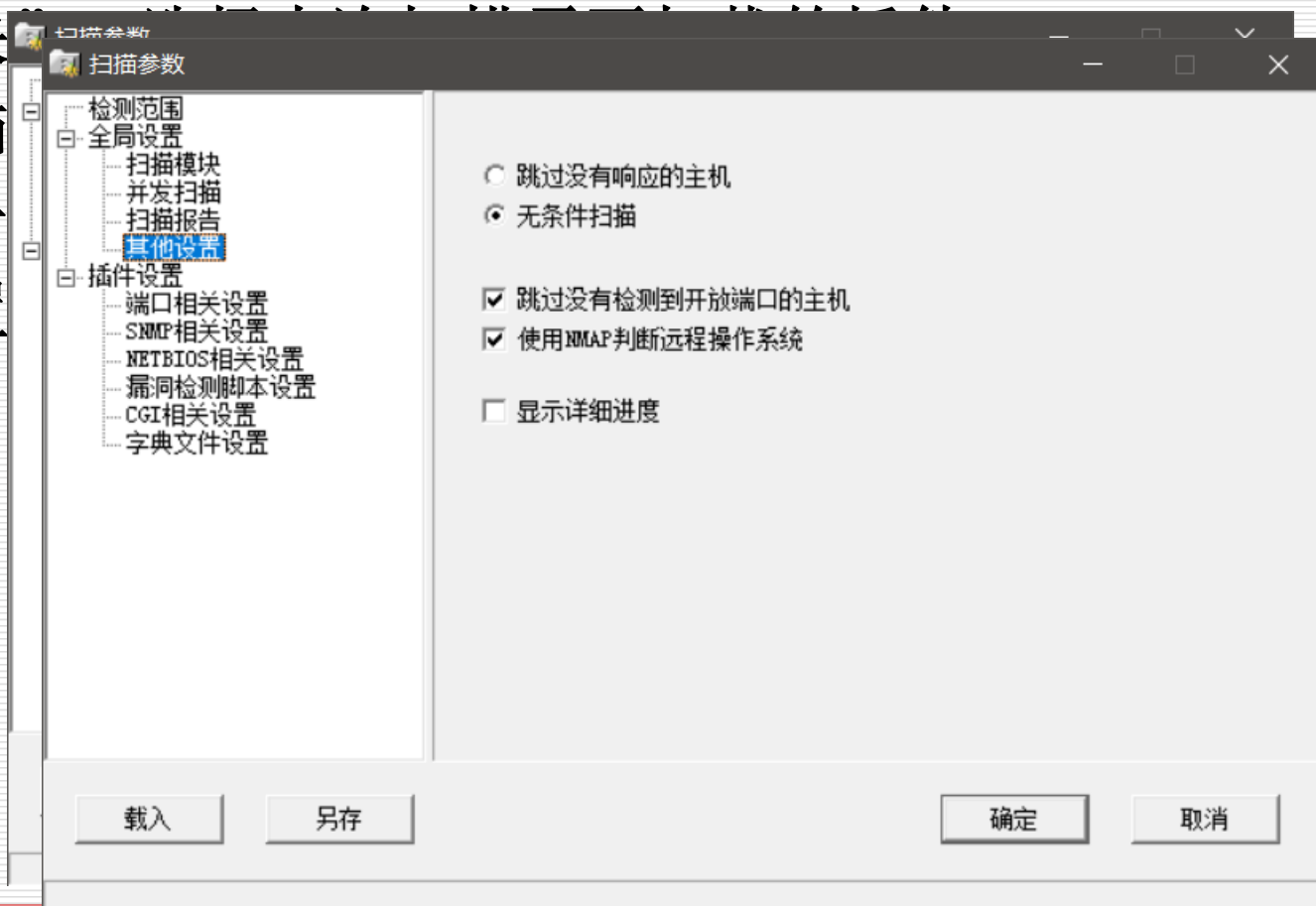
□ 在右侧窗口“指定**IP**范围”填入

- 域名或独立IP
- “-”和“,”分割IP
- 含掩码的IP地址



# 1 设置扫描参数——全局设置

- “扫描模块
- “并发扫描
- “扫描报告
- “其他设置





# 1 设置扫描参数——插件设置

□ 该模块提供对各个插件的设置方法



## 2 开始扫描

- 在此案例中，设置检测范围为**192.168.1.10-20**，其它使用默认设置，扫描过程如图所示



### 3 查看扫描报告

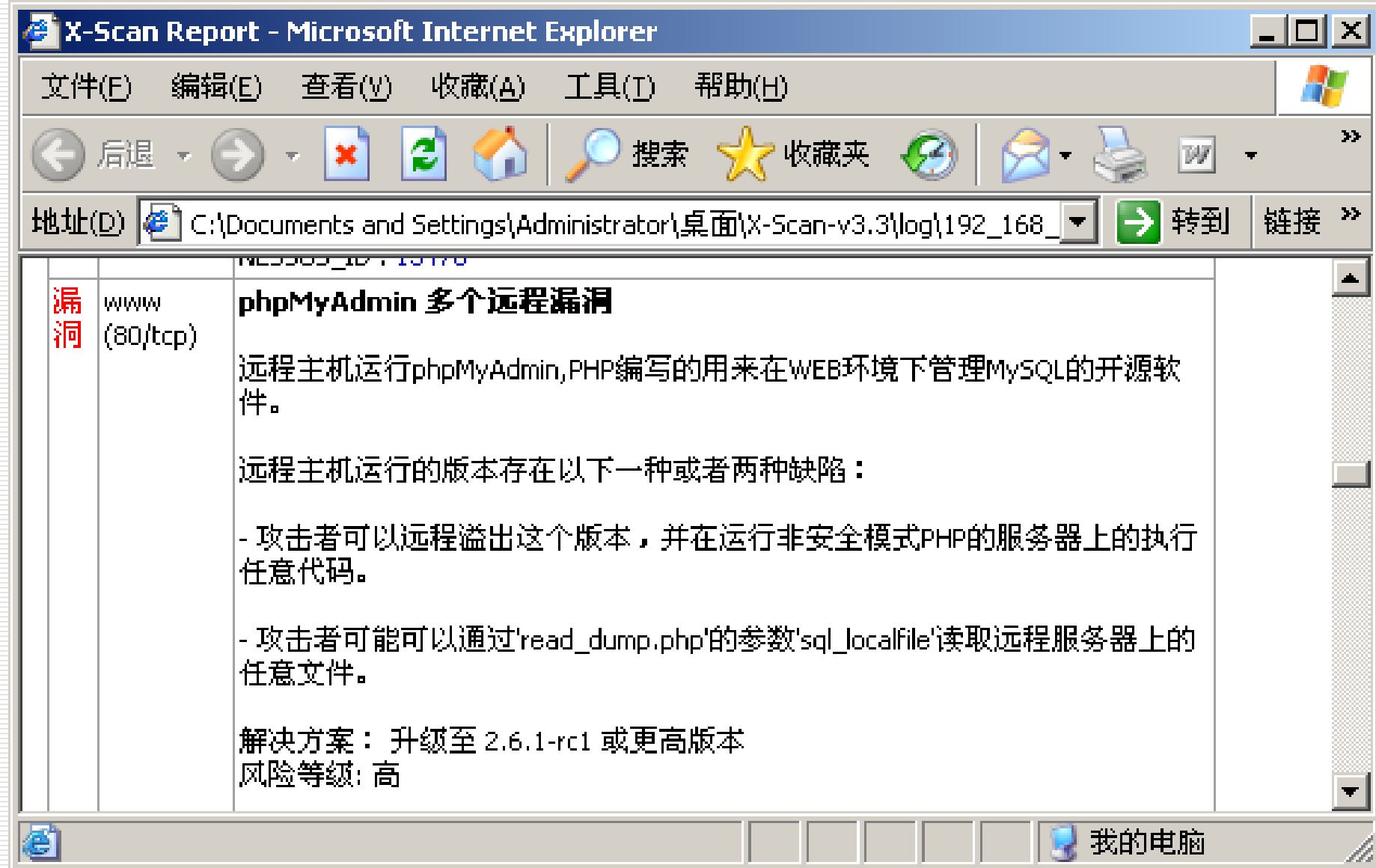
**主机分析: 192.168.1.19**

主机地址	端口/服务	服务漏洞
192.168.1.19	netbios-ssn (139/tcp)	发现安全警告
192.168.1.19	microsoft-ds (445/tcp)	发现安全漏洞
192.168.1.19	www (80/tcp)	发现安全漏洞
192.168.1.19	Windows Terminal Services (3389/tcp)	发现安全提示
192.168.1.19	ftp (21/tcp)	发现安全漏洞
192.168.1.19	pop3 (110/tcp)	发现安全提示
192.168.1.19	netbios-ns (137/udp)	发现安全提示
192.168.1.19	tcp	发现安全提示
192.168.1.19	msrdp (3389/tcp)	发现安全警告

**安全漏洞及解决方案: 192.168.1.19**

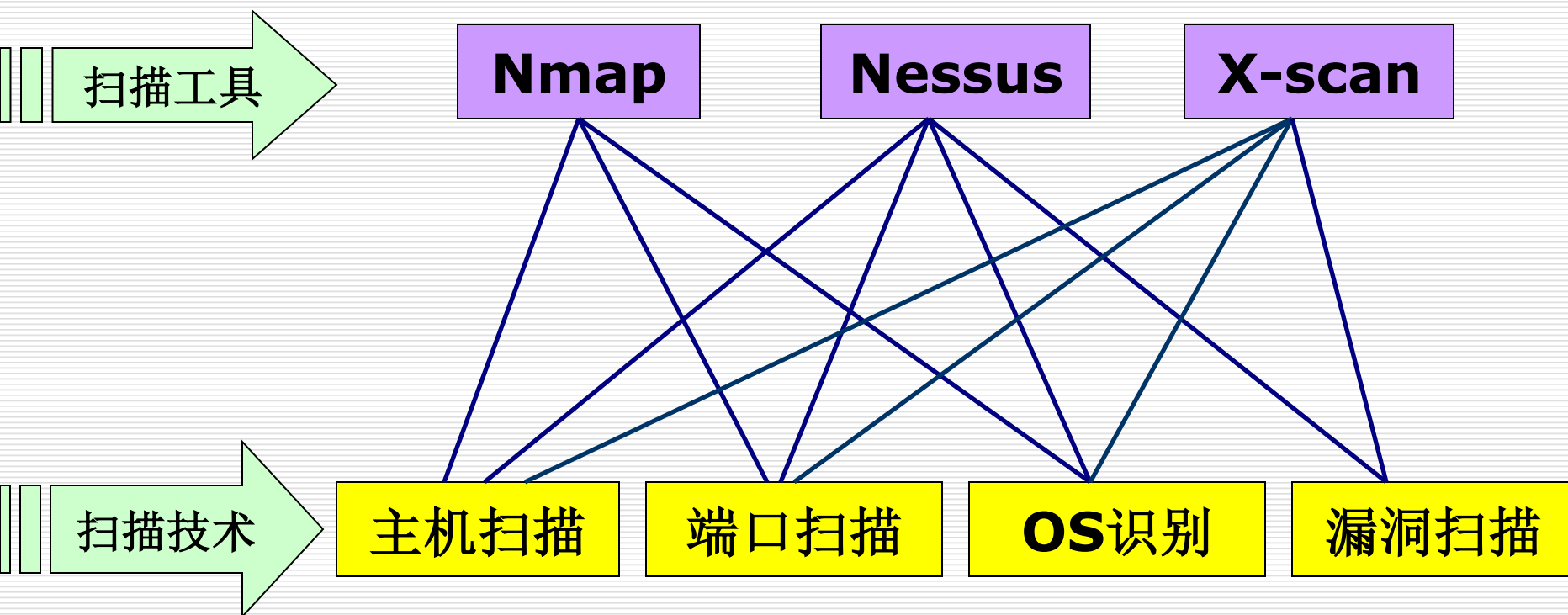
类型	端口/服务	安全漏洞及解决方案
警告	netbios-ssn (139/tcp)	<b>NetBios信息</b> [服务器信息 Level 101]: 主机名称: "192.168.1.19"

详细漏洞信息见下页图



**说明：192.168.1.19的phpMyAdmin存在高危漏洞**

# 常用扫描工具比较



# 其它扫描工具

---

- ☐ **Advance LAN Scanner**
- ☐ **Blue's PortScanner**
- ☐ **NBSI2**
- ☐ **Fluxay**（流光）
- ☐ **X-port**
- ☐ **SuperScan**
- ☐ **ISS**

## 2.4 扫描的防御

---

- 反扫描技术
- 端口扫描监测工具
- 防火墙技术
- 审计技术
- 其它防御技术



# 反扫描技术

---

- 反扫描技术是针对扫描技术提出的。
- 扫描技术一般可以分为主动扫描和被动扫描两种，它们的共同点在于在其执行的过程中都需要与受害主机互通正常或非正常的数据报文。



# 主动扫描

---

- 其中主动扫描是主动向受害主机发送各种探测数据包，根据其回应判断扫描的结果。
- 因此防范主动扫描可以从以下几个方面入手：
  - (1) 减少开放端口，做好系统防护；**
  - (2) 实时监测扫描，及时做出告警；**
  - (3) 伪装知名端口，进行信息欺骗。**

# 被动扫描

---

- 被动扫描由其性质决定，它与受害主机建立的通常是正常连接，发送的数据包也属于正常范畴，而且被动扫描不会向受害主机发送大规模的探测数据，因此其防范方法到目前为止只能采用**信息欺骗（如返回自定义的 banner 信息或伪装知名端口）**这一种方法。

# 端口扫描监测工具

---

- 对网络管理员来说，尽早的发现黑客的扫描活动，也许就能及时采取措施，避免黑客进一步实施真正的攻击和破坏。
  - 监测端口扫描的工具好多种
    - 最简单的一种是在某个不常用的端口进行监听，如果发现有对该端口的外来连接请求，就认为有端口扫描。一般这些工具都会对连接请求的来源进行反探测，同时弹出提示窗口。
    - 另一类工具，是在混杂模式下抓包并进一步分析判断。它本身并不开启任何端口。这类端口扫描监视器十分类似IDS系统中主要负责行使端口扫描监测职责的模块。
    - 蜜罐系统也是一种非常好的防御方法。
-

# 端口扫描监测工具

---

## □ 下面列出几种端口扫描的监测工具

### ■ ProtectX

- 典型的反黑工具，有几项常用功能：

- Port Sentry：用来检测外部对主机的端口扫描

- Trojan Sentry：在一些周知的木马端口监听

- Identd Server：扮演Identd服务器的角色

### ■ Winetd和DTK：蜜罐工具

### ■ Snort：轻量级的网络入侵检测系统

---

# 防火墙技术

---

- 防火墙技术是一种允许内部网接入外部网络，但同时又能识别和抵抗非授权访问的网络技术，是网络控制技术中的一种。
- 防火墙的目的是要在内部、外部两个网络之间建立一个安全控制点，控制所有从因特网流入或流向因特网的信息都经过防火墙，并检查这些信息，通过允许、拒绝或重新定向经过防火墙的数据流，实现对进、出内部网络的服务和访问的审计和控制。

# 防火墙技术(2)

---

- 个人防火墙和企业级防火墙因为其应用场景不同，也在功能、性能等方面有所差异。
  - 下面列出几种个人防火墙产品
    - ZoneAlarm Pro
    - Black ICE
    - Norton Personal Firewall
    - 天网防火墙
-

# 审计技术

---

- 审计技术是使用信息系统自动记录下的网络中机器的使用时间、敏感操作和违纪操作等，为系统进行事故原因查询、事故发生后的实时处理提供详细可靠的依据或支持。
- 审计技术可以记录网络连接的请求、返回等信息，从中识别出扫描行为。

## 审计技术(2)

---

- 以**Web**服务器为例，它的日志记录能帮助我们跟踪客户端**IP**地址，确定其地理位置信息，检测访问者所请求的路径和文件，了解访问状态，检查访问者使用的浏览器版本和操作系统类型等。
-



# 其它反扫描技术——修改**Banner**

---

- 许多网络服务器通常在用户正常连接或登录时，提供给用户一些无关紧要的提示信息，其中往往包括操作系统类型、用户所连接服务器的软件版本、几句无关痛痒的欢迎信息等，这些信息可称之为旗标信息(**Banner**)。
  - 殊不知，通过这些**Banner**黑客们可以很方便的收集目标系统的操作系统类型以及网络服务软件漏洞信息，现在很多扫描器如**Nmap**都具备了自动获取**Banner**的功能。可以对**Banner**进行修改，隐藏主机信息，减小被入侵的风险。
-

# 其它反扫描技术——修改Banner

---

## □ 修改Banner的方法：

- 修改网络服务的配置文件，许多服务都在其配置文件中提供了对显示版本号的配置选项；
- 修改服务软件的源代码，然后重新编译；
- 直接修改软件的可执行文件，这种方法往往具有一定的“危险性”，不提倡使用。当然，也可以利用一些专业的Banner修改工具。

## □ 下面以Linux系统中的几个服务器为例，说明一些典型的Banner信息修改方法，更多的技巧还要在实际的网络管理与操作中去总结和体会。

---

# 其它反扫描技术——修改Banner

---

## □ wu-ftp服务器

- 如果没有修改Banner，用户连接时，它会提供版本信息。
- Banner修改方法是在wu-ftp的配置文件/etc/ftppass中添一行：

greeting text<New Banners Information>

//New Banners Information是指新的Banner信息，由用户自己设置

---

# 其它反扫描技术——修改Banner

---

## □ telnet服务器

- 正常情况下，成功连接telnet服务器后，会返回操作系统的类型信息。
  - 用户从网络登录时看到的提示信息存放在/etc/issue文件中（文件/etc/issue.net中的内容与之相同），因此可以通过修改这两个文件里的登录提示信息来抹掉原来所提供的真实的操作系统类型信息。
  - 由于每次系统启动时，都会通过/etc/rc.d/rc.local文件重新创建这两个文件，所以可以在/etc/rc.d/rc.local文件中修改输出到两个文件中的内容，或者直接注释掉/etc/rc.d/rc.local文件中相应的脚本行。
-

# 其它反扫描技术——修改Banner

---

## □ **Sendmail**服务器

- 在Sendmail的配置文件/etc/sendmail.cf中，有这样一行：O SmtgGreetingMessage=\$j Sendmail \$v/\$Z;\$b
- 修改其中的\$v\$Z即可。然后重新启动服务。

## □ **DNS**服务器（**bind**）

- 在bind的配置文件 /usr/local/named/etc/named.conf的options里找到version，修改version字段内容，就可以修改这一Banner信息。
  - 重新启动DNS服务器，使改动生效。
-

## 2.5 小结

---

- 扫描器能够自动的扫描检测本地和远程系统的弱点，为用户提供帮助。系统或网络管理员可以利用它来检测其所管理的网络和主机中存在哪些漏洞，以便及时打上补丁，增加防护措施，或用来对系统进行安全等级评估。黑客可以利用它来获取主机信息，寻找具备某些弱点的主机，为进一步攻击做准备。因此，扫描器是一把双刃剑。
  - 用户要减少开放的端口，关闭不必要的服务，合理地配置防火墙，以防范扫描行为。
-

# Socket 编程

---

- 如果想自己编写扫描器，首先必须熟悉 **Socket**编程。
- 对于初学者，在**Linux**下的**socket**编程可以参考 **W.Richard Stevens**的几本著作：
  - 1: 《**unix**网络编程》 清华大学出版社
  - 2: 《**unix**环境高级编程》 简称**APUE**，机械工业出版社

# tcp full scan example

```
1.#! -*- coding:utf-8 -*-
2.import time
3.import socket
4.
5.socket_timeout = 0.1
6.
7.
8.def tcp_scan(ip, port):
9.    try:
10.        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11.        s.settimeout(socket_timeout)
12.        c = s.connect_ex((ip, port))
13.        if c == 0:
14.            print("%s:%s is open" % (ip, port))
15.        else:
16.            # print "%s:%s is not open" % (ip,port)
17.            pass
18.    except Exception as e:
19.        print(e)
20.
21.    finally:
22.        s.close()
```



## 本章参考文献

---

- 张玉清等，安全扫描技术，清华大学出版社，2004
- <http://www.cert.org.cn/>
- <http://www.cnnic.net.cn/>

---

谢谢各位!