



网络与系统安全

第五章 网络扫描

黄 珂



- 访问控制是（操作）系统安全的基础
- 局域网的安全管理是网络安全的**网络基础**
 - 任何网络层加密数据在一个**不安全的**局域网中都可能被嗅探
 - 攻击者一旦渗透进入内部网络，后果不堪设想
 - 内网安全先从管好ARP协议开始
- 网络监听是**被动分析**网络安全的重要手段



- 网络扫描是主动分析网络安全的重要手段
 - 对于黑客来说
 - 网络扫描是网络入侵的序曲
 - 网络扫描是信息收集的手段之一
 - 对于网络管理员来说
 - 网络扫描是网络安全防御的自我检测手段



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



- 信息收集
 - 知己知彼，百战不殆
 - 语出《孙子·谋攻》
 - 知己知彼，百战不殆；不知彼而知己，一胜一负；不知彼，不知己，每战必殆。
- 信息收集的目标对象包括
 - 目标主机
 - 目标网络
 - 目标应用/服务
 - 目标人



信息收集之网络扫描

- 网络扫描是实现信息收集目的的手段之一
 - 目标主机
 - 在线状态
 - 端口开放情况
 - 网络服务详细信息
 - 网络拓扑
 - 目标应用/服务
 - 版本信息
 - 是否存在漏洞



信息收集之踩点

- 直接访问
 - 目标主机
 - 目标网络
 - 目标应用/服务
- 黑盒测试
 - 使用特定客户端连接指定端口/应用/服务
 - 浏览器 / FTP / telnet
 - 使用特定帐号和口令尝试登录
 - 模仿交互



- 目标人
 - 社交网络应用信息
 - 微博 / SNS / blog ...
 - **Google Hacking**
 - site: / filetype: / inurl:
 - 钓鱼
 - 电子邮件 / 即时通信 / 电话 ...



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解

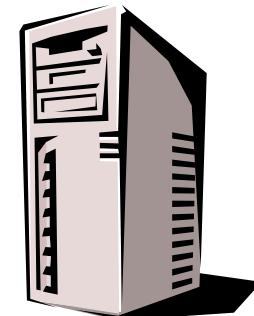


网络扫描原理

- 网络扫描的基本思想
- 网络扫描的基本原理
- 网络扫描的主要实现技术

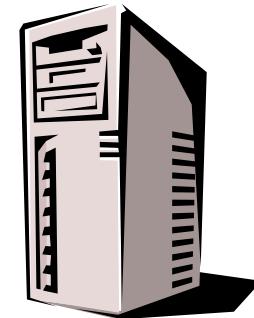


网络扫描的基本思想





网络扫描的基本思想

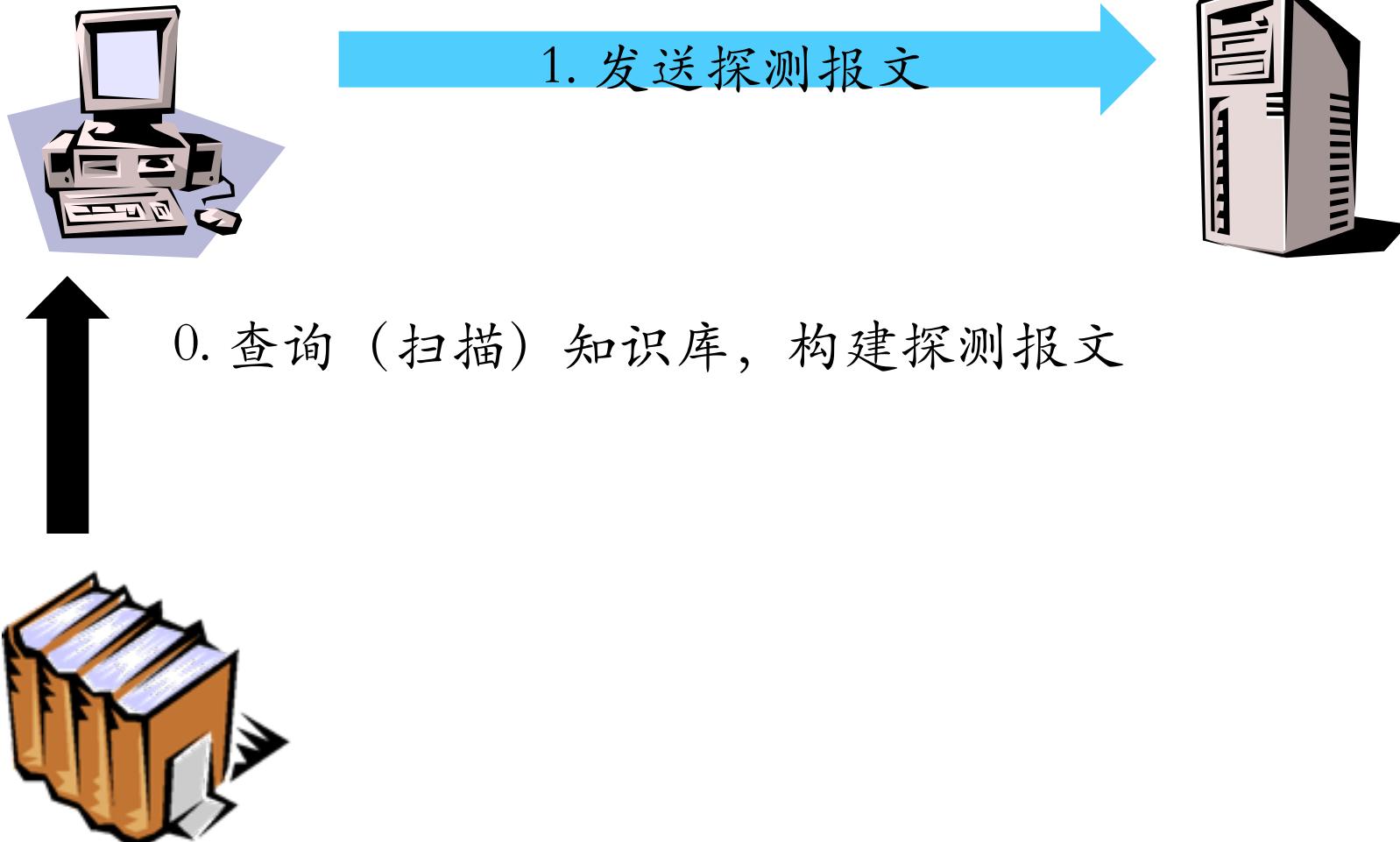


0. 查询（扫描）知识库，构建探测报文



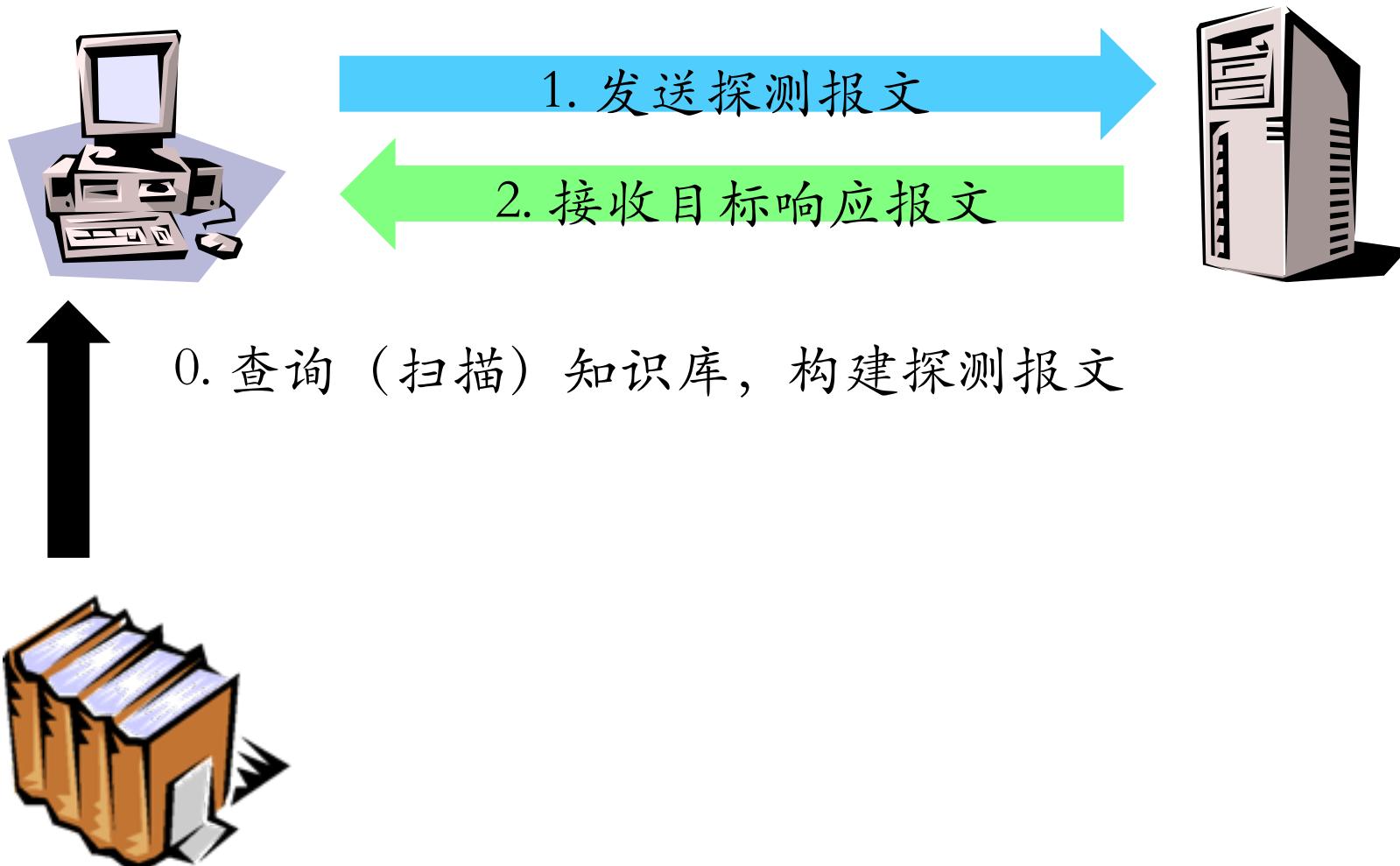


网络扫描的基本思想



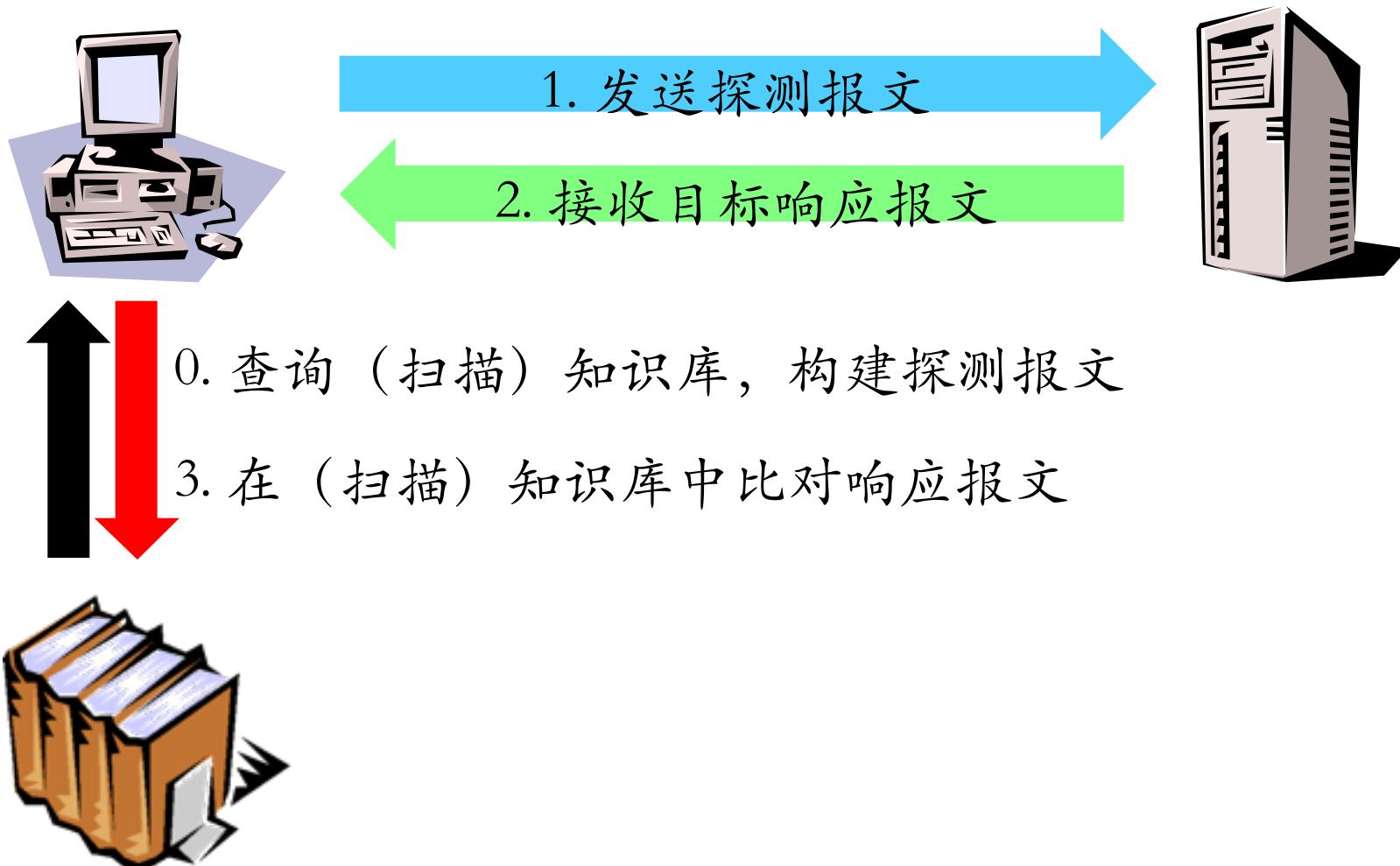


网络扫描的基本思想



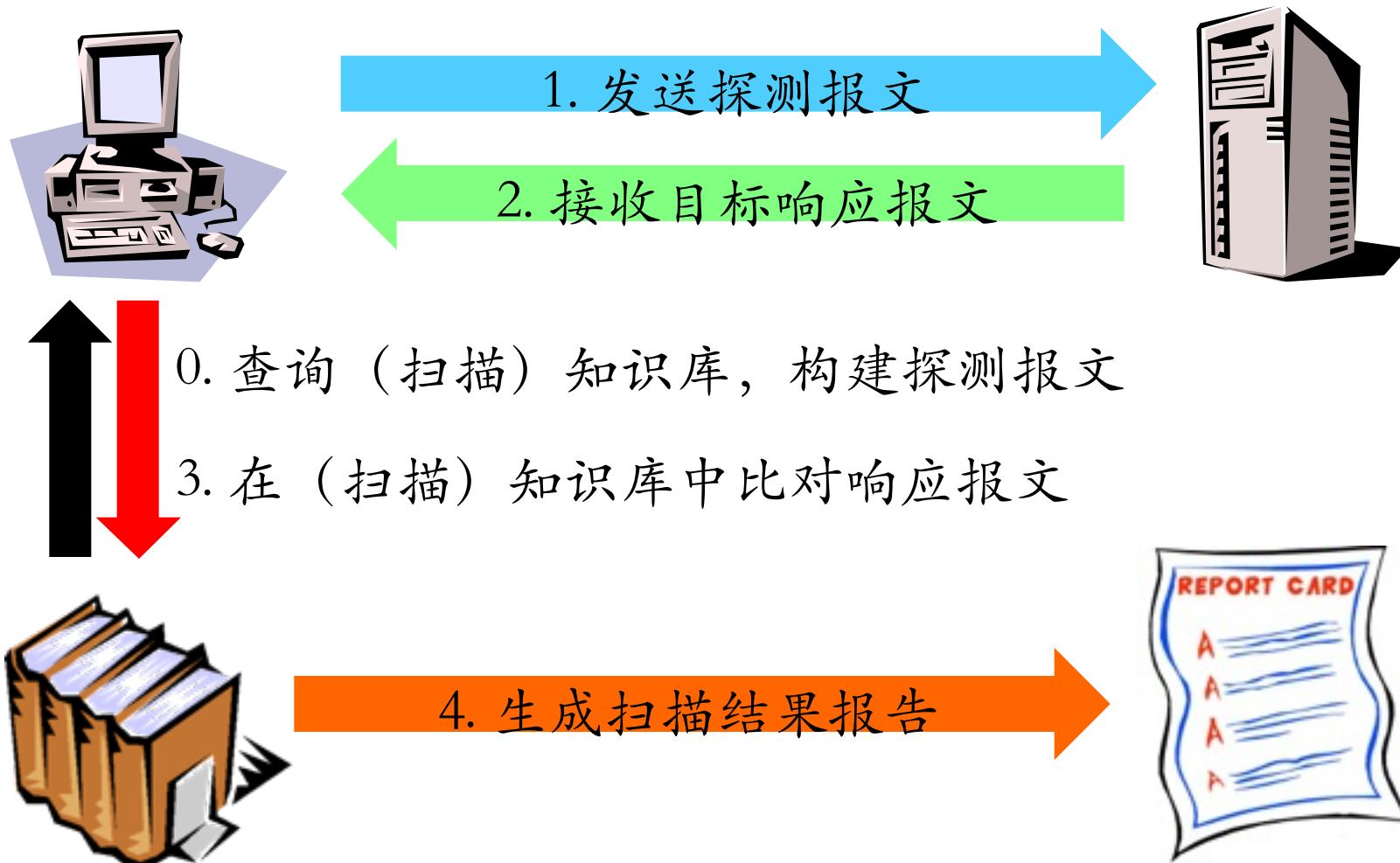


网络扫描的基本思想





网络扫描的基本思想





网络扫描的基本原理

- 报文发送与接收
- 扫描知识库构建与规则匹配
- 扫描报告生成



网络扫描的基本原理

- 报文发送与接收
- 扫描知识库构建与规则匹配
- 扫描报告生成

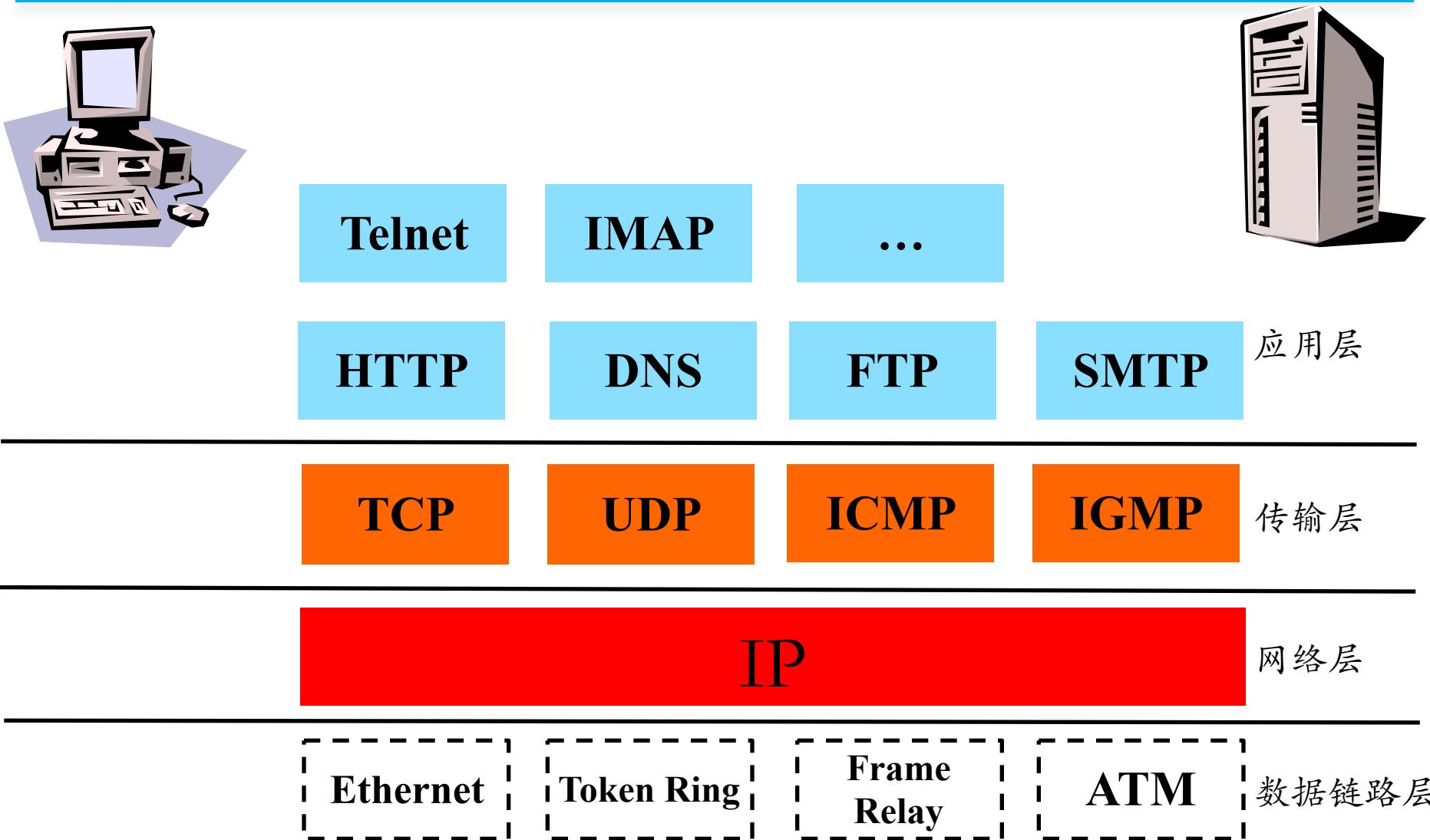


网络扫描的基本原理——报文发送与接收

- TCP/IP协议栈标准
- 传输层协议基础回顾
 - TCP协议
 - UDP协议
 - ICMP协议
- 协议标准和（操作系统）协议栈实现的关系

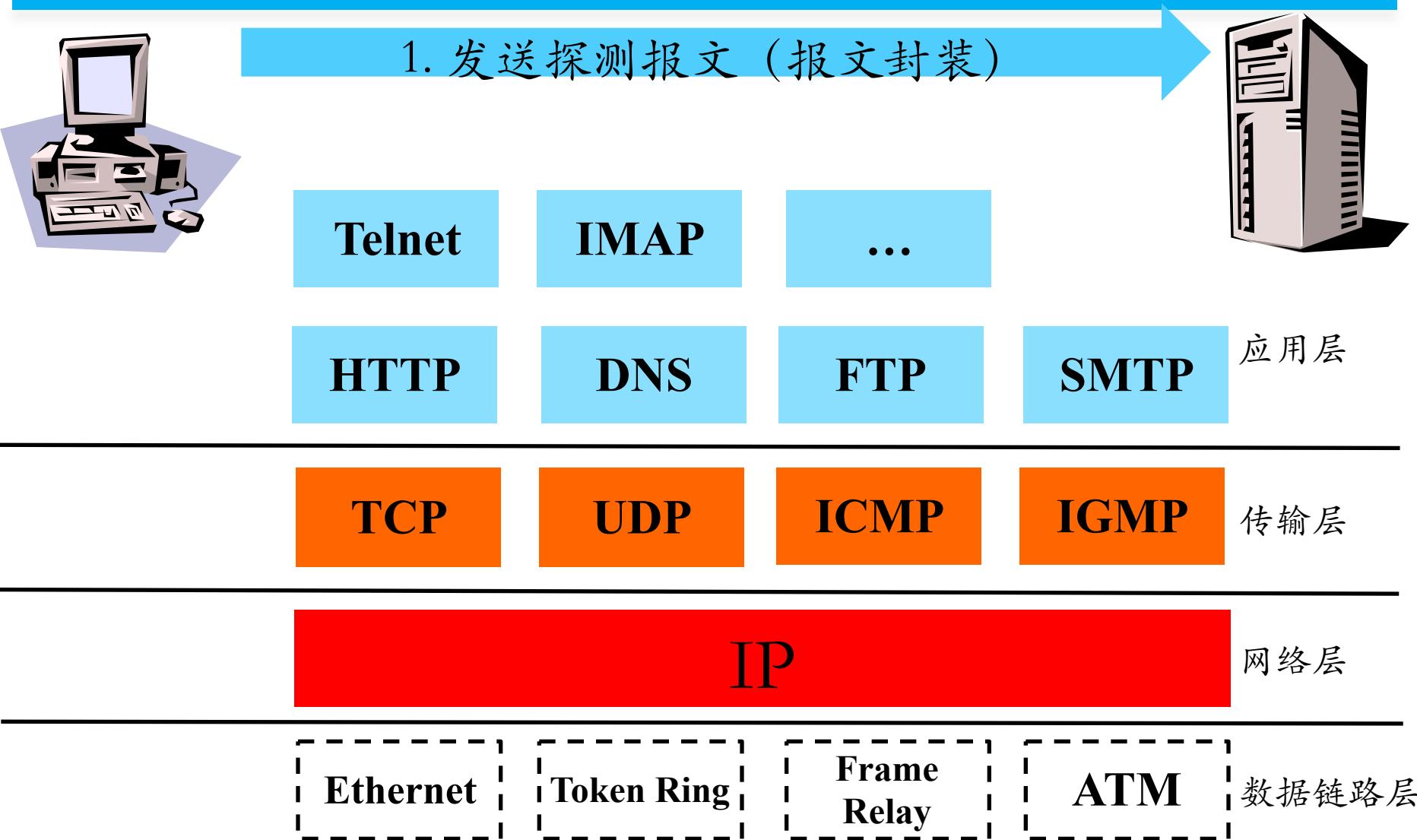


网络扫描的基本原理——TCP/IP协议栈标准



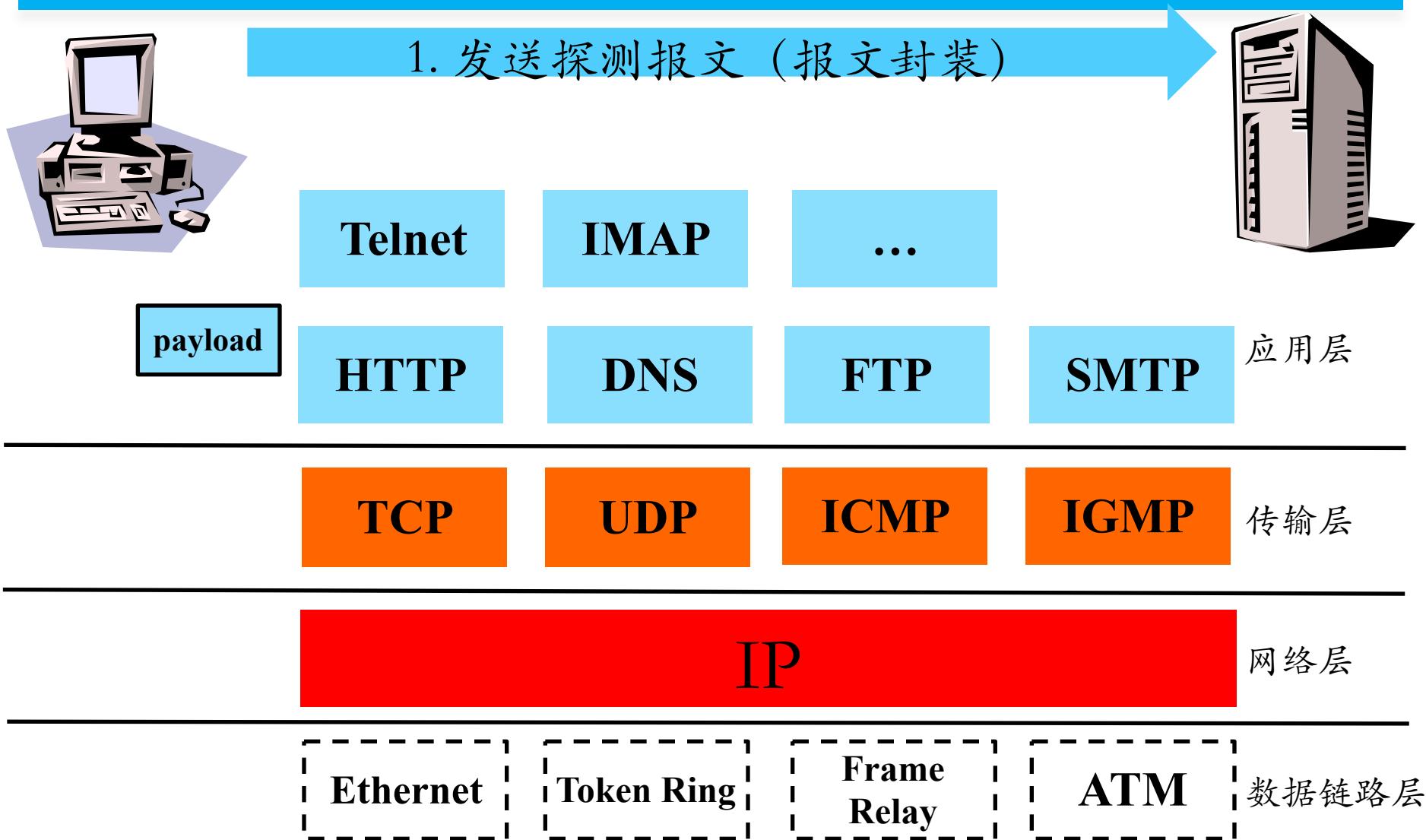


网络扫描的基本原理——TCP/IP协议栈标准



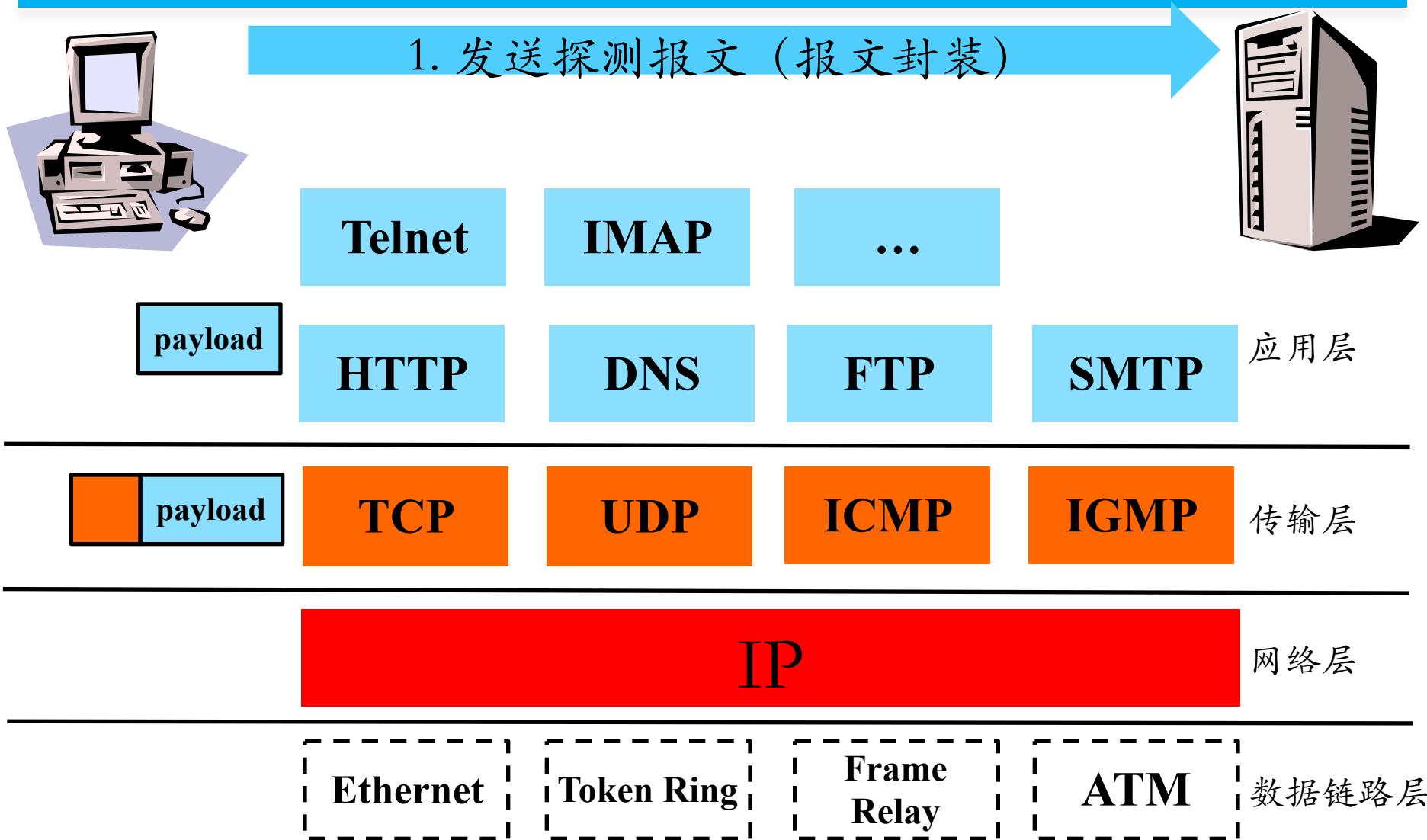


网络扫描的基本原理——TCP/IP协议栈标准



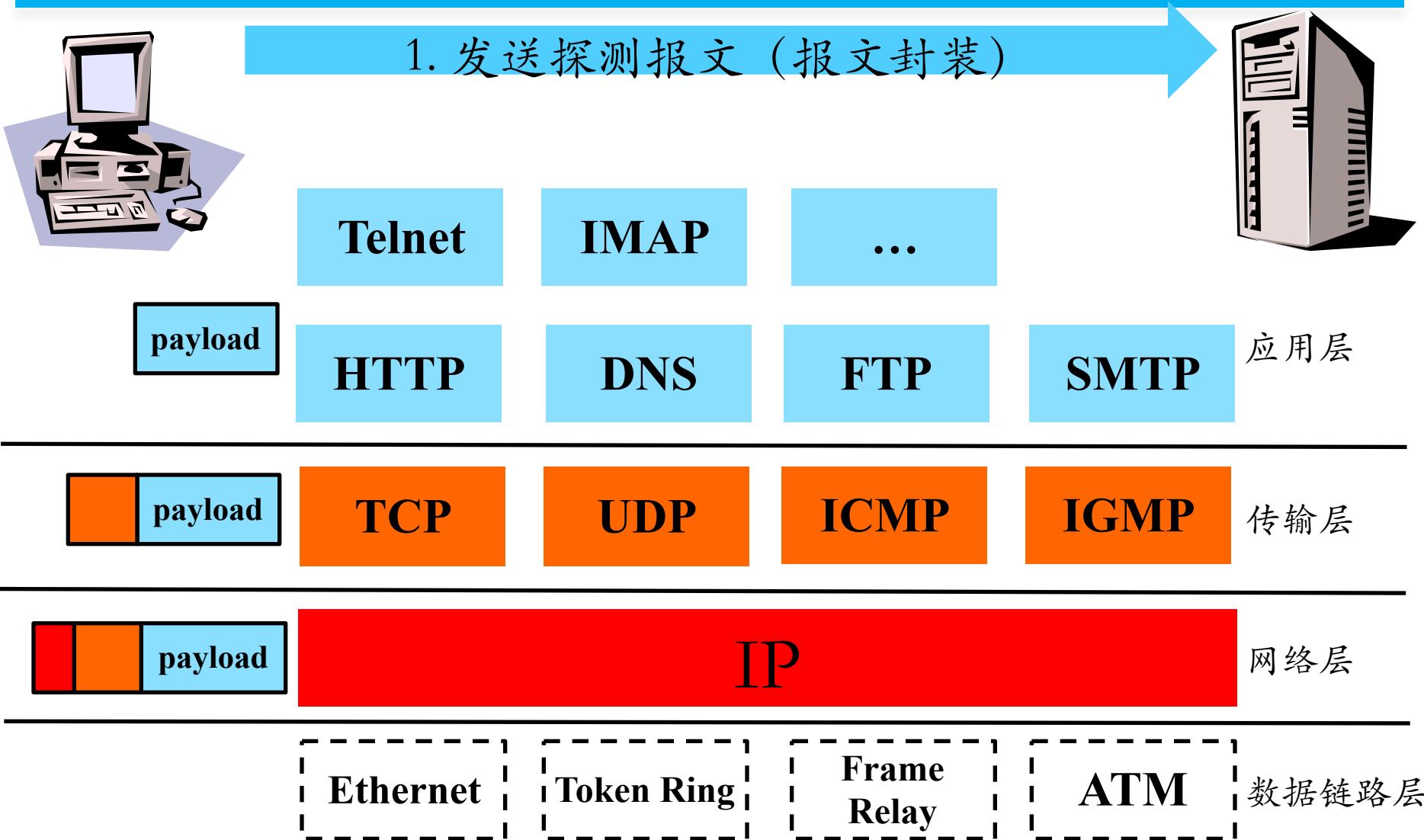


网络扫描的基本原理——TCP/IP协议栈标准





网络扫描的基本原理——TCP/IP协议栈标准

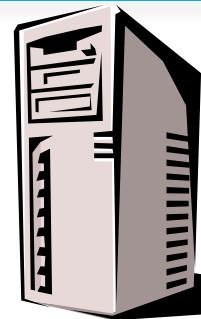




网络扫描的基本原理——TCP/IP协议栈标准



1. 发送探测报文（报文封装）



Telnet

IMAP

...

payload

HTTP

DNS

FTP

SMTP

应用层

payload

TCP

UDP

ICMP

IGMP

传输层

payload

IP

网络层

payload

Ethernet

Token Ring

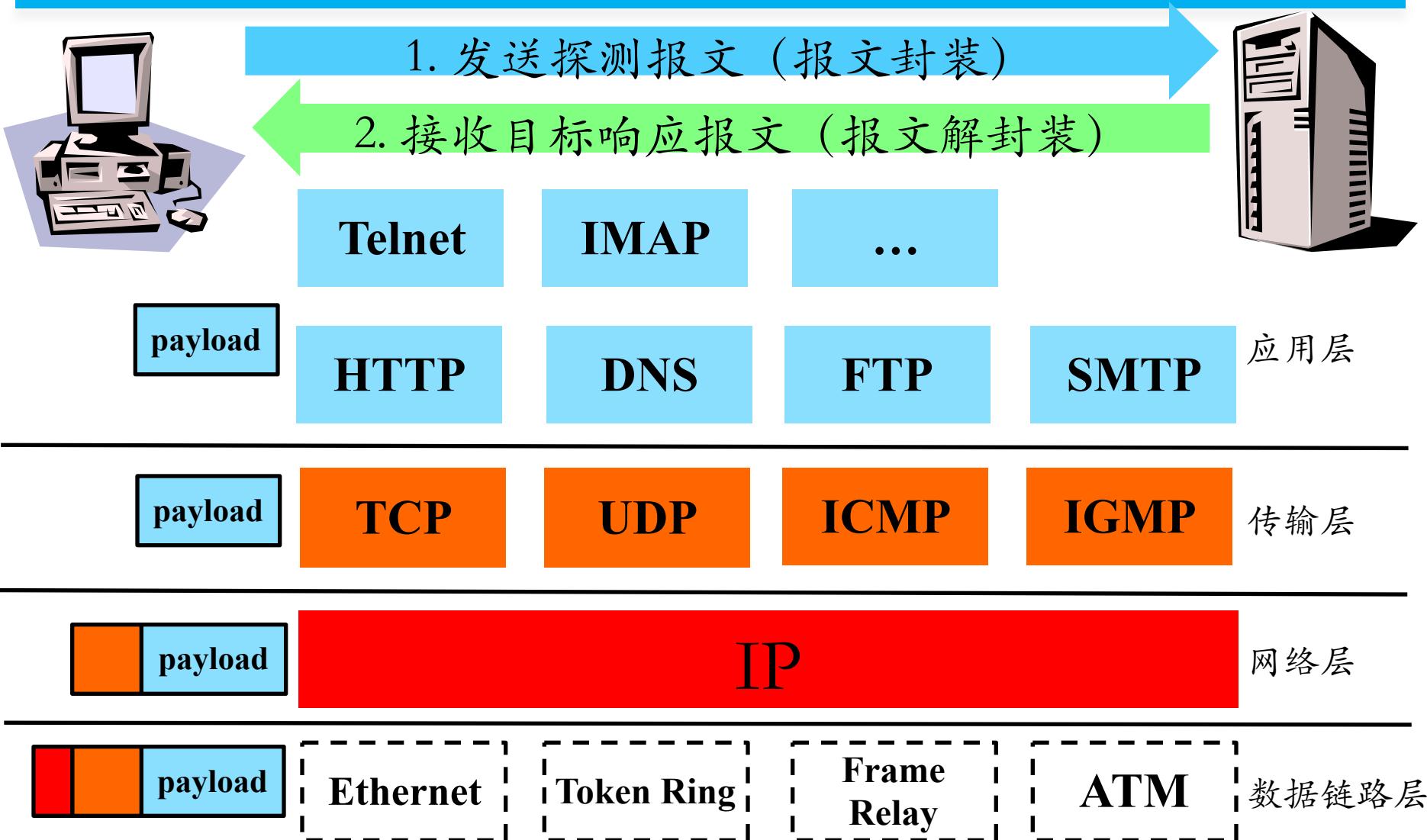
Frame Relay

ATM

数据链路层

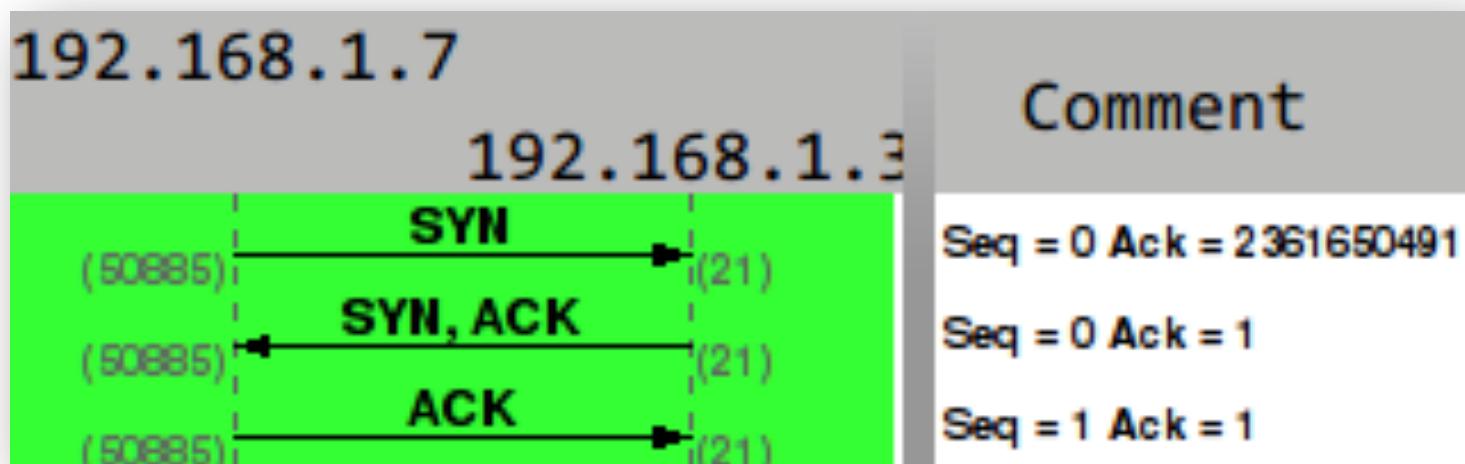


网络扫描的基本原理——TCP/IP协议栈标准



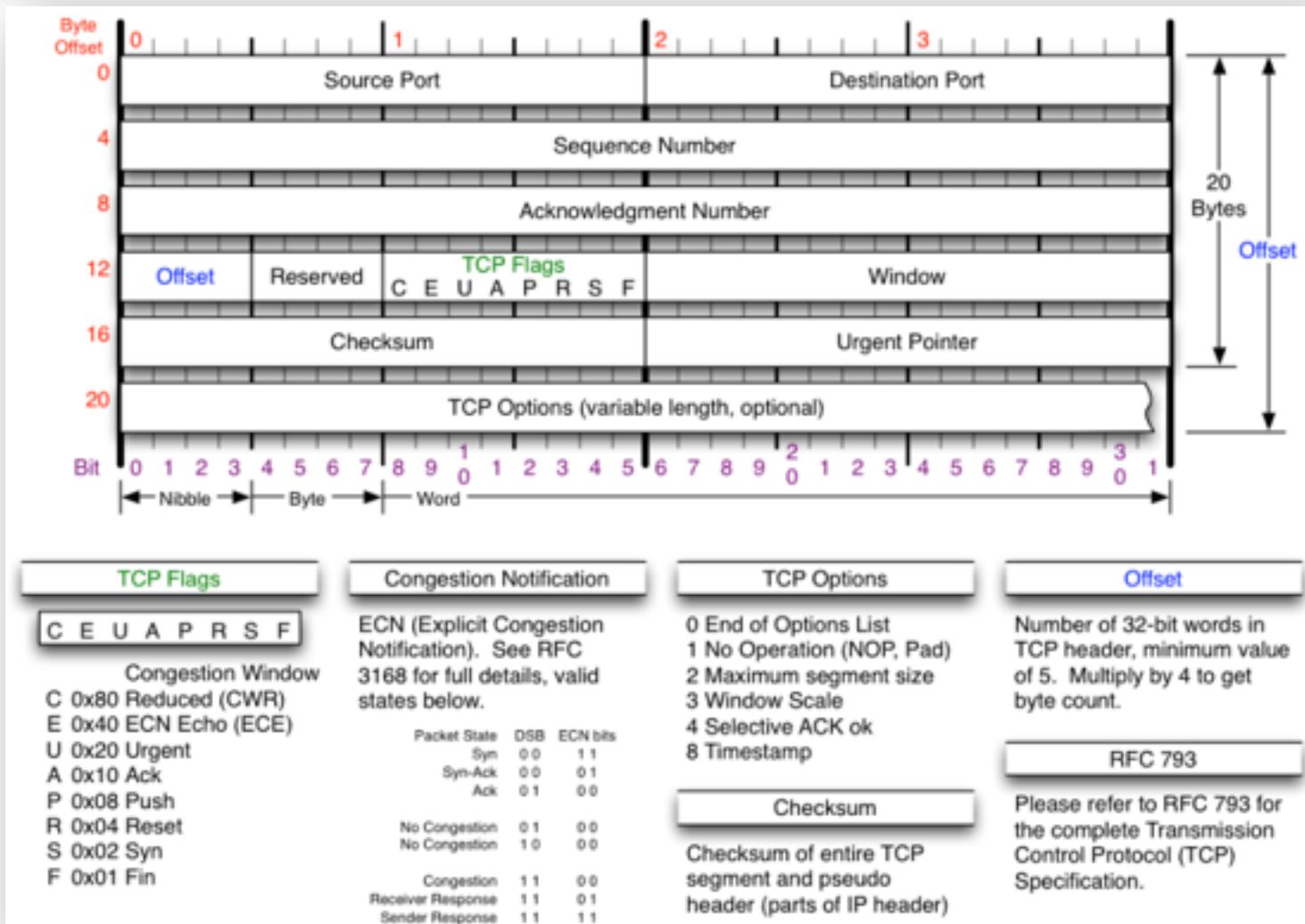


- Transmission Control Protocol
- TCP是一种面向连接的，可靠的传输层协议
- TCP建立连接过程称为三次握手



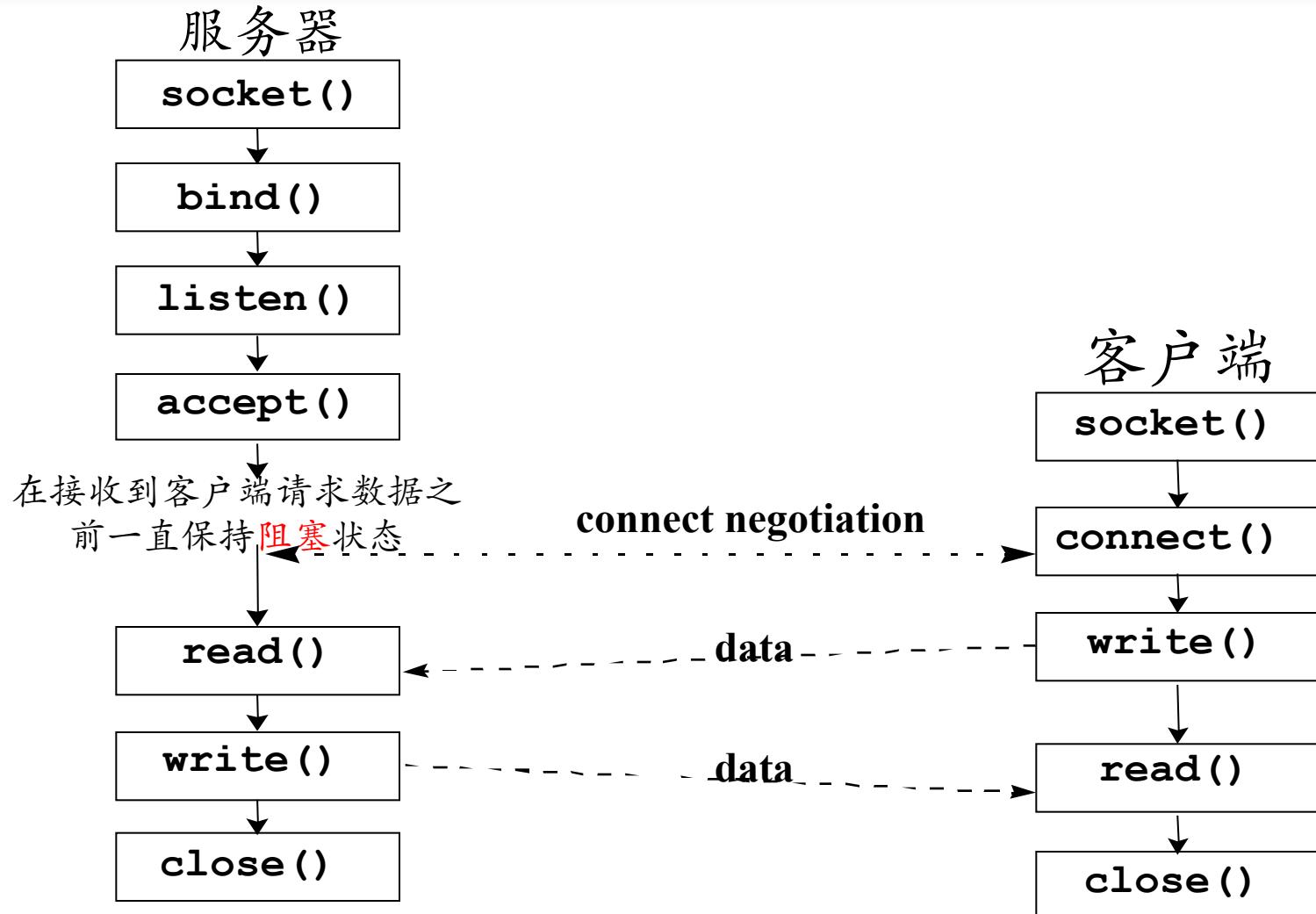


TCP报文头部格式





TCP协议栈实现举例

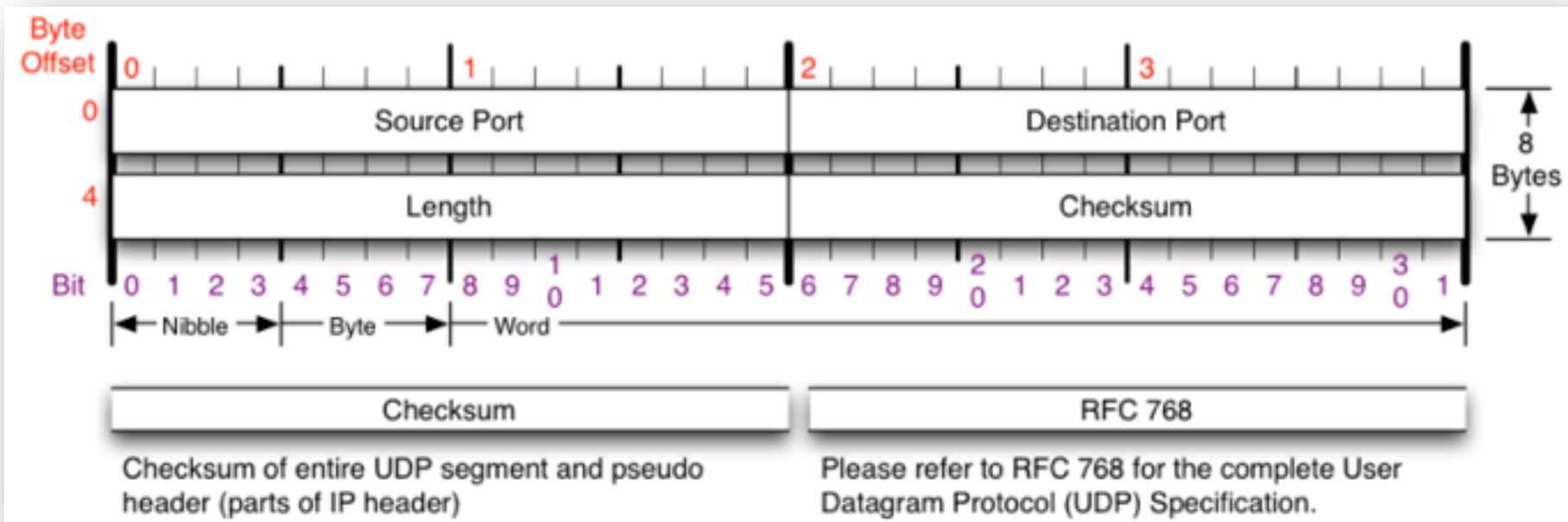




- **User Datagram Protocol**
- UDP是一种提供面向事务的简单不可靠信息传送服务
 - 无连接协议：源和目的端在数据传输之前不建立连接
 - 收发双方均无需维护连接状态信息
 - 应用层按需维护连接状态信息
 - 尽力而为
 - 相较于TCP协议的操作系统协议栈实现而言

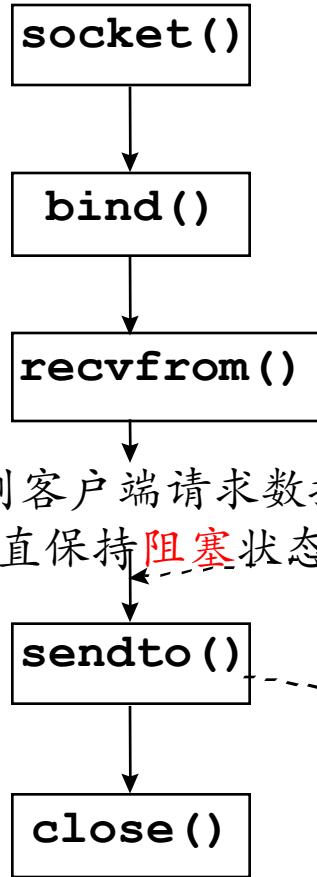


UDP报文头部格式

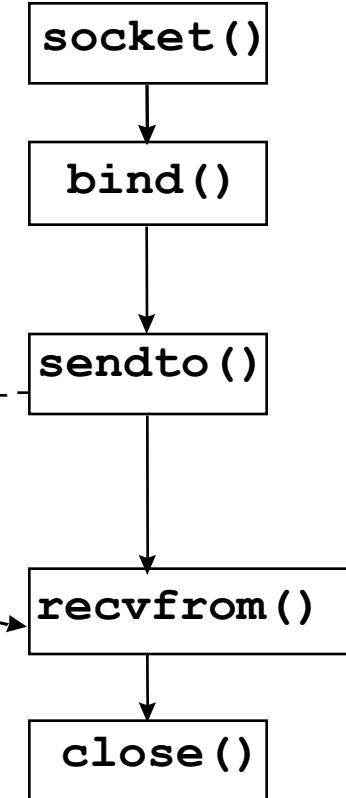




服务器



客户端

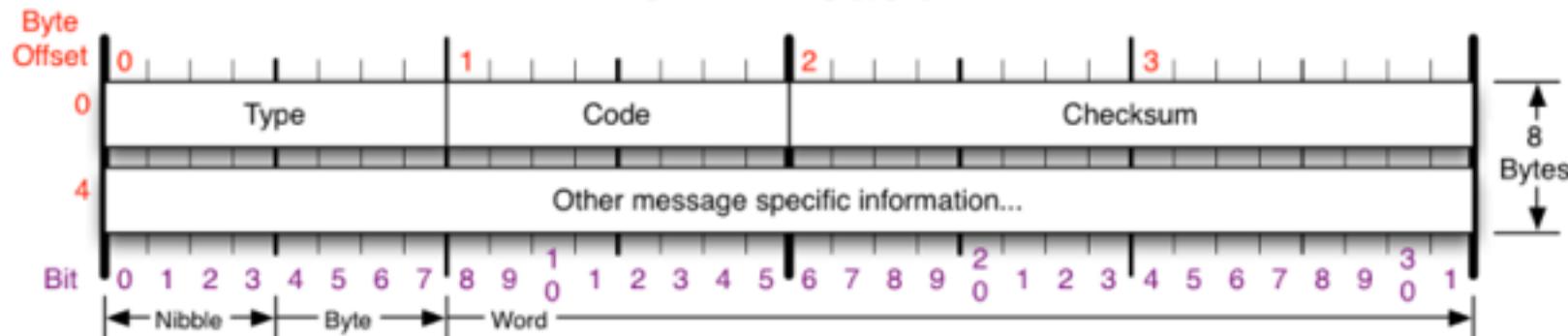




- Internet Control Message Protocol
- 用途
 - 网关或者目标机器利用**ICMP**与源通讯
 - 当出现问题时，提供反馈信息用于报告错误
- 特点
 - 其控制能力并不用于保证传输的可靠性
 - 它本身也不是可靠传输的
 - 并不用来反映**ICMP**报文的传输情况



ICMP报文头部格式



ICMP Message Types

Type	Code/Name
0	Echo Reply
3	Destination Unreachable
0	Net Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation required, and DF set
5	Source Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Network Administratively Prohibited
10	Host Administratively Prohibited
11	Network Unreachable for TOS

Type	Code/Name
3	Destination Unreachable (continued)
12	Host Unreachable for TOS
13	Communication Administratively Prohibited
4	Source Quench
5	Redirect
0	Redirect Datagram for the Network
1	Redirect Datagram for the Host
2	Redirect Datagram for the TOS & Network
3	Redirect Datagram for the TOS & Host
8	Echo
9	Router Advertisement
10	Router Selection

Type	Code/Name
11	Time Exceeded
0	TTL Exceeded
1	Fragment Reassembly Time Exceeded
12	Parameter Problem
0	Pointer Problem
1	Missing a Required Operand
2	Bad Length
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
30	Traceroute

Checksum

Checksum of ICMP header

RFC 792

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.



协议标准和（操作系统）协议栈的关系

- 协议标准（声明）

- 严格统一和规范标准

- RFC 793 TRANSMISSION CONTROL PROTOCOL
 - RFC 768 User Datagram Protocol
 - RFC 791 INTERNET PROTOCOL
 - RFC 792 Internet Control Message Protocol

- 协议栈（实现）

- 严格遵循标准实现

- 标准未尽之处的实现会略有差异



Winsock API

UNIX®



LINUX
POWERED

Berkeley sockets



网络扫描的基本原理

- 报文发送与接收
- 扫描知识库构建与规则匹配
- 扫描报告生成



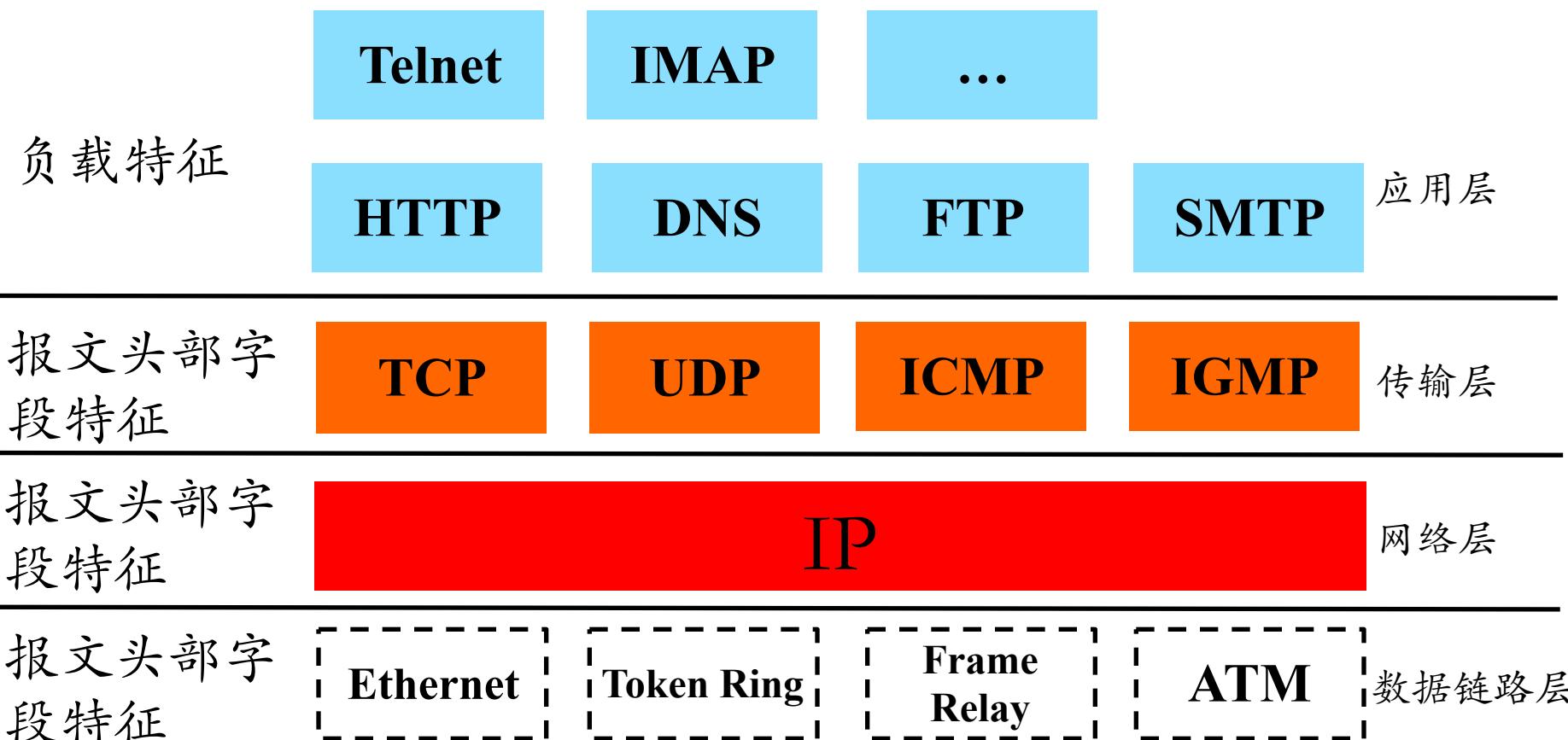
网络扫描的基本原理

- 报文发送与接收
- 扫描知识库构建与规则匹配
- 扫描报告生成



扫描知识库的构建原理

- 每一层都可以构建独立知识库





扫描知识库构建举例——IANA注册端口号

- 传输层报文头部字段特征
- 公用端口
 - 0 到 1023
- 注册端口
 - 1024 到 49151
- 动态的或私有端口
 - 49152 到 65535

*端口号注册流程参考：RFC4340，章节19.9



扫描知识库构建举例——/etc/services

```
43 domain          53/tcp           # name-domain server
44 domain          53/udp
45 mtp             57/tcp           # deprecated
46 tacacs-ds       65/tcp           # TACACS-Database Service
47 tacacs-ds       65/udp
48 bootps          67/tcp           # BOOTP server
49 bootps          67/udp
50 bootpc          68/tcp           # BOOTP client
51 bootpc          68/udp
52 tftp             69/udp
53 gopher          70/tcp           # Internet Gopher
54 gopher          70/udp
55 rje              77/tcp           netrjs
56 finger           79/tcp
57 www              80/tcp           http      # WorldWideWeb HTTP
58 www              80/udp           # HyperText Transfer Protocol
```



端口状态

- **开放** ●
 - 应用/服务监听该端口
 - 有条件有规则地响应请求数据报文
- **关闭** ●
 - 无应用/服务监听该端口
 - 有条件有规则地响应或忽略请求数据报文
 - 操作系统会针对SYN请求报文回应RST报文
- **被过滤** ●
 - 报文过滤程序监听该端口
 - 有条件有规则地响应或忽略请求数据报文
 - 报文过滤程序可能会返回报文拒绝消息



端口状态

- **开放** ●
 - 应用/服务监听该端口
 - **有条件有规则地响应请求数据报文**
- **关闭** ●
 - 无应用/服务监听该端口
 - **有条件有规则地响应或忽略请求数据报文**
 - 操作系统会针对SYN请求报文回应RST报文
- **被过滤** ●
 - 报文过滤程序监听该端口
 - **有条件有规则地响应或忽略请求数据报文**
 - 报文过滤程序可能会返回报文拒绝消息

端口状态
知识库构建的基础



- 可达（在线）
 - 对至少一种类型的请求数据包有响应（充要条件）
 - 有开放端口（充分非必要条件）
- 不可达（离线）
 - 对任何类型的请求数据包均无响应（充分条件）
 - 无开放端口（必要非充分条件）
 - 例如普通PC机，在线时但无开放端口
 - 在线主机在防火墙的保护下也可能是不可达状态



- 可达（在线）
 - 对至少一种类型的请求数据包有响应（充要条件）
 - 有开放端口（充分非必要条件）
 - 不可达（离线）
 - 对任何类型的请求数据包均无响应（充分条件）
 - 无开放端口（必要非充分条件）
 - 例如普通PC机，在线时但无开放端口
 - 在线主机在防火墙的保护下也可能是不可达状态
- 注意：端口状态是传输层的概念，



- 可达（在线）
 - 对至少一种类型的请求数据包有响应（充要条件）
 - 有开放端口（充分非必要条件）
 - 不可达（离线）
 - 对任何类型的请求数据包均无响应（充分条件）
 - 无开放端口（必要非充分条件）
 - 例如普通PC机，在线时但无开放端口
 - 在线主机在防火墙的保护下也可能是不可达状态
- 注意：端口状态是传输层的概念，
不要和网络层、数据链路层、物理层可达混淆



主机状态详细信息

- 操作系统信息
 - 版本
- 端口/应用/服务状态信息
 - 端口状态
 - 应用程序版本
- 远程获取主机状态详细信息的基础
 - 主机扫描
 - TCP/IP 协议栈实现知识库
 - 不同操作系统/应用程序的差异



主机扫描技术——可达状态检测基本技术

- 局域网
 - ARP**扫描
- 广域网
 - ICMP Echo**扫描
 - ICMP Sweep**扫描
 - ICMP Broadcast**扫描
 - ICMP Non-Echo**扫描



局域网主机可达状态检测——ARP扫描

- ARP广播请求

—局域网连通状态主机必定会响应正常ARP广播请求

No.	Time	Source	Destination	Protocol.	Info
1	0.000000	f0:4d:a2:2a:61:26	Broadcast	ARP	Who has 222.31.66.220? Tell 222.31.66.224
2	0.001709	Msi_ad:e9:18	f0:4d:a2:2a:61:26	ARP	222.31.66.220 is at 00:16:17:ad:e9:18
3	0.001717	222.31.66.224	222.31.66.220	ICMP	Echo (ping) request
4	0.001875	222.31.66.220	222.31.66.224	ICMP	Echo (ping) reply

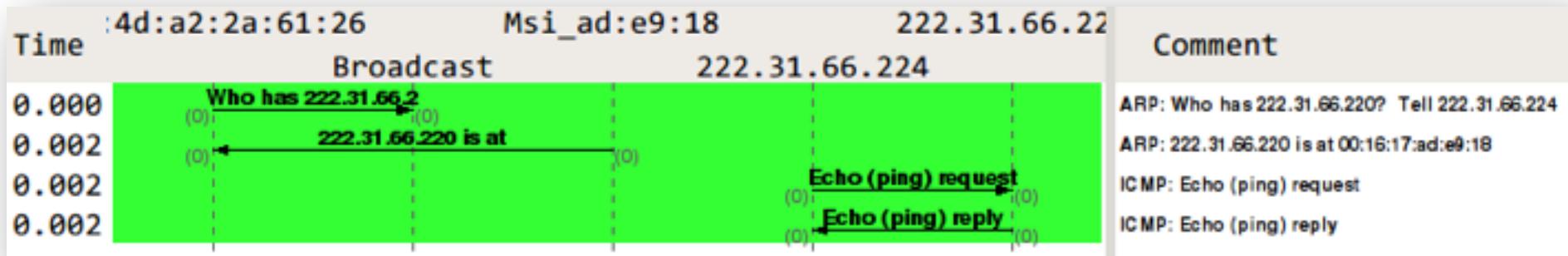


局域网主机可达状态检测——ARP扫描

- ARP广播请求

—局域网连通状态主机必定会响应正常ARP广播请求

No.	Time	Source	Destination	Protocol.	Info
1	0.000000	f0:4d:a2:2a:61:26	Broadcast	ARP	Who has 222.31.66.220? Tell 222.31.66.224
2	0.001709	Msi_ad:e9:18	f0:4d:a2:2a:61:26	ARP	222.31.66.220 is at 00:16:17:ad:e9:18
3	0.001717	222.31.66.224	222.31.66.220	ICMP	Echo (ping) request
4	0.001875	222.31.66.220	222.31.66.224	ICMP	Echo (ping) reply



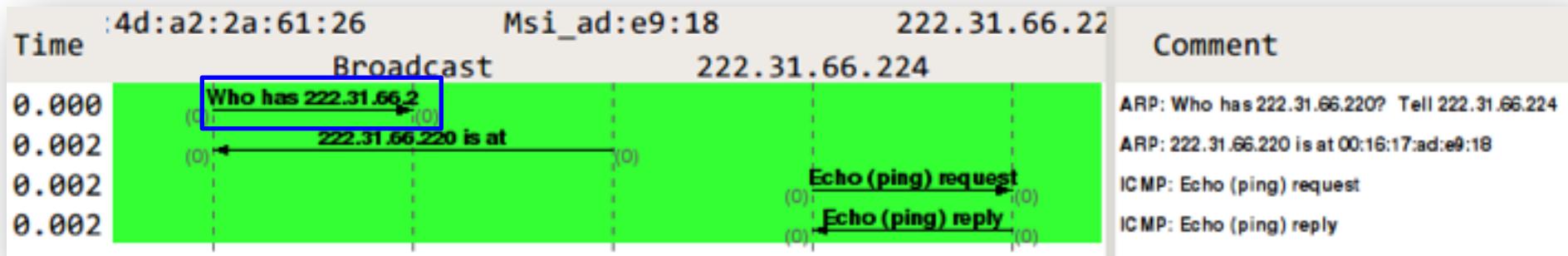


局域网主机可达状态检测——ARP扫描

- ARP广播请求

—局域网连通状态主机必定会响应正常ARP广播请求

No.	Time	Source	Destination	Protocol.	Info
1	0.000000	f0:4d:a2:2a:61:26	Broadcast	ARP	Who has 222.31.66.220? Tell 222.31.66.224
2	0.001709	Msi_ad:e9:18	f0:4d:a2:2a:61:26	ARP	222.31.66.220 is at 00:16:17:ad:e9:18
3	0.001717	222.31.66.224	222.31.66.220	ICMP	Echo (ping) request
4	0.001875	222.31.66.220	222.31.66.224	ICMP	Echo (ping) reply



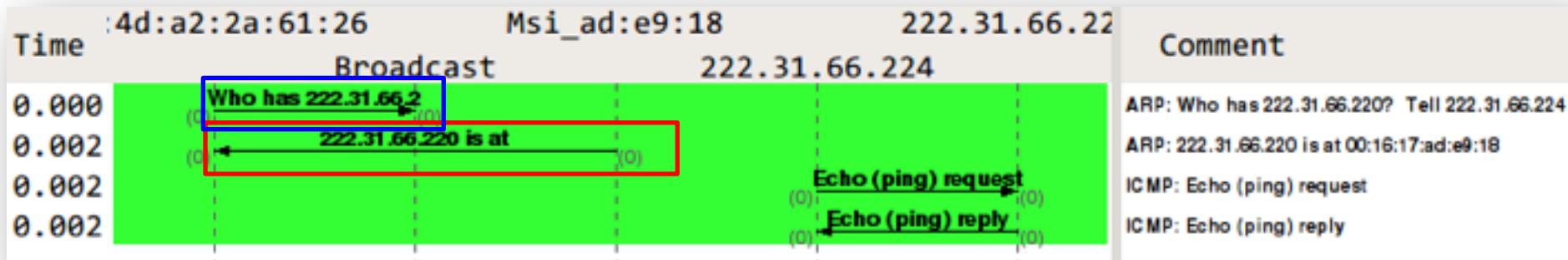


局域网主机可达状态检测——ARP扫描

- ARP广播请求

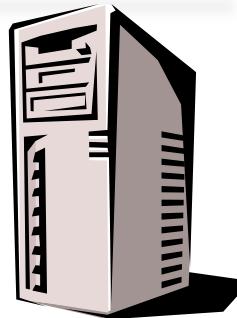
—局域网连通状态主机必定会响应正常ARP广播请求

No.	Time	Source	Destination	Protocol.	Info
1	0.000000	f0:4d:a2:2a:61:26	Broadcast	ARP	Who has 222.31.66.220? Tell 222.31.66.224
2	0.001709	Msi_ad:e9:18	f0:4d:a2:2a:61:26	ARP	222.31.66.220 is at 00:16:17:ad:e9:18
3	0.001717	222.31.66.224	222.31.66.220	ICMP	Echo (ping) request
4	0.001875	222.31.66.220	222.31.66.224	ICMP	Echo (ping) reply





ICMP echo扫描



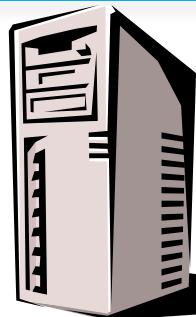
- 实现原理
 - ping的实现机制
- 优点
 - 简单，系统支持
- 缺点
 - 很容易被防火墙限制



ICMP echo扫描



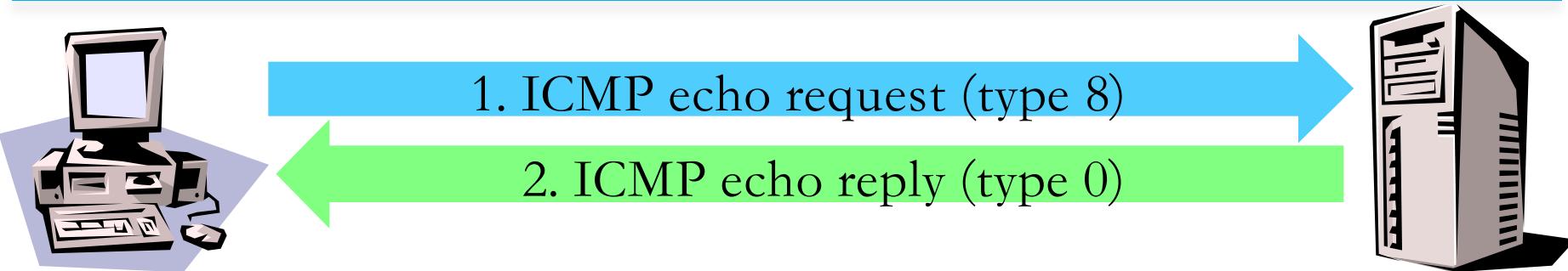
1. ICMP echo request (type 8)



- 实现原理
 - ping的实现机制
- 优点
 - 简单，系统支持
- 缺点
 - 很容易被防火墙限制



ICMP echo扫描

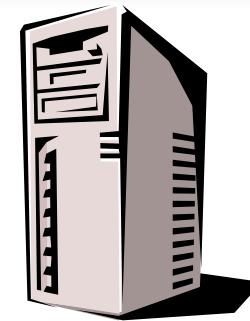


- 实现原理
 - ping的实现机制
- 优点
 - 简单，系统支持
- 缺点
 - 很容易被防火墙限制

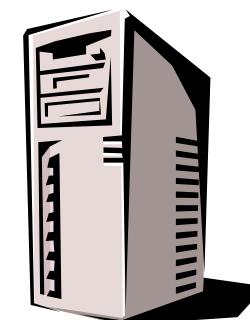
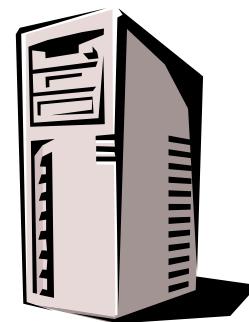


ICMP Sweep扫描

- 并行多路ICMP echo扫描



...

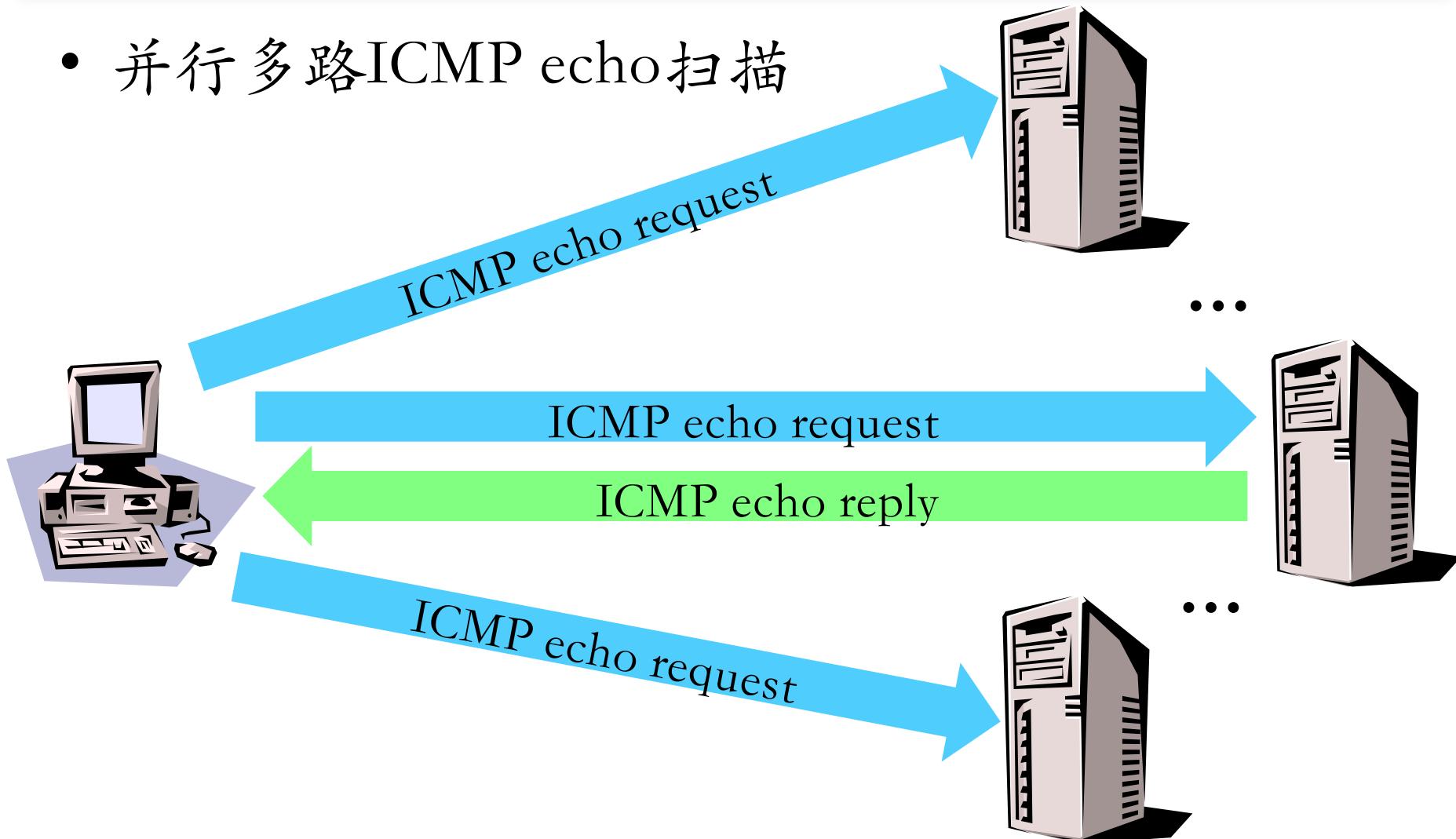


...



ICMP Sweep扫描

- 并行多路ICMP echo扫描





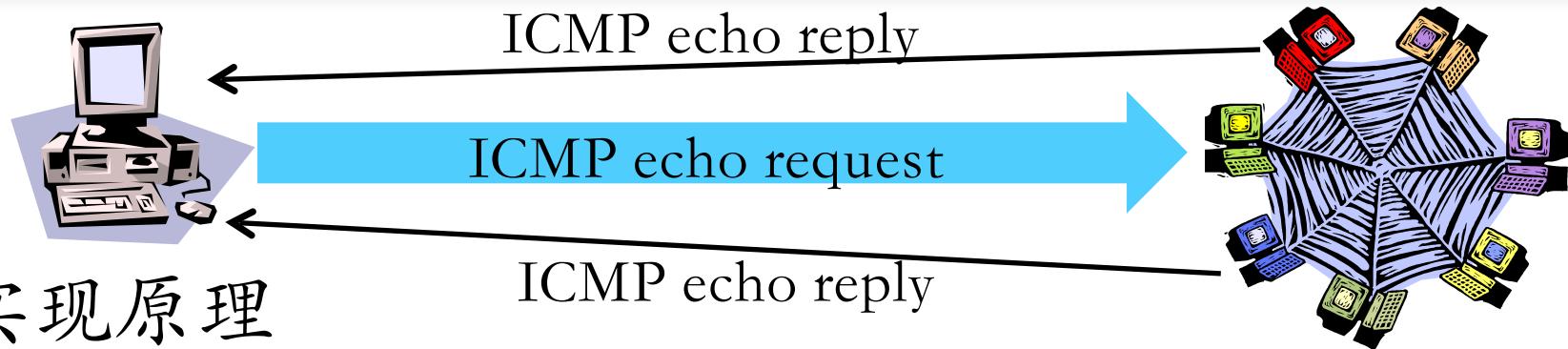
ICMP Broadcast扫描



- 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



ICMP Broadcast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



- 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)



- 防火墙和网络过滤设备的存在
 - 常常导致传统的探测手段变得无效
 - 为了突破这种限制
 - ① 异常的IP包头
 - ② 在IP头中设置无效的字段值
 - ③ 错误的数据分片
 - ④ 通过超长包探测内部路由器
 - ⑤ 反向映射探测



① 异常的IP包头

一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 Field 和IP Options Field

② 在IP头中设置无效的字段值

一向目标主机发送的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



③ 错误的数据分片

—当目标主机接收到错误的数据分片，并且在规定的时间间隔内得不到更正时，将丢弃这些错误数据包，并向发送主机反馈ICMP Fragment Reassembly Time Exceeded 错误报文

④ 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，从而获取目标系统的网络拓扑结构



⑤ 反向映射探测

—用处

- 该技术用于探测被过滤设备和防火墙保护的网络和主机

—方法

- 构造可能的内部IP地址列表，并向这些地址发送数据包
- 对方路由器进行IP识别并路由
- 根据是否返回错误报文来进行探测

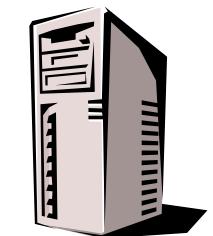
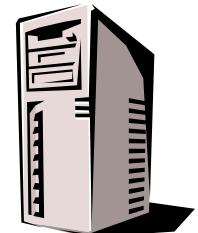


端口扫描技术——主机状态详细信息探测技术

- 开放扫描
 - 会产生大量审计数据，容易被对方发现，但其可靠性高
- 隐蔽扫描
 - 能有效避免对方入侵检测系统和防火墙的检测，但这种扫描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
- 半开放扫描
 - 隐蔽性和可靠性介于前两者之间

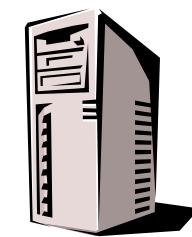
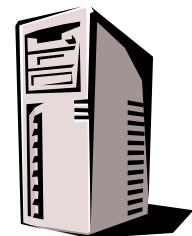
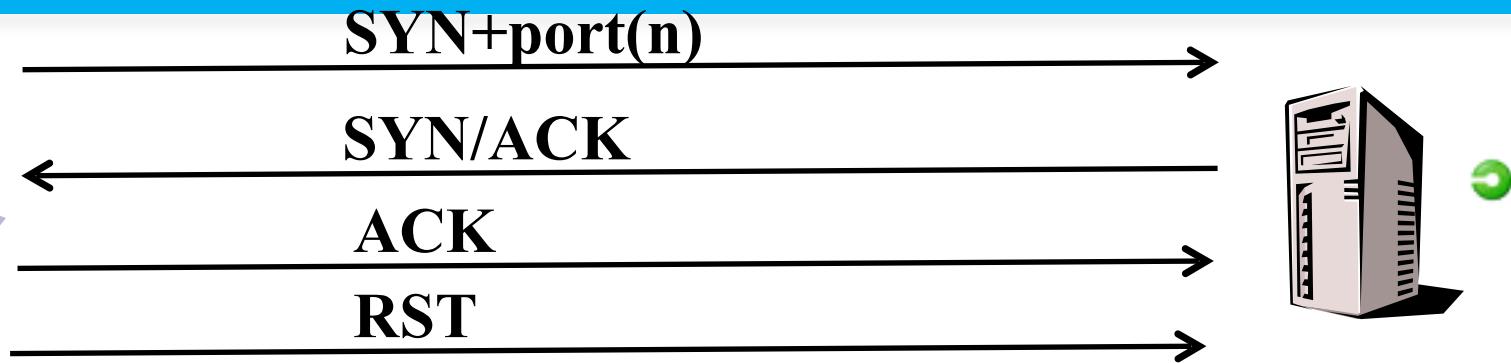


开放扫描——TCP Connect扫描



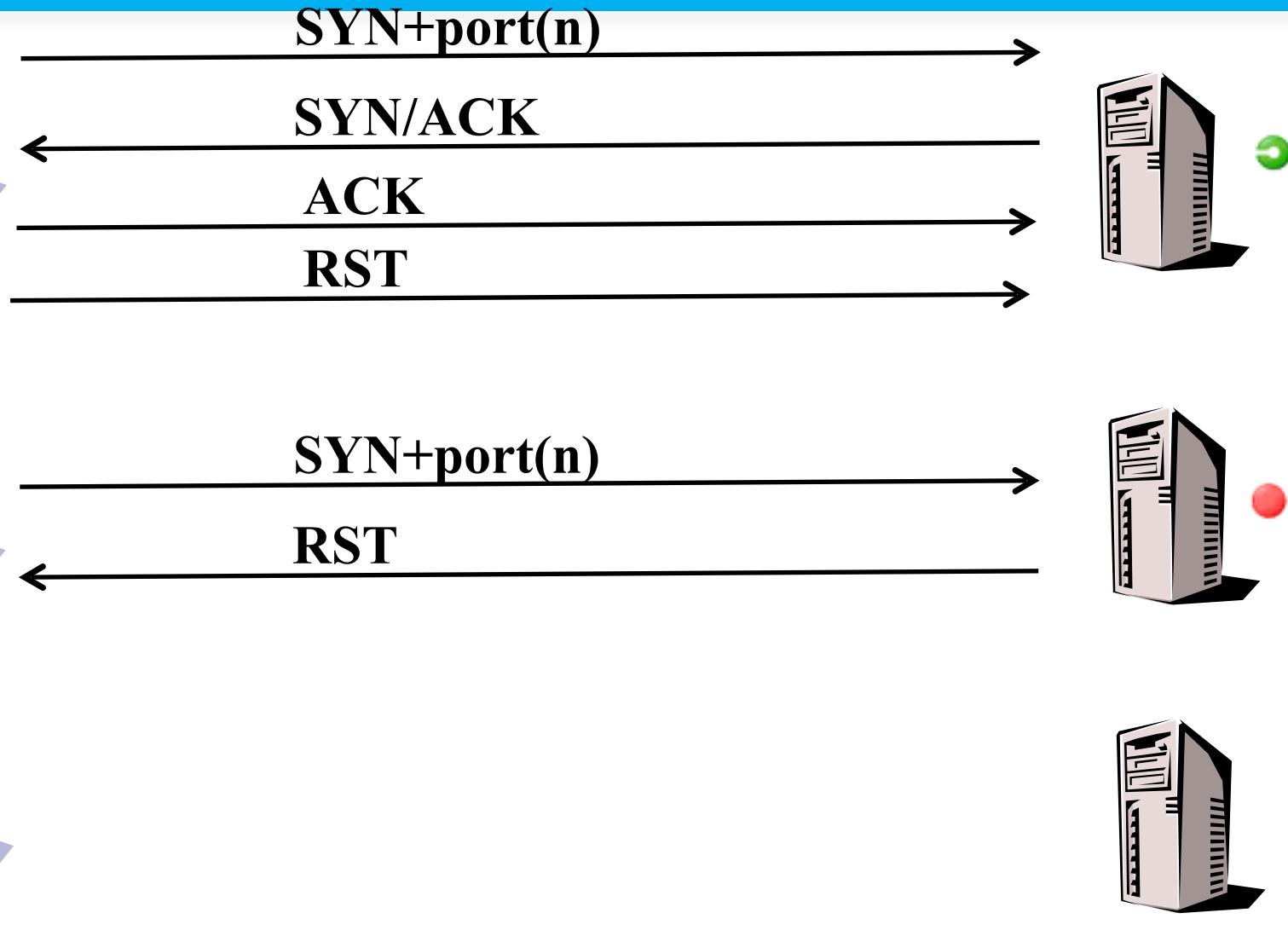


开放扫描——TCP Connect 扫描



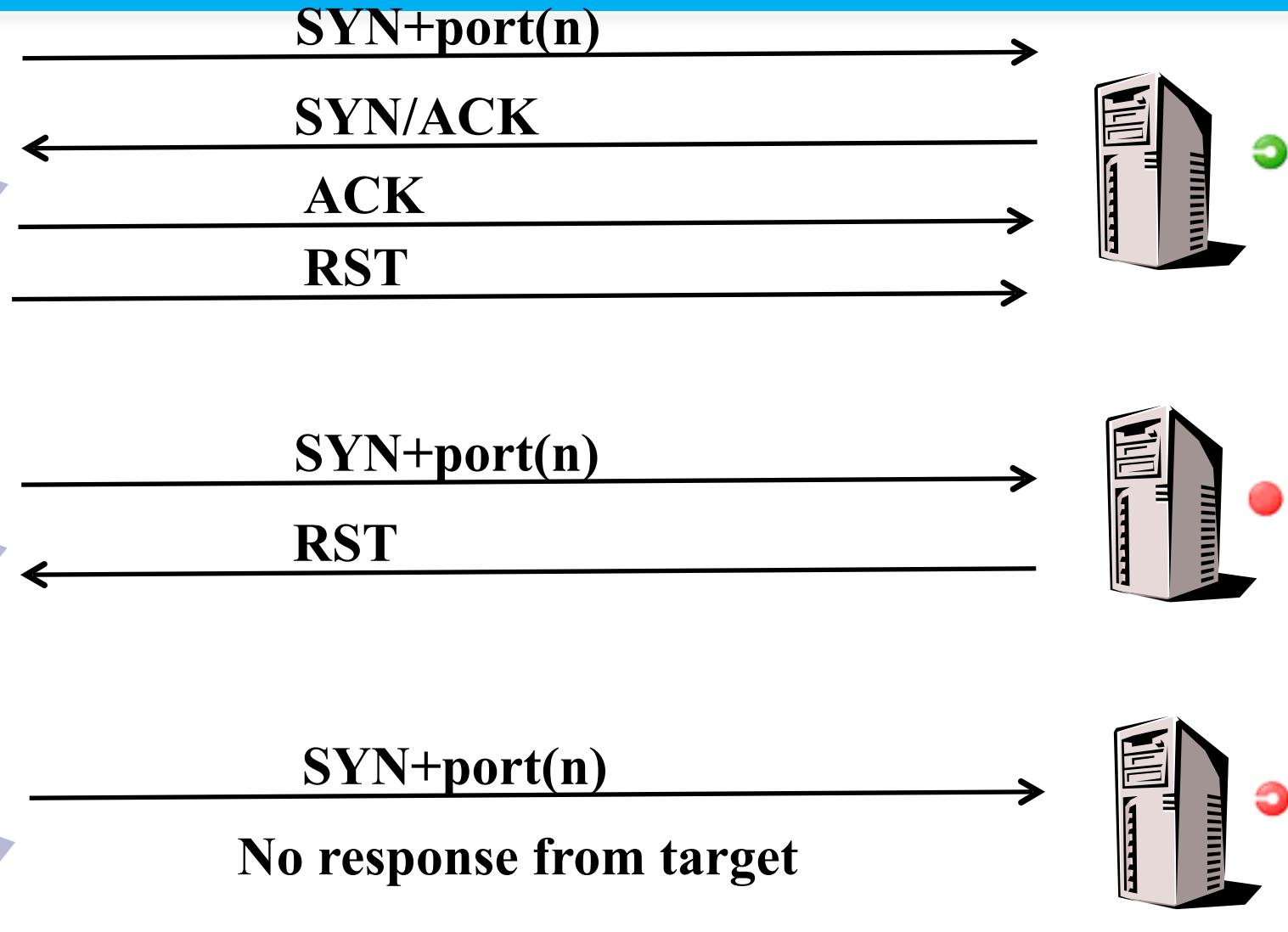


开放扫描——TCP Connect 扫描





开放扫描——TCP Connect 扫描





开放扫描

• TCP Connect扫描

— 实现原理

- `connect()`
- 完成TCP三次握手

— 优点

- 稳定可靠，不需要特殊的权限

— 缺点

- 扫描方式不隐蔽，服务器日志会记录下大量密集的连接和错误记录，并容易被防火墙发现和屏蔽

• TCP 反向ident扫描

— 实现原理

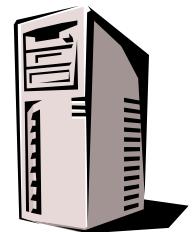
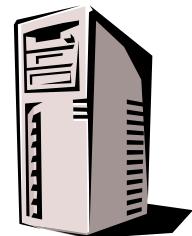
- 利用TCP的认证协议**ident**的漏洞

— 缺点

- 这种方法只能在和目标端口建立了一个完整的TCP连接后才能看到

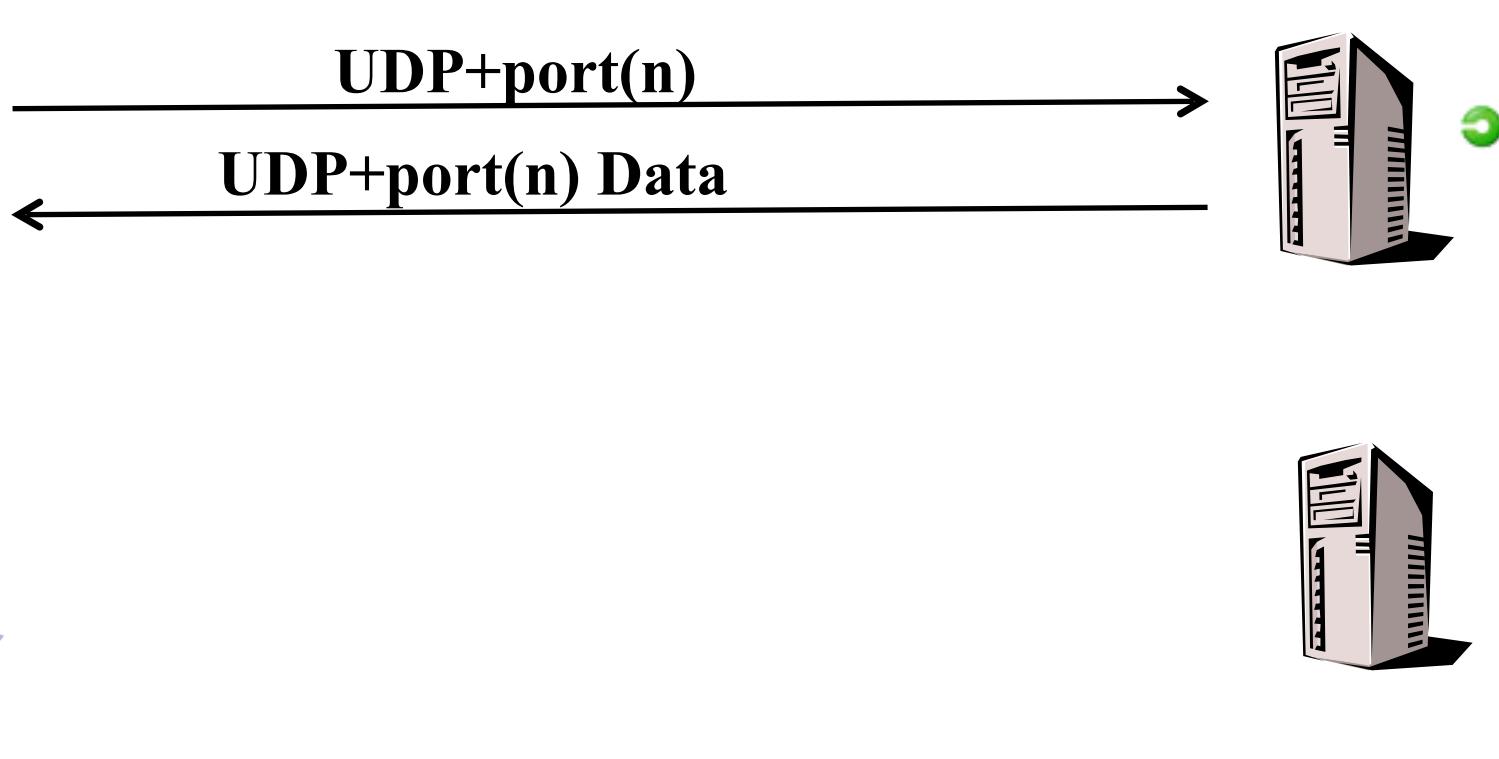


开放扫描——UDP扫描



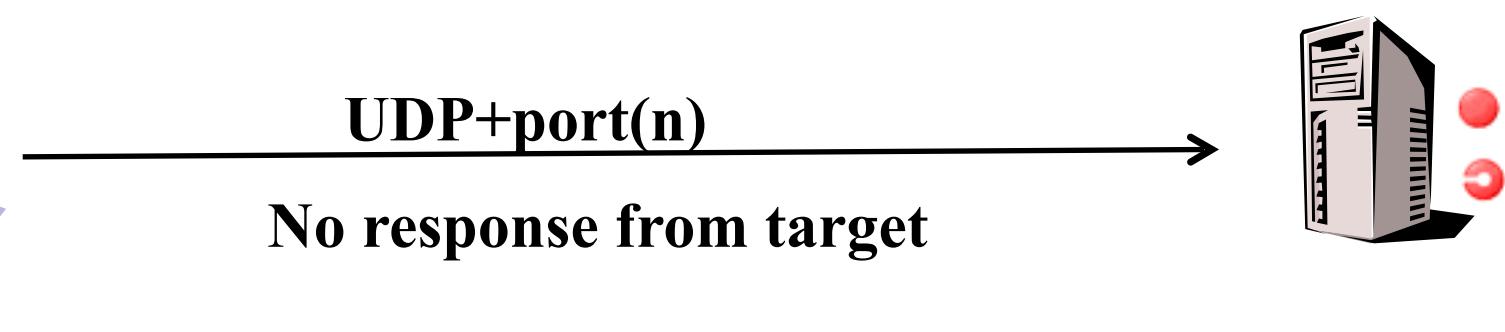
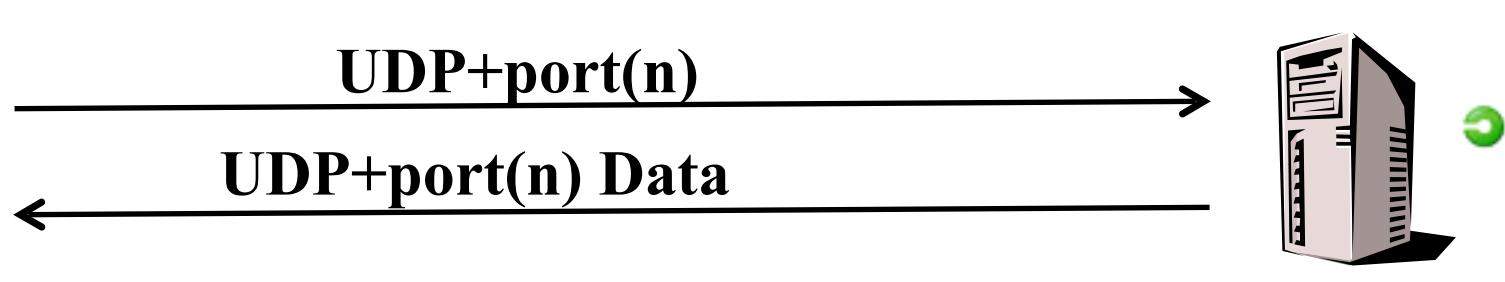


开放扫描——UDP扫描



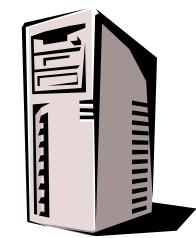
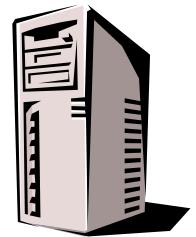
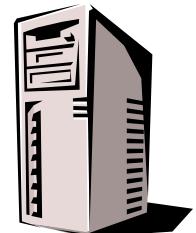


开放扫描——UDP扫描



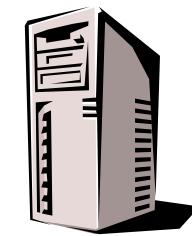
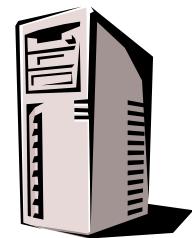
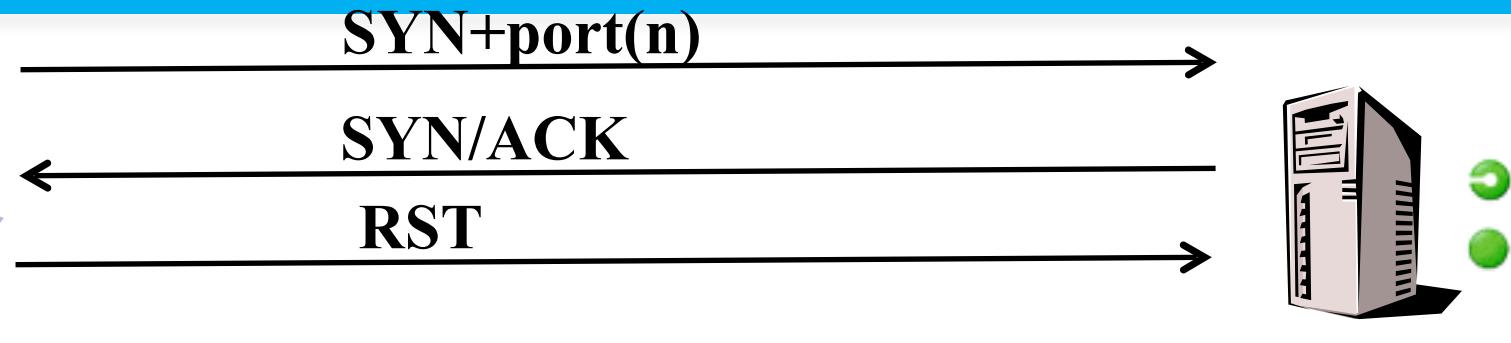


半开放扫描——TCP SYN扫描



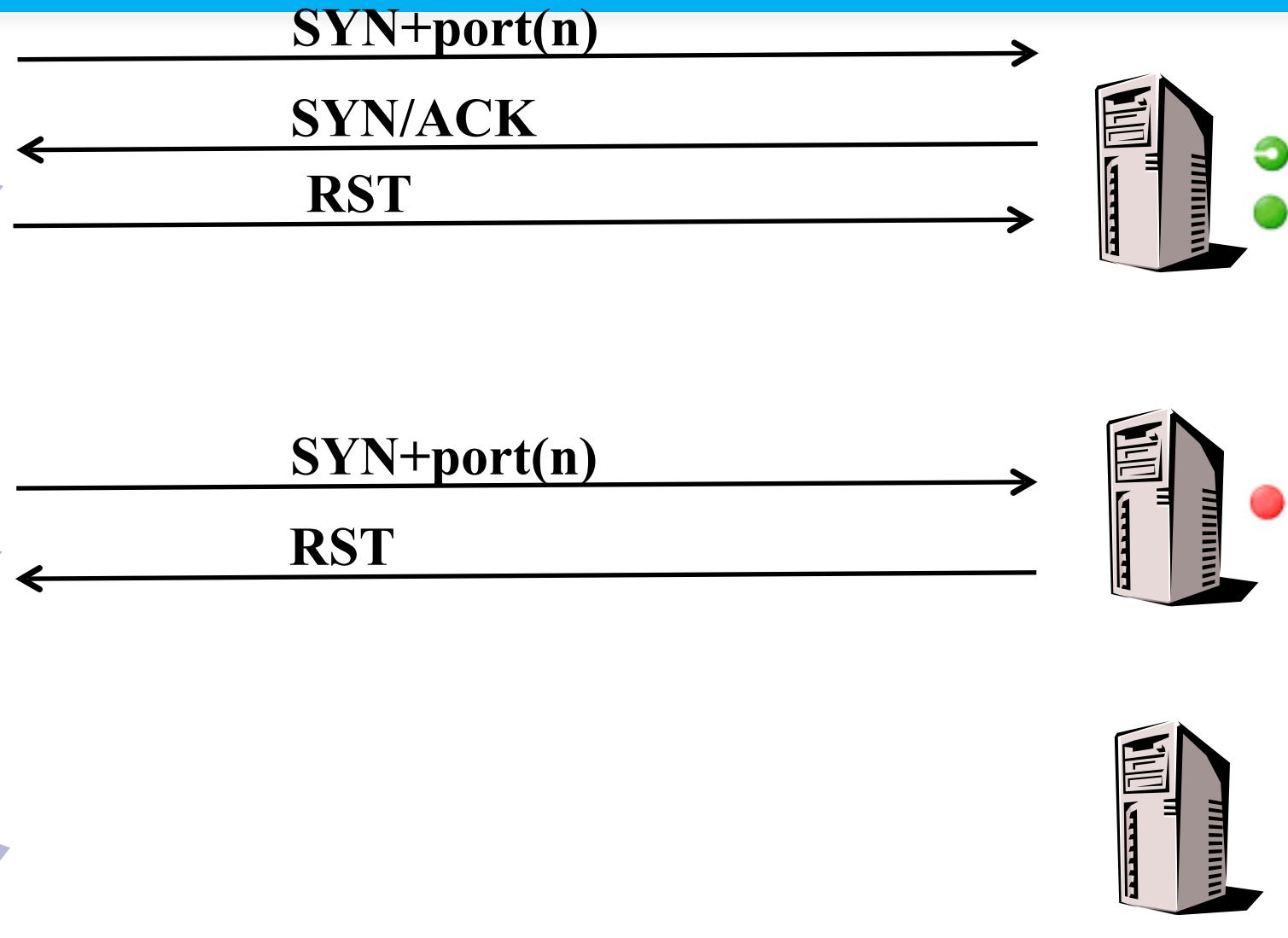


半开放扫描——TCP SYN扫描



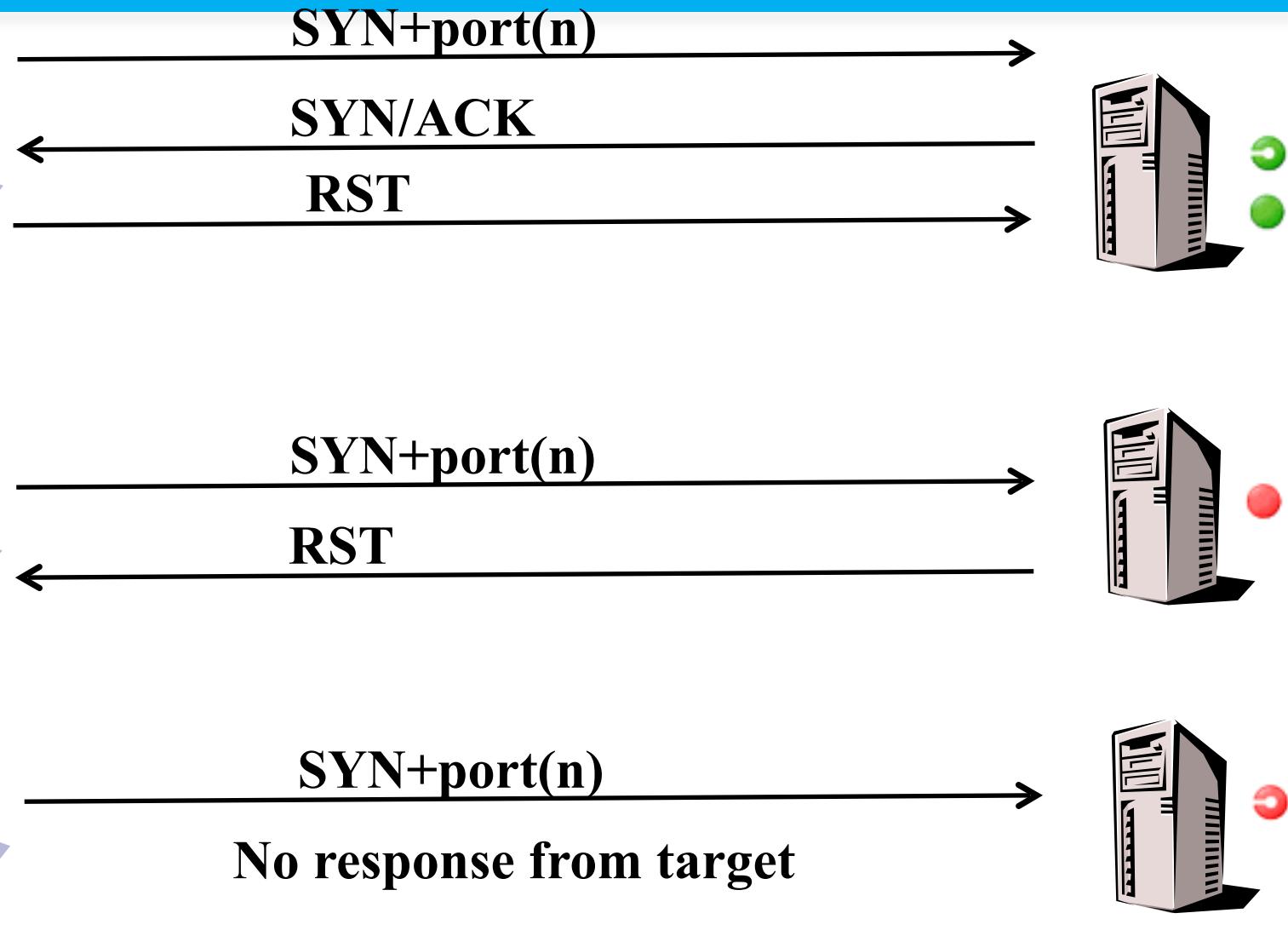


半开放扫描——TCP SYN扫描



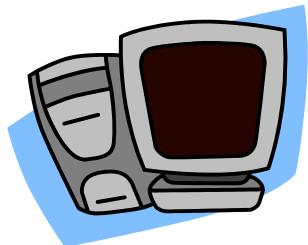
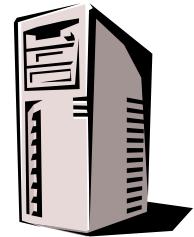


半开放扫描——TCP SYN扫描





半开放扫描——TCP间接扫描

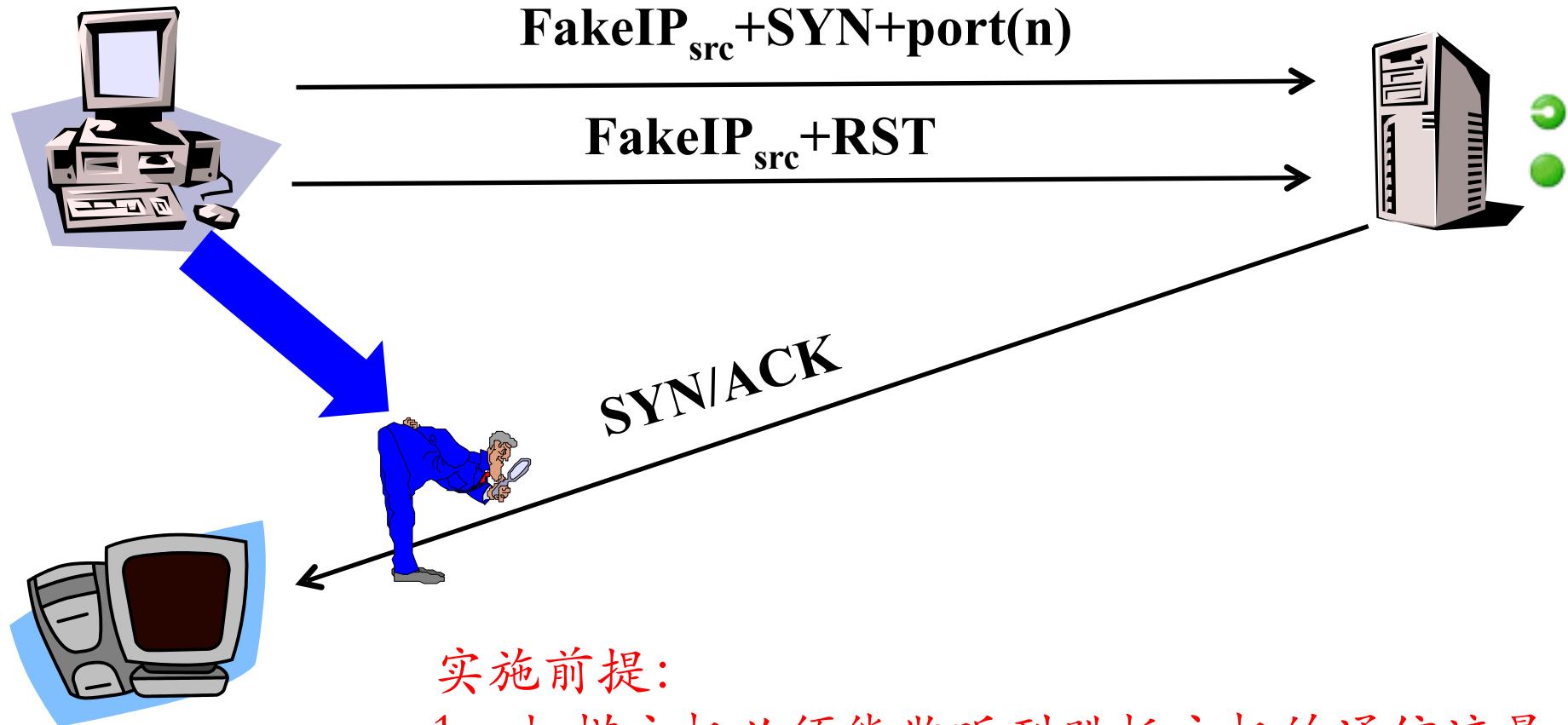


实施前提：

1. 扫描主机必须能监听到跳板主机的通信流量
2. 广域网上的路由器必须允许伪造源IP地址



半开放扫描——TCP间接扫描





半开放扫描

- TCP SYN扫描

- 实现原理

- 仅发送SYN包
 - 不建立完整TCP连接
 - 又称为半开放/半连接扫描

- 优点

- 隐蔽性较全连接扫描好，很多系统对这种半扫描很少记录

- 缺点

- 需要超级用户权限构造SYN报文
 - 网络防护设备会有记录

- TCP间接扫描

- 实现原理

- 伪造第三方源IP发起SYN扫描

- 优点

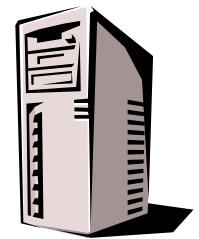
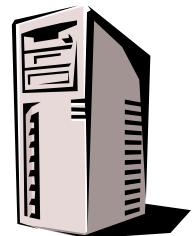
- 隐蔽性好

- 缺点

- 对跳板主机的要求较高
 - 广域网中受制于路由器的包过滤规则

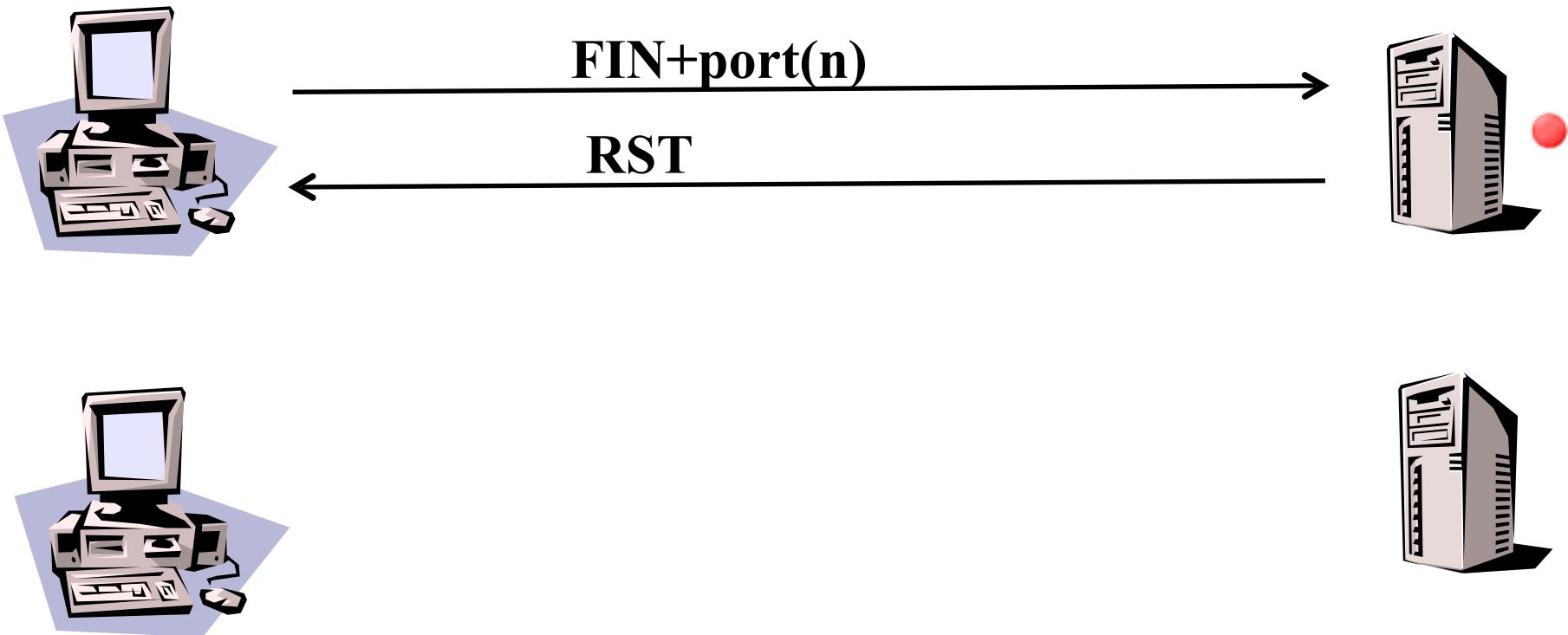


隐蔽扫描——TCP FIN扫描(1/2)



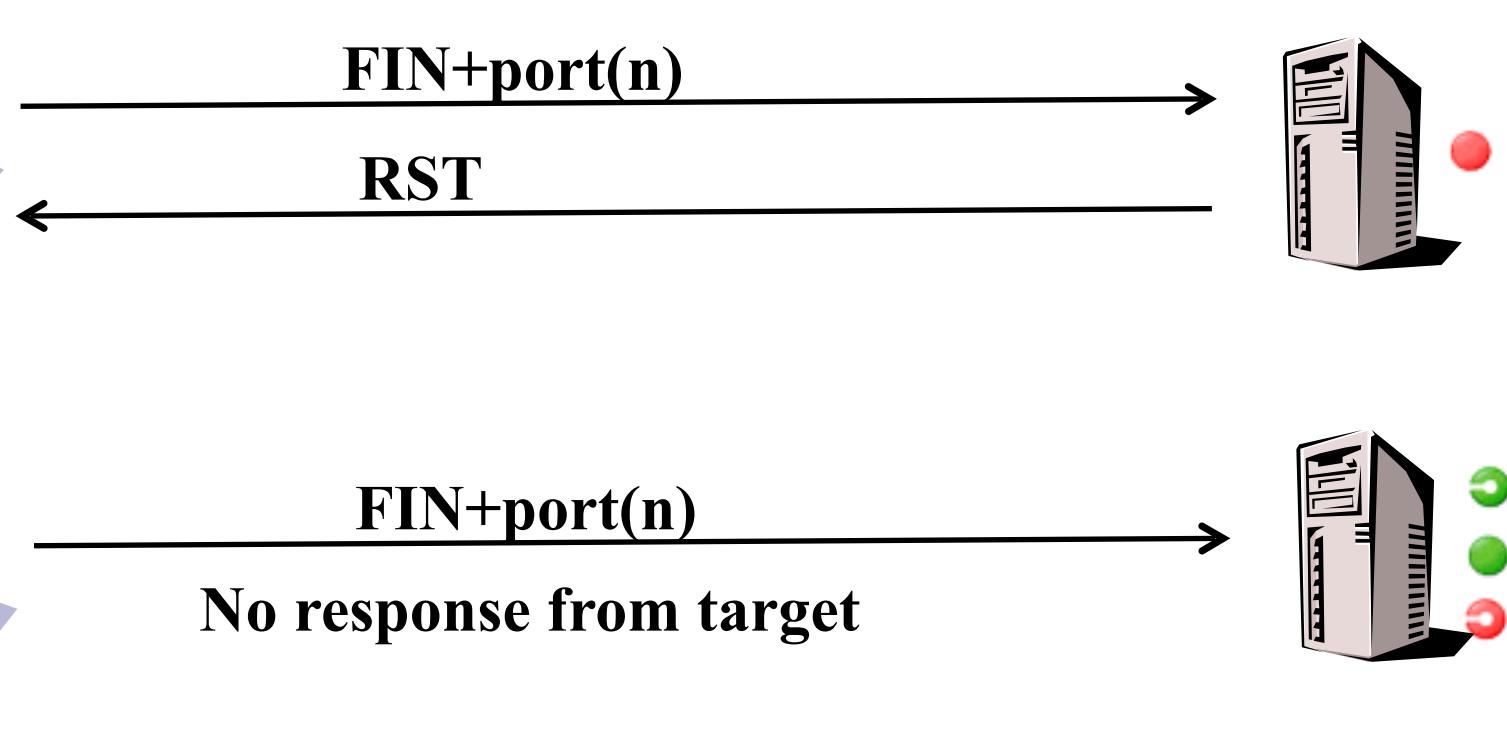


隐蔽扫描——TCP FIN扫描(1/2)





隐蔽扫描——TCP FIN扫描(1/2)



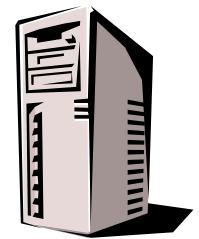
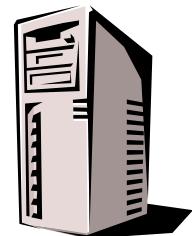


隐蔽扫描——TCP FIN扫描(2/2)

- 实现原理
 - 仅发送FIN包
- 优点
 - FIN数据包能够通过只监测SYN包的包过滤器
 - 隐蔽性较SYN扫描更高
- 缺点
 - 跟SYN扫描类似，需要自己构造数据包，要求由超级用户或者授权用户访问专门的系统调用

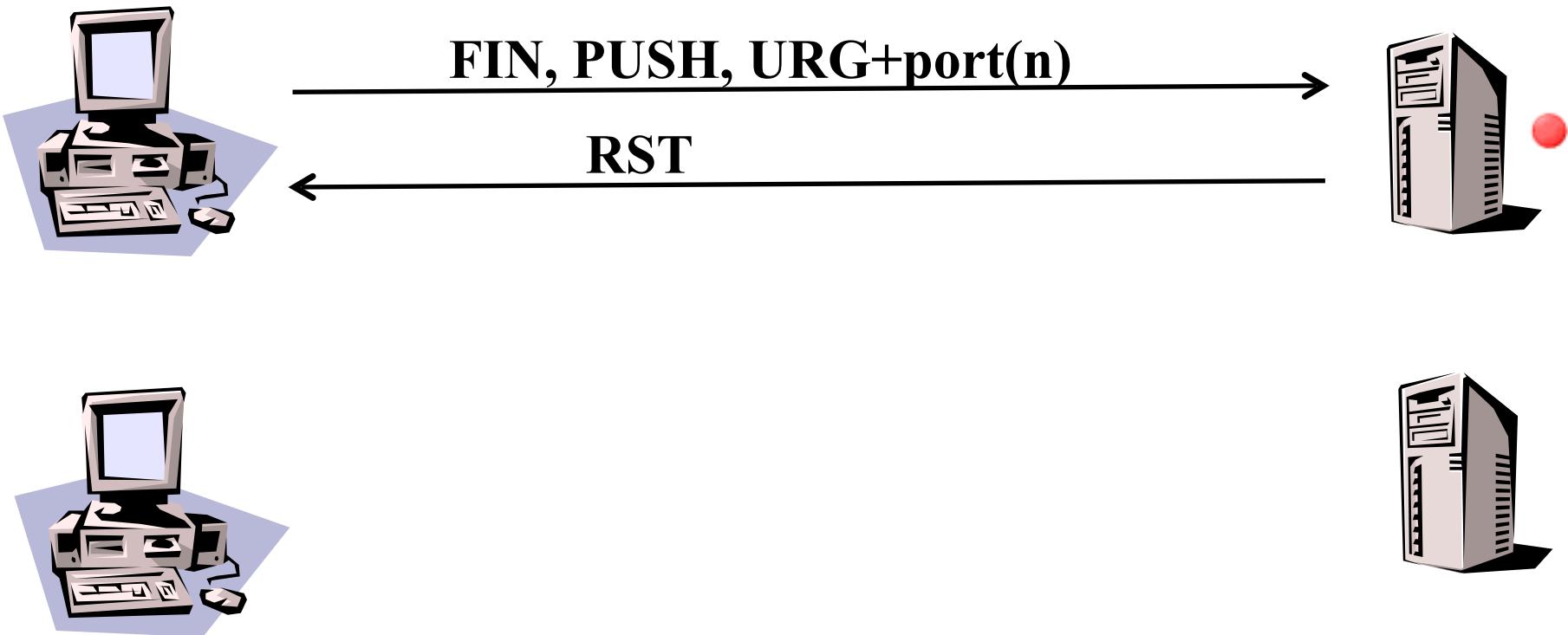


隐蔽扫描——TCP Xmas扫描



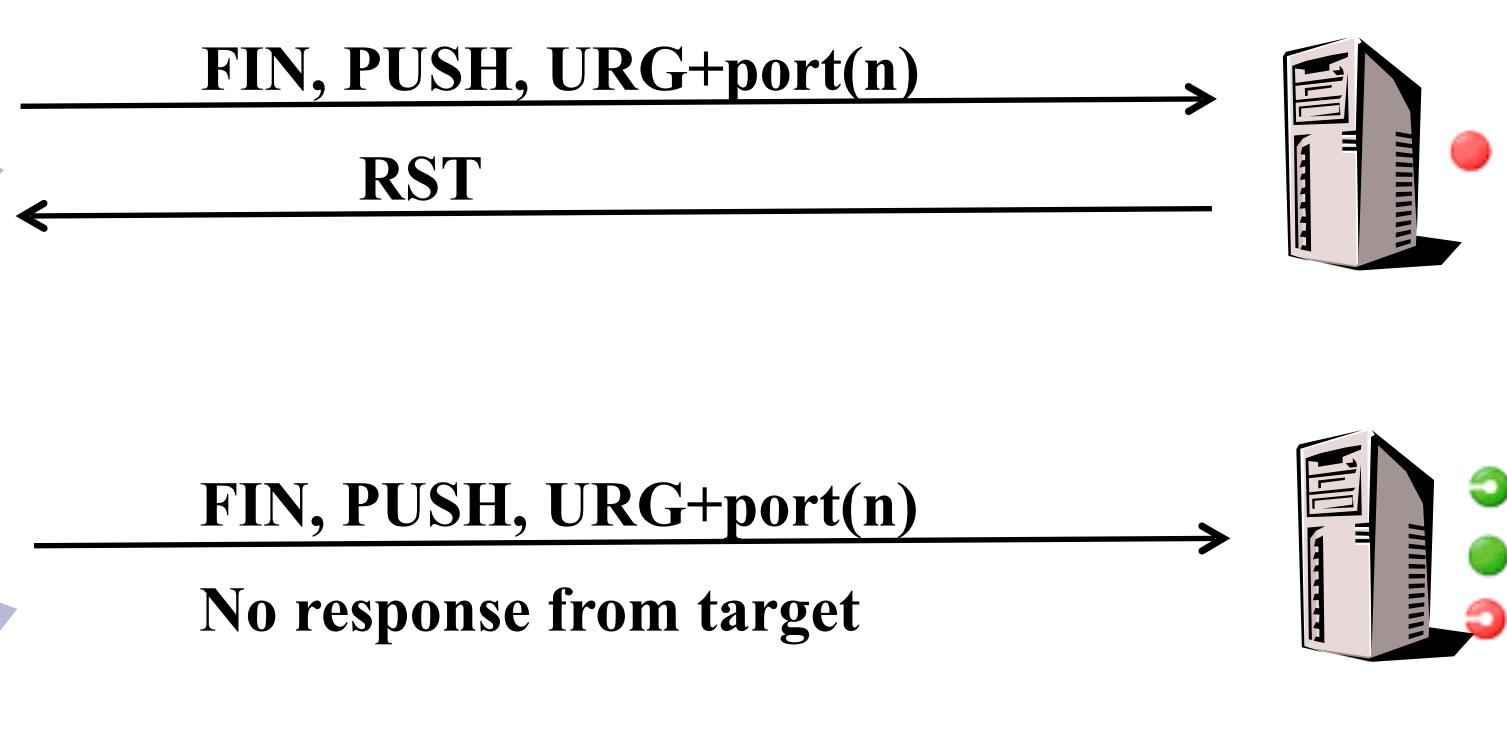


隐蔽扫描——TCP Xmas扫描



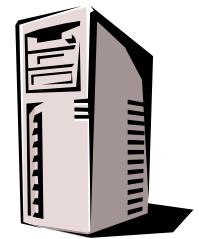
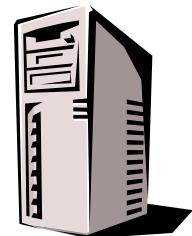


隐蔽扫描——TCP Xmas 扫描



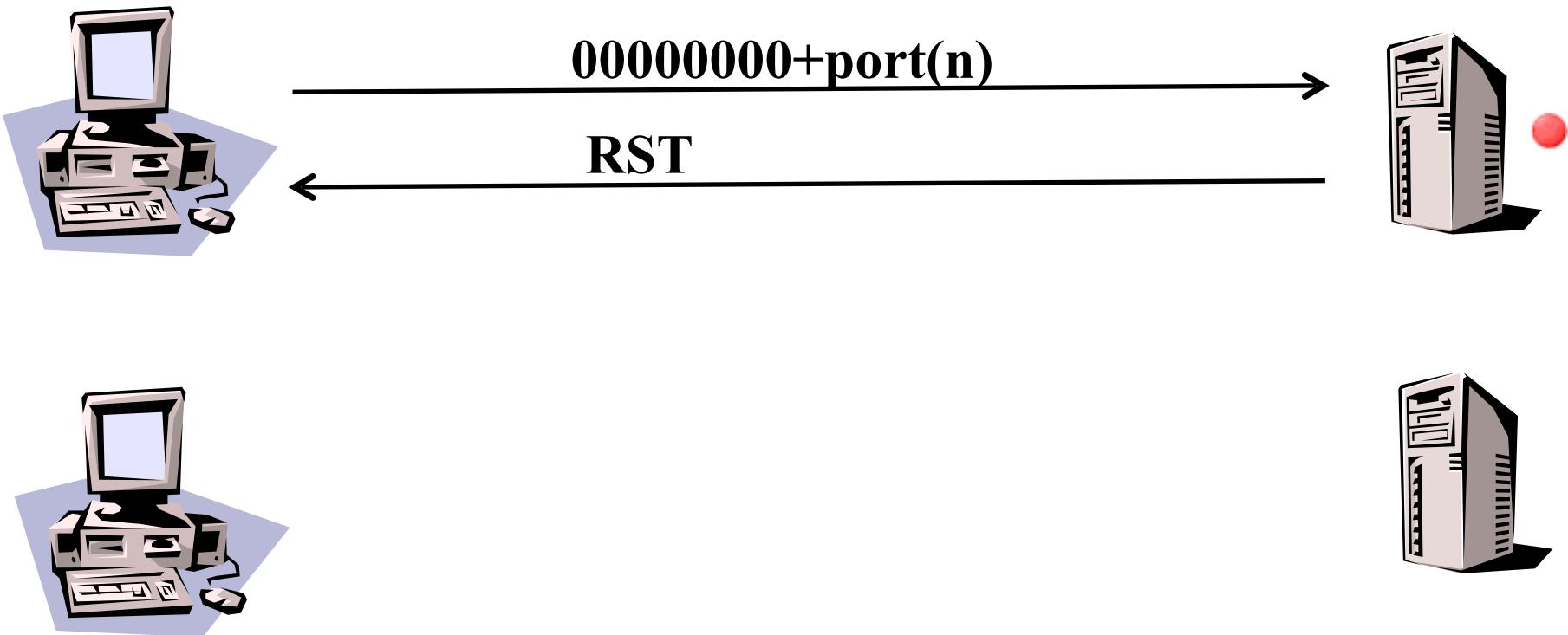


隐蔽扫描——TCP Null扫描



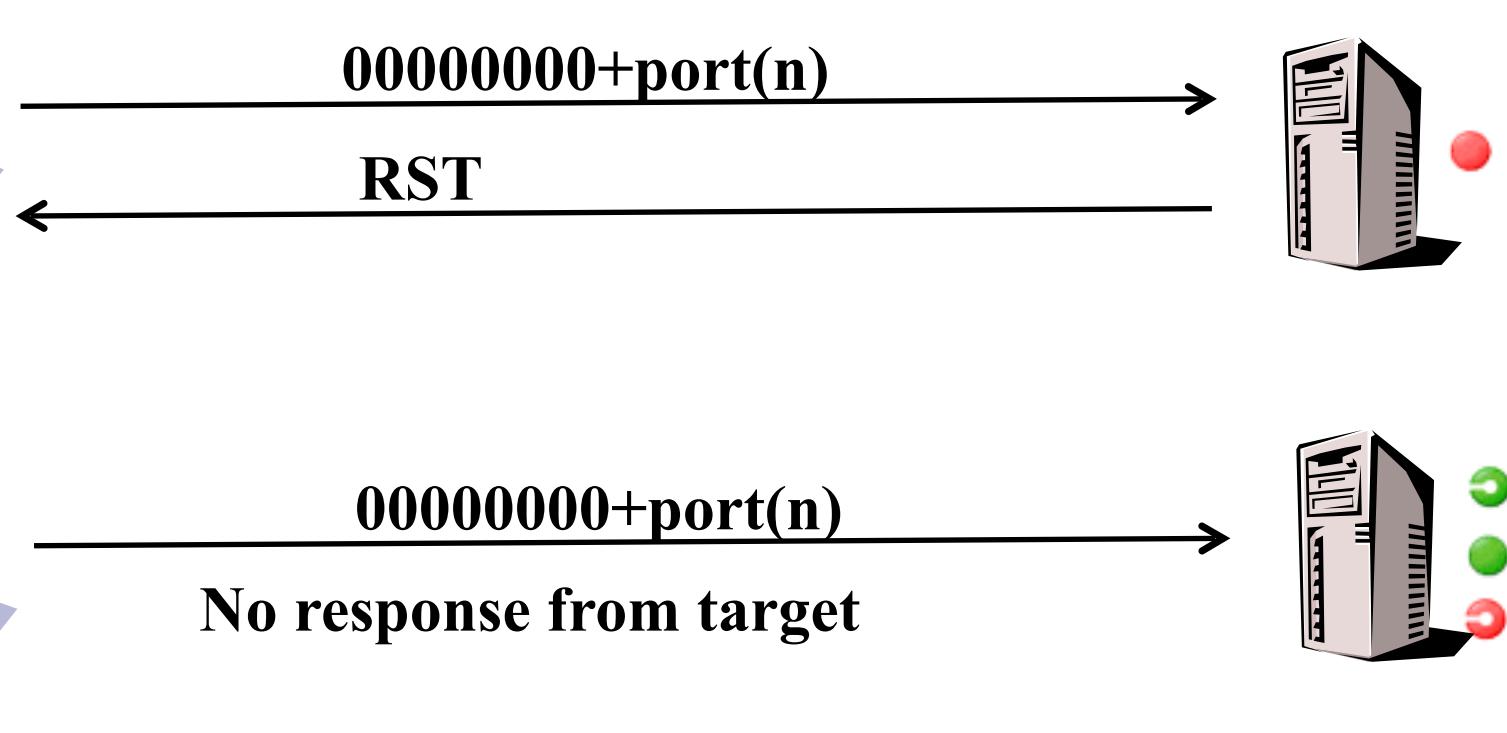


隐蔽扫描——TCP Null扫描





隐蔽扫描——TCP Null扫描





隐蔽扫描——TCP Xmas和TCP Null扫描

- 实现原理
 - Xmas: 设置TCP报文头FIN、URG和PUSH标记
 - Null: 关闭所有TCP报文头标记
- 优点
 - 隐蔽性好
- 缺点
 - 需要自己构造数据包，要求有超级用户或者授权用户权限

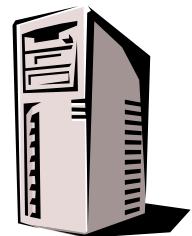
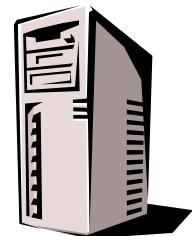


隐蔽扫描——分段扫描

- 实现原理
 - 将一个完整TCP报文分割封装到2个或多个IP报文分别独立发送
- 优点
 - 隐蔽性好，可穿越防火墙
- 缺点
 - 可能被丢弃
 - 某些程序在处理这些小数据包时会出现异常

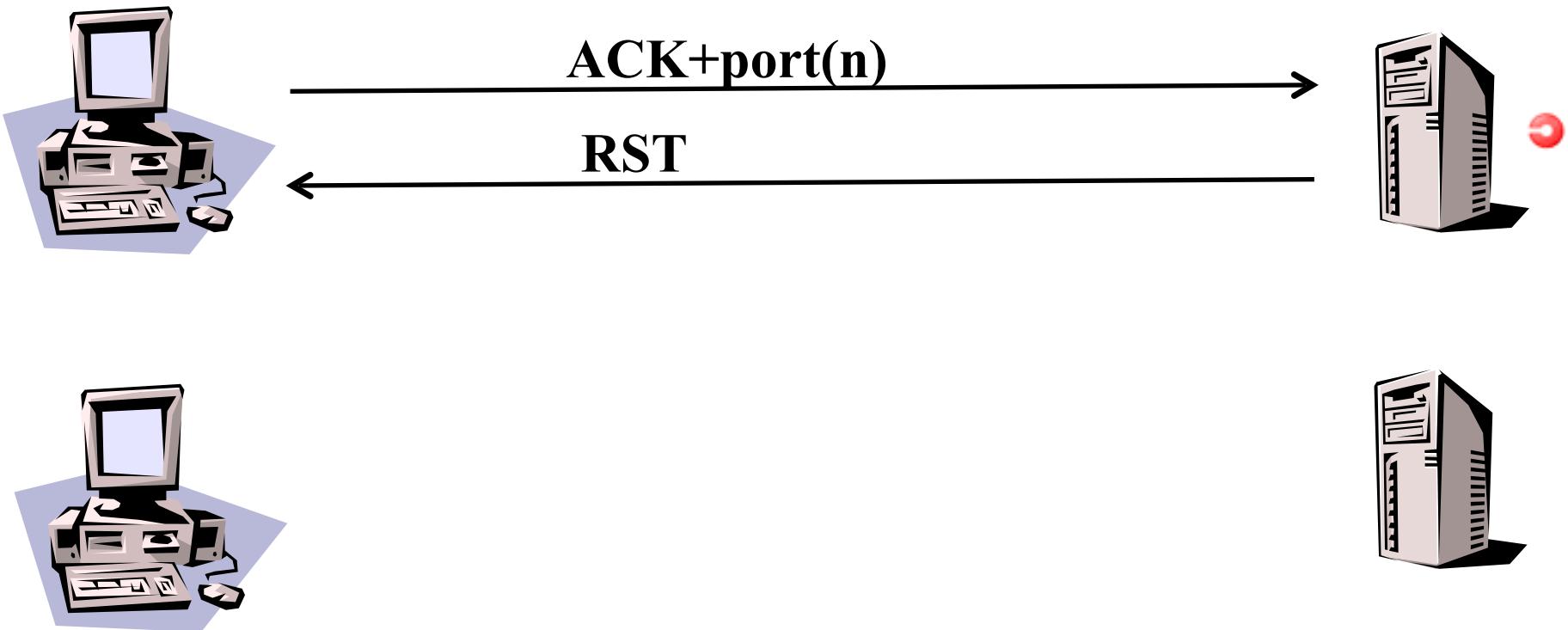


隐蔽扫描——ACK扫描(1/2)



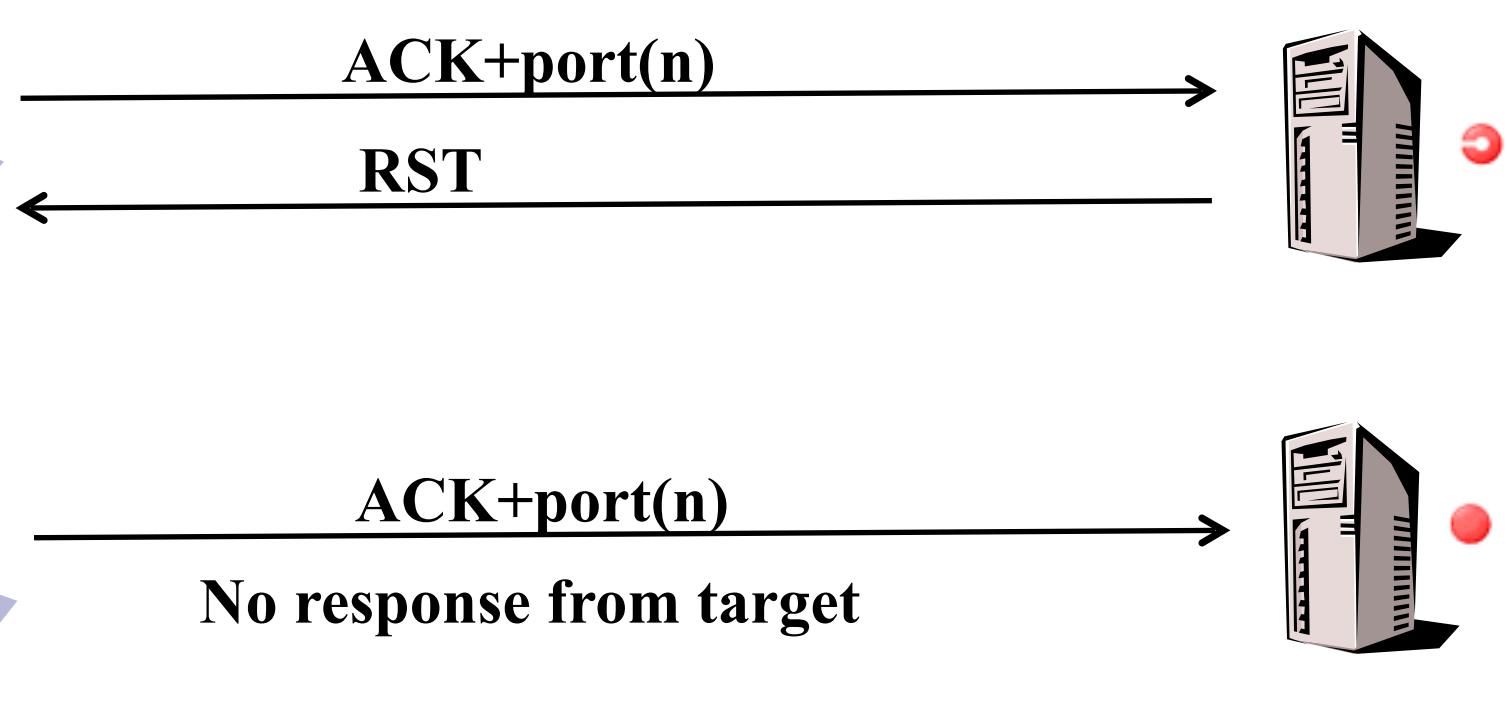


隐蔽扫描——ACK扫描(1/2)





隐蔽扫描——ACK扫描(1/2)





隐蔽扫描——ACK扫描(2/2)

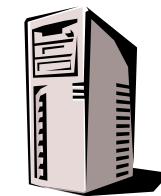
- 实现原理
 - 构造并发送ACK报文
 - 包过滤防火墙会检查TCP会话状态列表，若发现无匹配会话则有可能返回RST报文
 - 正常主机的关闭端口则不会响应该报文
- 优点
 - 探测目标主机的包过滤规则
- 缺点
 - 可能被丢弃
 - 不能用于判断端口是否开放



隐蔽扫描——IDLE扫描(1/2)

实施前提：

1. 跳板主机处于网络空闲状态
2. 跳板主机的IP序列号产生规则是连续递增
3. 广域网上的路由器必须允许伪造源IP地址

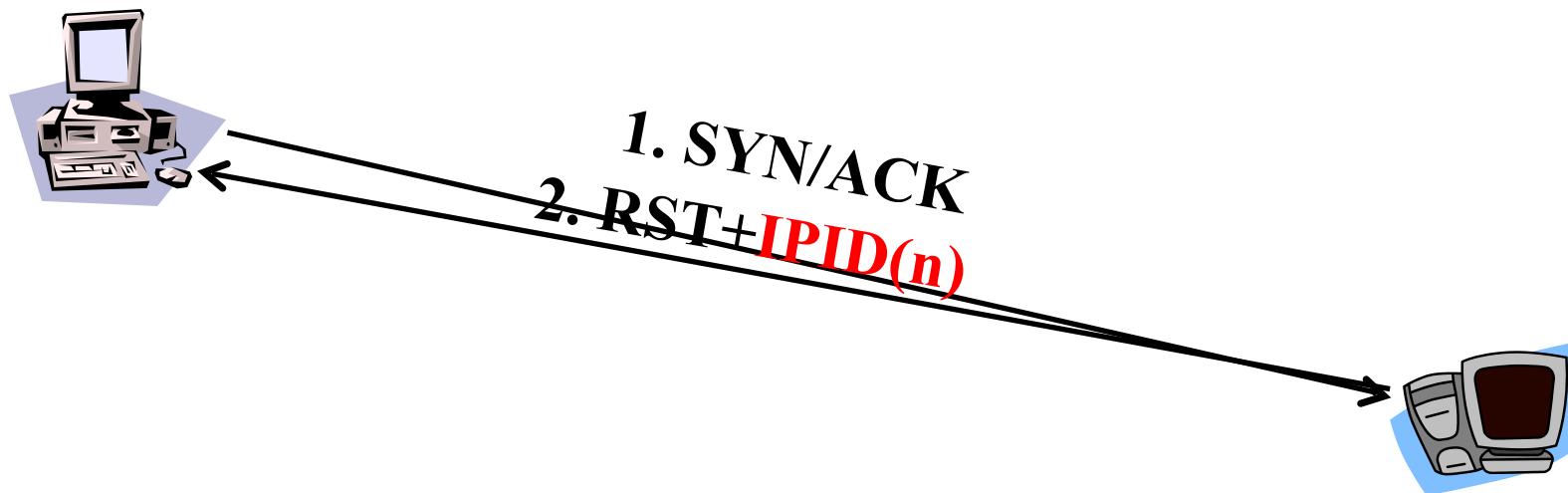
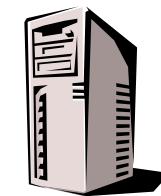




隐蔽扫描——IDLE扫描(1/2)

实施前提：

1. 跳板主机处于网络空闲状态
2. 跳板主机的IP序列号产生规则是连续递增
3. 广域网上的路由器必须允许伪造源IP地址

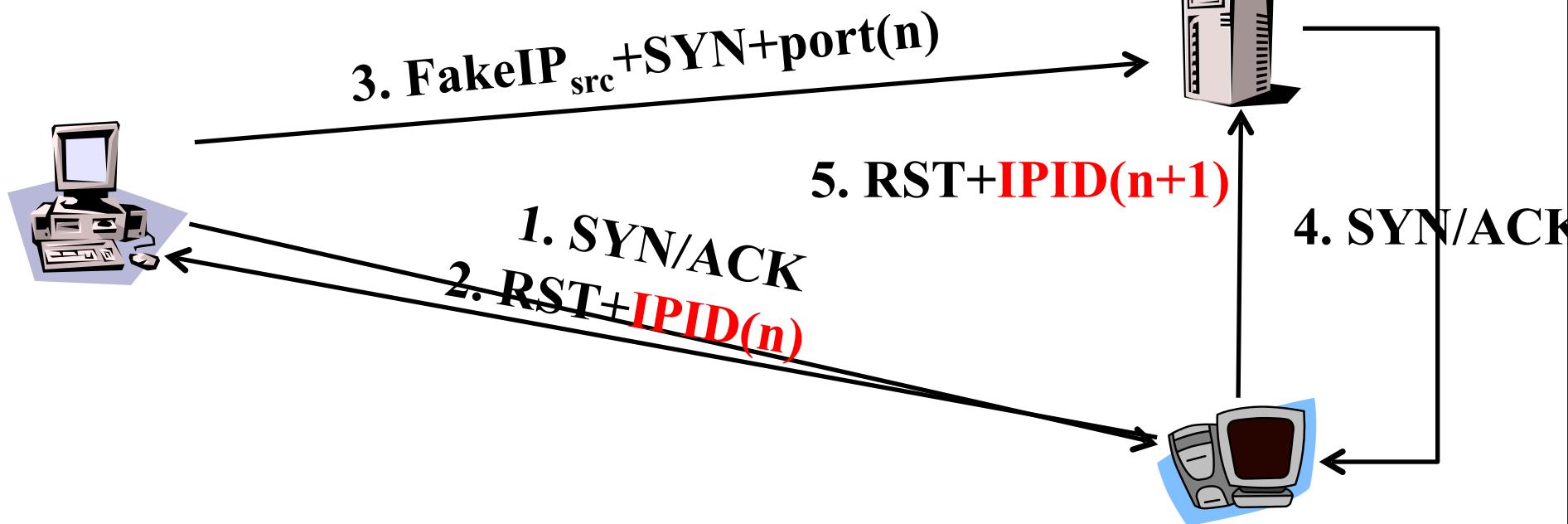




隐蔽扫描——IDLE扫描(1/2)

实施前提：

1. 跳板主机处于网络空闲状态
2. 跳板主机的IP序列号产生规则是连续递增
3. 广域网上的路由器必须允许伪造源IP地址

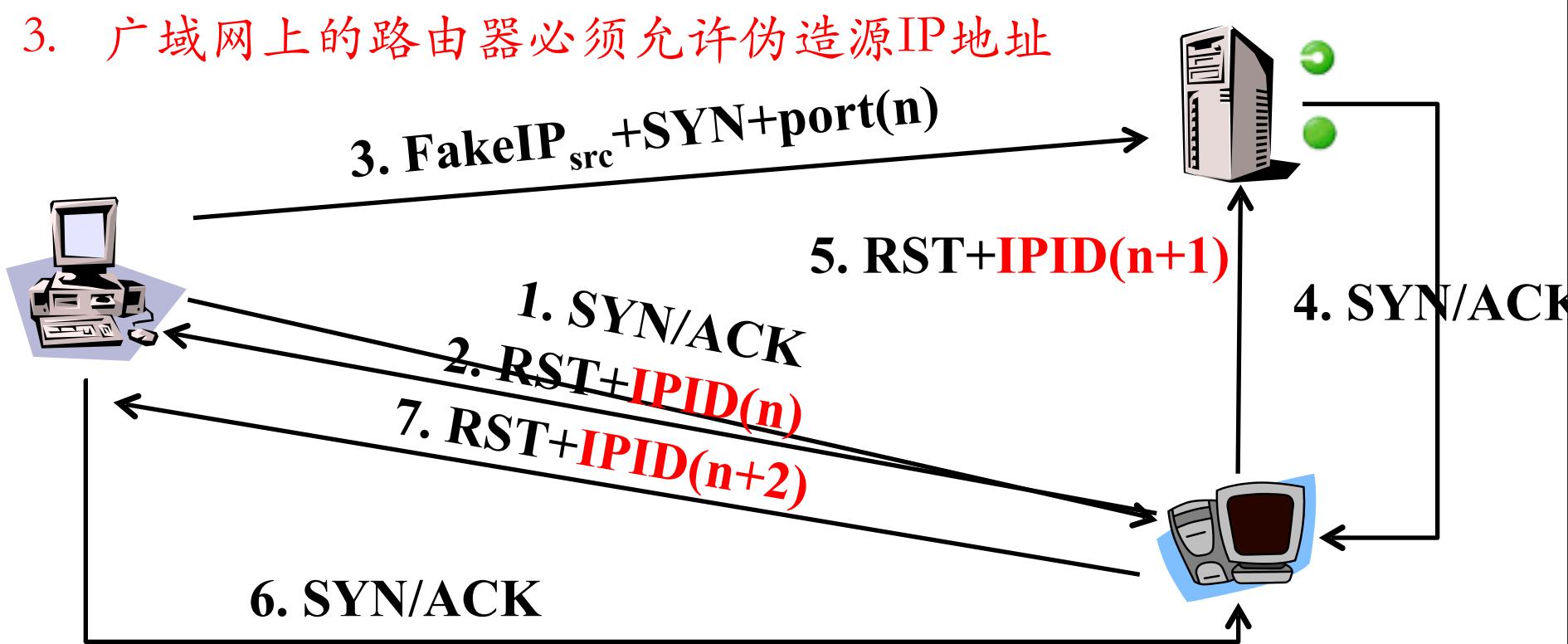




隐蔽扫描——IDLE扫描(1/2)

实施前提：

1. 跳板主机处于网络空闲状态
2. 跳板主机的IP序列号产生规则是连续递增
3. 广域网上的路由器必须允许伪造源IP地址





隐蔽扫描——IDLE扫描(2/2)

- 实现原理
—如前图所示
- 优点
 - 相比较于TCP间接扫描，无需监听跳板主机的通信流量
 - 目标主机很难发现真正的扫描源，扫描隐蔽性高
- 缺点
 - 对跳板主机的要求较多



- 实现原理
 - 根据各个OS在TCP/IP协议栈实现上的不同特点
 - 采用黑盒测试方法
 - 研究其对各种探测的响应形成识别指纹进行识别
- 根据采集指纹信息的方式，分为
 - 被动扫描方式
 - 主动扫描方式



- 被动扫描

- 实现原理

- 通过网络嗅探工具收集数据包，再对数据包的不同特征（TCP Window-size、IP TTL、IP TOS、DF位等参数）进行分析，来识别操作系统

- 优点

- 隐蔽性好

- 缺点

- 速度慢
 - 可靠性不高

- 主动扫描

- 实现原理

- 采用向目标系统发送构造的特殊包并监控其应答的方式来识别操作系统的类型

- 优点

- 速度快，可靠性高

- 缺点

- 严重依赖目标系统网络拓扑结构和过滤规则



栈指纹OS识别技术(3/5)

- Windows 主机的ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▼ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSSCP 0x00: Default; ECN: 0x00)
  Total Length: 60
  Identification: 0x0000 (0)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 127
  Protocol: ICMP (0x01)
  Header checksum: 0xb9c7 [correct]
  Source: 222.31.66.224 (222.31.66.224)
  Destination: 222.31.66.218 (222.31.66.218)
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x4d5a [correct]
  Identifier: 0x0001
  Sequence number: 1 (0x0001)
▼ Data (32 bytes)
  Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
  [Length: 32]
0000  5c f3 fc 49 43 e8 f0 4d  a2 2a 61 26 08 00 45 00  \..IC..M .*a&..E.
0010  00 3c 00 00 40 00 7f 01  b9 c7 de 1f 42 e0 de 1f  .<..@....B...
0020  42 da 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66  B...MZ.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69  wabcdefg hi
```



栈指纹OS识别技术(3/5)

- Windows 主机的ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▼ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSSCP 0x00: Default; ECN: 0x00)
  Total Length: 60
  Identification: 0x0000 (0)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 127
  Protocol: ICMP (0x01)
  Header checksum: 0xb9c7 [correct]
  Source: 222.31.66.224 (222.31.66.224)
  Destination: 222.31.66.218 (222.31.66.218)
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x4d5a [correct]
  Identifier: 0x0001
  Sequence number: 1 (0x0001)
  Data (32 bytes)
    Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
    [Length: 32]
0000  5c f3 fc 49 43 e8 f0 4d  a2 2a 61 26 08 00 45 00  \..IC..M .*a&..E.
0010  00 3c 00 00 40 00 7f 01  b9 c7 de 1f 42 e0 de 1f  .<..@....B...
0020  42 da 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66  B...MZ.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69  wabcdefg hi
```



栈指纹OS识别技术(3/5)

- Windows 主机的ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▼ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSSCP 0x00: Default; ECN: 0x00)
  Total Length: 60
  Identification: 0x0000 (0)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 127
  Protocol: ICMP (0x01)
  Header checksum: 0xb9c7 [correct]
  Source: 222.31.66.224 (222.31.66.224)
  Destination: 222.31.66.218 (222.31.66.218)
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x4d5a [correct]
  Identifier: 0x0001
  Sequence number: 1 (0x0001)
▼ Data (32 bytes)
  Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
  [Length: 32]
0000  5c f3 fc 49 43 e8 f0 4d  a2 2a 61 26 08 00 45 00  \..IC..M .*a&..E.
0010  00 3c 00 00 40 00 7f 01  b9 c7 de 1f 42 e0 de 1f  .<..@.... ....B...
0020  42 da 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66  B...MZ.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69  wabcdefg hi
```



栈指纹OS识别技术(3/5)

- Windows 主机的ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▼ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSSCP 0x00: Default; ECN: 0x00)
  Total Length: 60
  Identification: 0x0000 (0)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 127
  Protocol: ICMP (0x01)
  Header checksum: 0xb9c7 [correct]
  Source: 222.31.66.224 (222.31.66.224)
  Destination: 222.31.66.218 (222.31.66.218)
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x4d5a [correct]
  Identifier: 0x0001
  Sequence number: 1 (0x0001)
▼ Data (32 bytes)
  Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
  [Length: 32]
0000  5c f3 fc 49 43 e8 f0 4d  a2 2a 61 26 08 00 45 00  \..IC..M .*a&..E.
0010  00 3c 00 00 40 00 7f 01  b9 c7 de 1f 42 e0 de 1f  .<..@.... ....B...
0020  42 da 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66  B...MZ.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69  wabcdefg hi
```



栈指纹OS识别技术(4/5)

- Linux 主机的 ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▶ Frame 1 (98 bytes on wire, 98 bytes captured)
▶ Ethernet II, Src: f0:4d:a2:2a:61:26 (f0:4d:a2:2a:61:26), Dst: 5c:f3:fc:49:43:e8 (5c:f3:fc:49:43:e8)
└ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 84
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
    Header checksum: 0xf8af [correct]
    Source: 222.31.66.224 (222.31.66.224)
    Destination: 222.31.66.218 (222.31.66.218)
└ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0x00b0 [correct]
    Identifier: 0xf16f
    Sequence number: 1 (0x0001)
    └ Data (56 bytes)
        Data: 90BC8A4EEF00040008090A0B0C0D0E0F1011121314151617...
        [Length: 56]
```

0000	5c f3 fc 49 43 e8 f0 4d	a2 2a 61 26 08 00 45 00	\..IC..M .*a&..E.
0010	00 54 00 00 40 00 40 01	f8 af de 1f 42 e0 de 1f	.T..@. @.B...
0020	42 d4 08 00 00 b0 f1 6f	00 01 9d 8c Ba 4e ef 00	B.....oN..
0030	04 00 08 09 0a 0b 0c 0d	0e 0f 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 ! "#\$%
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&` ()*+, - ./012345
0060	36 37		67



栈指纹OS识别技术(4/5)

- Linux 主机的 ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▶ Frame 1 (98 bytes on wire, 98 bytes captured)
▶ Ethernet II, Src: f0:4d:a2:2a:61:26 (f0:4d:a2:2a:61:26), Dst: 5c:f3:fc:49:43:e8 (5c:f3:fc:49:43:e8)
└ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 84
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
    Header checksum: 0xf8af [correct]
    Source: 222.31.66.224 (222.31.66.224)
    Destination: 222.31.66.218 (222.31.66.218)
└ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0x00b0 [correct]
    Identifier: 0xf16f
    Sequence number: 1 (0x0001)
└ Data (56 bytes)
    Data: 90B0C8A4EEF00040008090A0B0C0D0E0F1011121314151617...
    [Length: 56]
```

0000	5c f3 fc 49 43 e8 f0 4d	a2 2a 61 26 08 00 45 00	\..IC..M .*a&..E.
0010	00 54 00 00 40 00 40 01	f8 af de 1f 42 e0 de 1f	.T..@. @.B...
0020	42 d4 08 00 00 b0 f1 6f	00 01 9d 8c Ba 4e ef 00	B.....oN..
0030	04 00 08 09 0a 0b 0c 0d	0e 0f 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 ! "#\$%
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&` ()*+, - ./012345
0060	36 37		67



栈指纹OS识别技术(4/5)

- Linux 主机的 ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▶ Frame 1 (98 bytes on wire, 98 bytes captured)
▶ Ethernet II, Src: f0:4d:a2:2a:61:26 (f0:4d:a2:2a:61:26), Dst: 5c:f3:fc:49:43:e8 (5c:f3:fc:49:43:e8)
└ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 84
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
    Header checksum: 0xf8af [correct]
    Source: 222.31.66.224 (222.31.66.224)
    Destination: 222.31.66.218 (222.31.66.218)
└ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0x00b0 [correct]
    Identifier: 0xf16f
    Sequence number: 1 (0x0001)
└ Data (56 bytes)
    Data: 90B0C8A4EEF00040008090A0B0C0D0E0F1011121314151617...
    [Length: 56]
```

0000	5c f3 fc 49 43 e8 f0 4d	a2 2a 61 26 08 00 45 00	\..IC..M .*a&..E.
0010	00 54 00 00 40 00 40 01	f8 af de 1f 42 e0 de 1f	.T..@. @.B...
0020	42 d4 08 00 00 b0 f1 6f	00 01 9d 8c Ba 4e ef 00	B.....oN..
0030	04 00 08 09 0a 0b 0c 0d	0e 0f 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 ! "#\$%
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&` ()*+, - ./012345
0060	36 37		67



栈指纹OS识别技术(4/5)

- Linux 主机的 ping 程序实现
 - TTL
 - payload
 - Data
 - Length

```
▶ Frame 1 (98 bytes on wire, 98 bytes captured)
▶ Ethernet II, Src: f0:4d:a2:2a:61:26 (f0:4d:a2:2a:61:26), Dst: 5c:f3:fc:49:43:e8 (5c:f3:fc:49:43:e8)
└ Internet Protocol, Src: 222.31.66.224 (222.31.66.224), Dst: 222.31.66.218 (222.31.66.218)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 84
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
    Header checksum: 0xf8af [correct]
    Source: 222.31.66.224 (222.31.66.224)
    Destination: 222.31.66.218 (222.31.66.218)
└ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0x00b0 [correct]
    Identifier: 0xf16f
    Sequence number: 1 (0x0001)
└ Data (56 bytes)
    Data: 908c8a4eef00040008090a0b0c0d0e0f1011121314151617...
    [Length: 56]
```

0000	5c f3 fc 49 43 e8 f0 4d	a2 2a 61 26 08 00 45 00	\..IC..M .*a&..E.
0010	00 54 00 00 40 00 40 01	f8 af de 1f 42 e0 de 1f	.T..@. @.B...
0020	42 d4 08 00 00 b0 f1 6f	00 01 9d 8c Ba 4e ef 00	B.....oN..
0030	04 00 08 09 0a 0b 0c 0d	0e 0f 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 ! "#\$%
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&` ()*+, - ./012345
0060	36 37		67



栈指纹OS识别技术(5/5)

- 仅仅通过ping即可粗略的判定目标操作系统类型

```
huangwei@huangwei-cuc:~/workspace/teaching/branches/temp$ ping cs.cuc.edu.cn
PING cs.cuc.edu.cn (202.205.18.194) 56(84) bytes of data.
64 bytes from 202.205.18.194: icmp_seq=1 ttl=125 time=0.242 ms
^C64 bytes from 202.205.18.194: icmp_seq=2 ttl=125 time=0.256 ms

--- cs.cuc.edu.cn ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5173ms
rtt min/avg/max/mdev = 0.242/0.249/0.256/0.007 ms

huangwei@huangwei-cuc:~/workspace/teaching/branches/temp$ ping www.baidu.com
PING www.a.shifen.com (119.75.218.45) 56(84) bytes of data.
64 bytes from 119.75.218.45: icmp_seq=1 ttl=51 time=1.98 ms
^C64 bytes from 119.75.218.45: icmp_seq=2 ttl=51 time=2.05 ms

--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5004ms
rtt min/avg/max/mdev = 1.988/2.023/2.058/0.035 ms
```



栈指纹OS识别技术(5/5)

- 仅仅通过ping即可粗略的判定目标操作系统类型

```
huangwei@huangwei-cuc:~/workspace/teaching/branches/temp$ ping cs.cuc.edu.cn
PING cs.cuc.edu.cn (202.205.18.194) 56(84) bytes of data.
64 bytes from 202.205.18.194: icmp_seq=1 ttl=125 time=0.242 ms
^C64 bytes from 202.205.18.194: icmp_seq=2 ttl=125 time=0.256 ms
--- cs.cuc.edu.cn ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5173ms
rtt min/avg/max/mdev = 0.242/0.249/0.256/0.007 ms

huangwei@huangwei-cuc:~/workspace/teaching/branches/temp$ ping www.baidu.com
PING www.a.shifen.com (119.75.218.45) 56(84) bytes of data.
64 bytes from 119.75.218.45: icmp_seq=1 ttl=51 time=1.98 ms
^C64 bytes from 119.75.218.45: icmp_seq=2 ttl=51 time=2.05 ms
--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5004ms
rtt min/avg/max/mdev = 1.988/2.023/2.058/0.035 ms
```

Windows



栈指纹OS识别技术(5/5)

- 仅仅通过ping即可粗略的判定目标操作系统类型

```
huangwei@huangwei-cuc:~/workspace/teaching/branches/temp$ ping cs.cuc.edu.cn
PING cs.cuc.edu.cn (202.205.18.194) 56(84) bytes of data.
64 bytes from 202.205.18.194: icmp_seq=1 ttl=125 time=0.242 ms
^C64 bytes from 202.205.18.194: icmp_seq=2 ttl=125 time=0.256 ms
```

Windows

```
--- cs.cuc.edu.cn ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5173ms
rtt min/avg/max/mdev = 0.242/0.249/0.256/0.007 ms
```

```
huangwei@huangwei-cuc:~/workspace/teaching/branches/temp$ ping www.baidu.com
PING www.a.shifen.com (119.75.218.45) 56(84) bytes of data.
64 bytes from 119.75.218.45: icmp_seq=1 ttl=51 time=1.98 ms
^C64 bytes from 119.75.218.45: icmp_seq=2 ttl=51 time=2.05 ms
```

*nix

```
--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5004ms
rtt min/avg/max/mdev = 1.988/2.023/2.058/0.035 ms
```



- FIN探测
 - 发送一个FIN包给一个打开的端口，一般的行为是不响应，但某些实现例如 MS Windows, BSDI, CISCO, HP/UX, MVS, 和 IRIX 发回一个RST
- BOGUS标记探测
 - 设置一个未定义的TCP 标记（64或128）在SYN包的TCP头里
- 不分段位
 - 许多操作系统开始在送出的一些包中设置IP的 Don't Fragment 位



栈指纹OS识别的其他技术(2/2)

- ACK值
 - 不同实现中一些情况下ACK域的值是不同的
- ICMP错误信息终结
 - 一些操作系统跟从限制各种错误信息的发送率
- SYN洪水限度
 - 如果收到过多的伪造SYN数据包，一些操作系统会停止新的连接尝试
 - 某些操作系统默认只处理8个伪造的SYN包



小结：扫描知识库构建与规则匹配

- 在扫描实践中总结知识库构建
 - TCP/IP协议栈的每一层
 - 报文的发送和接收均有规律可循，有规则可总结
 - 黑盒模糊测试思想
- 规则匹配
 - 不仅仅是字符串静态匹配
 - 可以是基于行为的匹配
 - 会话：传输层 / 应用层
 - 报文交互序列



网络扫描的基本原理

- 报文发送与接收
- 扫描知识库构建与规则匹配
- 扫描报告生成



网络扫描的基本原理

- 报文发送与接收
- 扫描知识库构建与规则匹配
- 扫描报告生成



扫描报告生成

- 数据的可视化呈现技术
—非本课程关注重点



扫描行为的检测与防护

- 扫描行为的检测
 - 被动监听
 - 利用第四章所学的知识、工具和方法
- 扫描行为的防护
 - 防火墙
 - 将在第七章介绍
 - 入侵检测
 - 将在第七章介绍



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



本章内容提要

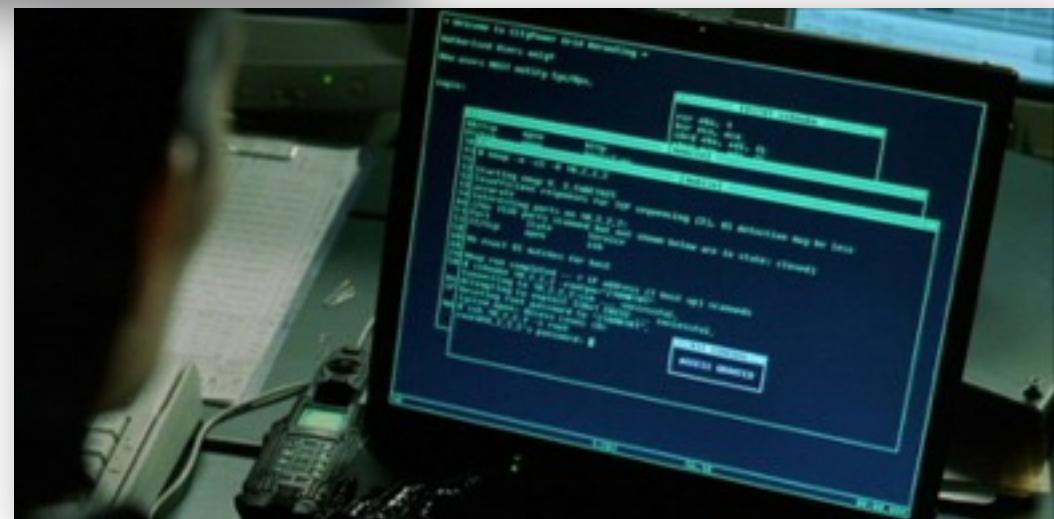
- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



网络扫描工具

- Nmap

```
1 Port      State     Service
2 22/tcp    open      ssh
3
4 No exact OS matches for host
5
6 Nmap run completed -- 1 IP address (1 host up) scanned
7 # sshnuke 10.2.2.2 -rootpw="Z10H0101"
8 Connecting to 10.2.2.2:ssh ... successful.
9 Attempting to exploit SSHv1 CRC32 ... successful.
P Resetting root password to "Z10H0101".
System open: Access Level <9>
m # ssh 10.2.2.2 -l root
root@10.2.2.2's password:
RRF-CONTROL> disable grid nodes 21 - 48
Warning: Disabling nodes 21-48 will disconnect sector 11 (27 nodes)
```





网络扫描工具

- Nmap

```
1 Port      State     Service
2 22/tcp    open      ssh
3
4 No exact OS matches for host
5
6 Nmap run completed -- 1 IP address (1 host up) scanned
7 # sshnuke 10.2.2.2 -rootpw="Z10H0101"
8 Connecting to 10.2.2.2:ssh ... successful.
9 Attempting to exploit SSHv1 CRC32 ... successful.
P Resetting root password to "Z10H0101".
System open: Access Level <9>
m # ssh 10.2.2.2 -l root
root@10.2.2.2's password:
RRF-CONTROL> disable grid nodes 21 - 48
Warning: Disabling nodes 21-48 will disconnect sector 11 (27 nodes)
```

黑客帝国





- Nmap简介
 - Nmap使用方法
 - ping扫描
 - 端口扫描
 - 隐蔽扫描
 - 选项功能
 - 操作系统识别
 - 防火墙/IDS躲避和哄骗
 - Nmap常见应用
-



Nmap简介(1/2)

- Network Mapper的缩写
- 功能特性
 - 多种协议扫描
 - 例如：TCP / UDP / ICMP
 - 支持大多数系统
 - 例如：Linux,MacOSX,Microsoft Windows
 - 支持插件扩展
 - NSE
- Nmap Script Engine



Nmap简介(1/2)

- Network Mapper的缩写
 - 功能特性
 - 多种协议扫描
 - 例如：TCP / UDP / ICMP
 - 支持大多数系统
 - 例如：Linux, MacOSX, Microsoft Windows
 - 支持插件扩展
 - NSE
- Nmap Script Engine



Nmap简介(2/2)

- 一个典型的Nmap扫描

```
root@wzy-desktop: /home/wzy
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@wzy-desktop:/home/wzy# nmap -sT -v 222.31.66.198

Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 14:41 CST
NSE: Loaded 0 scripts for scanning.
Initiating ARP Ping Scan at 14:41
Scanning 222.31.66.198 [1 port]
Completed ARP Ping Scan at 14:41, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:41
Completed Parallel DNS resolution of 1 host. at 14:41, 0.00s elapsed
Initiating Connect Scan at 14:41
Scanning 222.31.66.198 [1000 ports]
Discovered open port 22/tcp on 222.31.66.198
Discovered open port 80/tcp on 222.31.66.198
Discovered open port 25/tcp on 222.31.66.198
Discovered open port 139/tcp on 222.31.66.198
Discovered open port 445/tcp on 222.31.66.198
Discovered open port 10000/tcp on 222.31.66.198
Completed Connect Scan at 14:41, 0.01s elapsed (1000 total ports)
Host 222.31.66.198 is up (0.00021s latency).
Interesting ports on 222.31.66.198:
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
MAC Address: F0:4D:A2:2A:04:C9 (Unknown)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
Raw packets sent: 1 (42B) | Rcvd: 1 (42B)
```



Nmap简介(2/2)

- 一个典型的Nmap扫描

```
root@wzy-desktop: /home/wzy
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@wzy-desktop:/home/wzy# nmap -sT -v 222.31.66.198
Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 14:41 CST
NSE: Loaded 0 scripts for scanning.
Initiating ARP Ping Scan at 14:41
Scanning 222.31.66.198 [1 port]
Completed ARP Ping Scan at 14:41, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:41
Completed Parallel DNS resolution of 1 host. at 14:41, 0.00s elapsed
Initiating Connect Scan at 14:41
Scanning 222.31.66.198 [1000 ports]
Discovered open port 22/tcp on 222.31.66.198
Discovered open port 80/tcp on 222.31.66.198
Discovered open port 25/tcp on 222.31.66.198
Discovered open port 139/tcp on 222.31.66.198
Discovered open port 445/tcp on 222.31.66.198
Discovered open port 10000/tcp on 222.31.66.198
Completed Connect Scan at 14:41, 0.01s elapsed (1000 total ports)
Host 222.31.66.198 is up (0.00021s latency).

Interesting ports on 222.31.66.198:
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
MAC Address: F0:4D:A2:2A:84:C9 (Unknown)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
Raw packets sent: 1 (42B) | Rcvd: 1 (42B)
```



Nmap应用程序目录结构解析

- 可执行程序目录
 - /usr/bin
- 文档目录
 - /usr/share/man
 - /usr/share/doc/nmap
- 插件目录
 - /usr/share/nmap/nselib
 - /usr/share/nmap/scripts
- 知识库目录
 - /usr/share/nmap



Nmap应用程序目录结构解析——可执行程序目录

- /usr/bin/ncat
 - 网络数据传输/重定向/加密/调试工具
- /usr/bin/ndiff
 - Nmap扫描报告差异比对工具
- /usr/bin/nmap
 - 主机和网络扫描工具



Nmap应用程序目录结构解析——插件目录(1/2)

- /usr/share/nmap/nselib

base64.lua	listop.lua	packet.lua	ssh2.lua
comm.lua	match.lua	pop3.lua	stdnse.lua
datafiles.lua	msrpc.lua	shortport.lua	strbuf.lua
dns.lua	msrpcperformance.lua	smbauth.lua	tab.lua
http.lua	msrpctypes.lua	smb.lua	unpwd.lua
imap.lua	netbios.lua	snmp.lua	url.lua
ipOps.lua	nsedebug.lua	ssh1.lua	



Nmap应用程序目录结构解析——插件目录(2/2)

- /usr/share/nmap/scripts

asn-query.nse	imap-capabilities.nse	smb-enum-users.nse
auth-owners.nse	irc-info.nse	smb-os-discovery.nse
auth-spoof.nse	ms-sql-info.nse	smb-pwdump.nse
banner.nse	mysql-info.nse	smb-security-mode.nse
daytime.nse	nbstat.nse	smb-server-stats.nse
dns-random-srcport.nse	p2p-conficker.nse	smb-system-info.nse
dns-random-txid.nse	pop3-brute.nse	smtp-commands.nse
dns-recursion.nse	pop3-capabilities.nse	smtp-open-relay.nse
dns-zone-transfer.nse	pptp-version.nse	smtp-strangeport.nse
finger.nse	realvnc-auth-bypass.nse	sniffer-detect.nse
ftp-anon.nse	robots.txt.nse	snmp-brute.nse
ftp-bounce.nse	rpcinfo.nse	snmp-sysdescr.nse
ftp-brute.nse	script.db	socks-open-proxy.nse
html-title.nse	skypev2-version.nse	sql-injection.nse
http-auth.nse	smb-brute.nse	ssh-hostkey.nse
http-iis-webdav-vuln.nse	smb-check-vulns.nse	sshv1.nse
http-open-proxy.nse	smb-enum-domains.nse	sslv2.nse
http-passwd.nse	smb-enum-processes.nse	telnet-brute.nse
http-trace.nse	smb-enum-sessions.nse	upnp-info.nse
iax2-version.nse	smb-enum-shares.nse	whois.nse

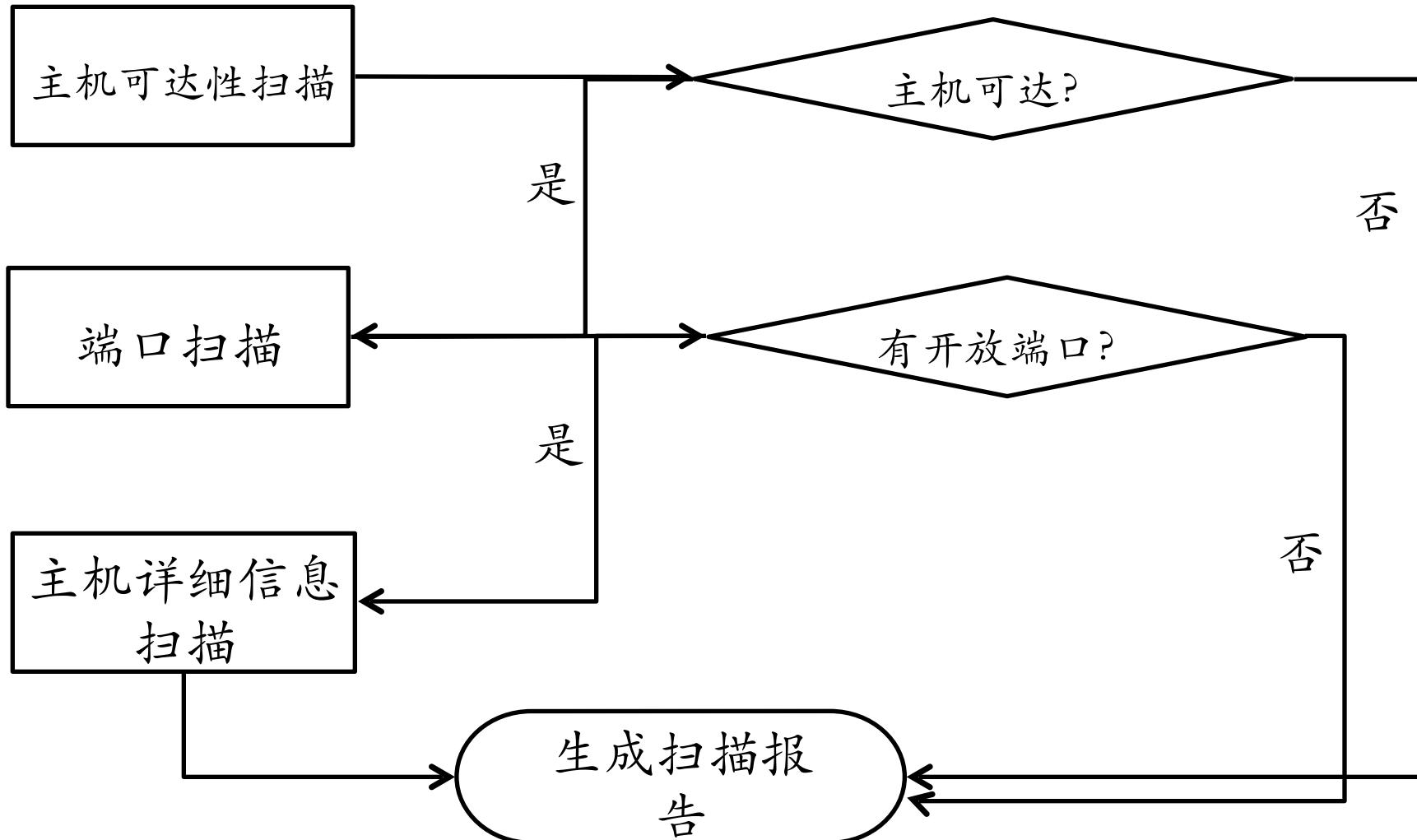


- /usr/share/nmap

文件	作用说明
<i>nmap-mac-prefixes</i>	MAC地址对应厂家知识库
<i>nmap-os-db</i>	操作系统指纹知识库
<i>nmap-protocols</i>	协议类型字段标识知识库
<i>nmap-rpc</i>	RPC应用类型指纹知识库
<i>nmap-service-probes</i>	应用程序交互行为指纹知识库
<i>nmap-services</i>	端口与应用/服务映射关系知识库



Nmap扫描一般流程





Nmap使用方法——Ping扫描(1/2)

- Ping扫描
 - 使用Nmap扫描整个网络寻找目标，通过使用-sP命令，进行Ping扫描
- Ping扫描原理
 - Nmap给每个扫描到的主机发送一个ICMP echo和一个TCP ACK，主机对任何一种的响应都会被Nmap得到
- Ping扫描举例



Nmap使用方法——Ping扫描(2/2)

- 举例：扫描指定网段的主机

The screenshot shows a terminal window with the following content:

```
root@wzy-desktop: /home/wzy
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@wzy-desktop:/home/wzy# nmap -sP 222.31.66.190-201

Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 14:55 CST
Host 222.31.66.190 is up (0.00014s latency).
MAC Address: F0:4D:A2:2A:04:C9 (Unknown)
Host 222.31.66.191 is up (0.00012s latency).
MAC Address: F0:4D:A2:E9:9E:97 (Unknown)
Host 222.31.66.192 is up (0.011s latency).
MAC Address: 00:1F:E2:5D:72:94 (Hon Hai Precision Ind. Co.)
Host 222.31.66.193 is up (0.011s latency).
MAC Address: 00:1F:E2:4E:72:19 (Hon Hai Precision Ind. Co.)
Host 222.31.66.196 is up (0.012s latency).
MAC Address: 00:22:68:56:62:34 (Hon Hai Precision Ind. Co.)
Host 222.31.66.198 is up (0.00057s latency).
MAC Address: 00:14:97:02:1E:AC (Zhiyuan Eletronics Co.)
Host 222.31.66.199 is up (0.00027s latency).
MAC Address: F0:DE:F1:16:BA:3B (Unknown)
Host 222.31.66.200 is up (0.00045s latency).
MAC Address: 00:13:72:3D:03:7E (Dell)
Nmap done: 12 IP addresses (8 hosts up) scanned in 0.31 seconds
```



Nmap使用方法——端口扫描

- 端口扫描原理
- -sT指令扫描介绍



Nmap使用方法——端口扫描

- 端口扫描原理
- -sT指令扫描介绍

通过Ping扫描发现目标网络上运行的主机
下一步就是进行端口扫描



端口扫描原理

- Nmap使用connect()系统调用打开目标机上相关端口的连接,并完成三次TCP握手.一个tcp连接扫描使用"-sT"命令
- 举例："-sT"命令发送一个SYN扫描探测主机

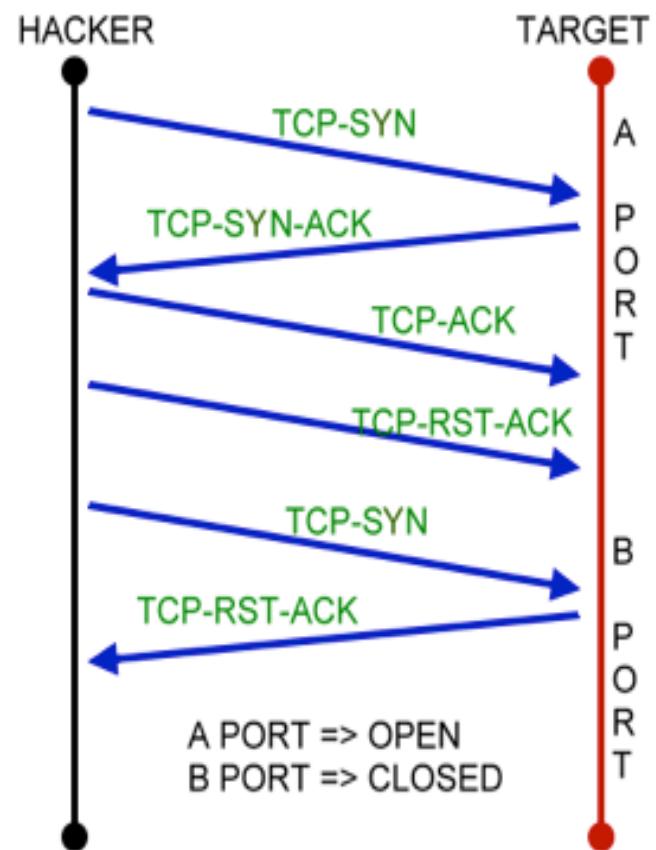
```
root@wzy-desktop: ~
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@wzy-desktop: ~ # nmap -sT 222.31.66.218

Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 15:13 CST
Interesting ports on 222.31.66.218:
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
2049/tcp  open  nfs
8000/tcp  open  http-alt
8081/tcp  open  blackice-icecap
9999/tcp  open  abyss
MAC Address: 5C:F3:FC:49:43:E8 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```



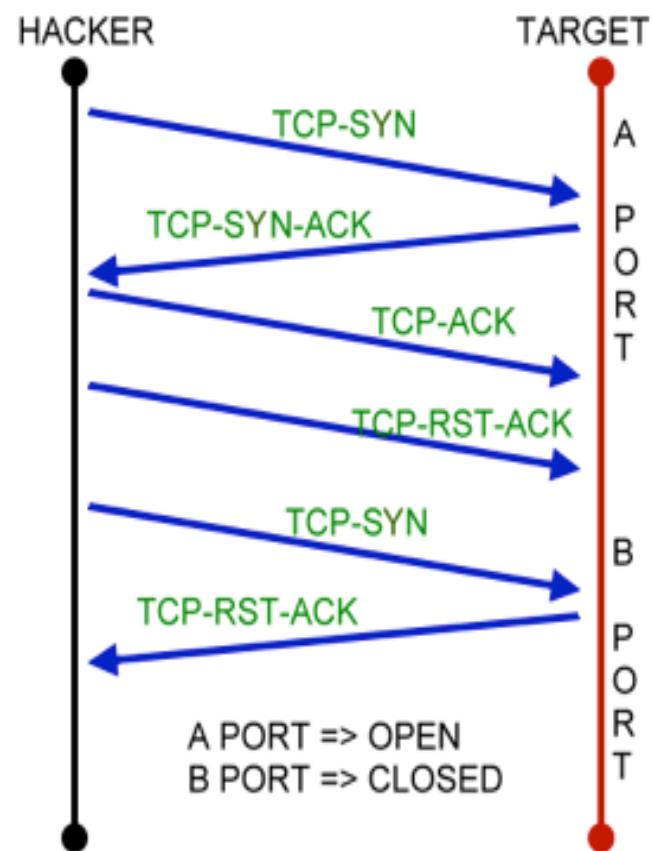
-sT指令扫描介绍





-sT指令扫描介绍

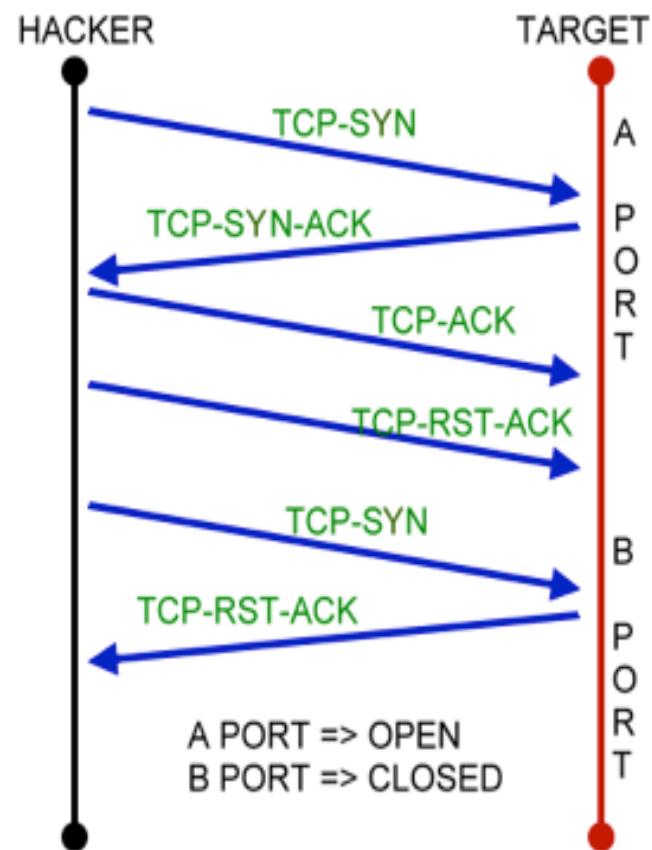
- 标准的TCP 全连接扫描方式





-sT指令扫描介绍

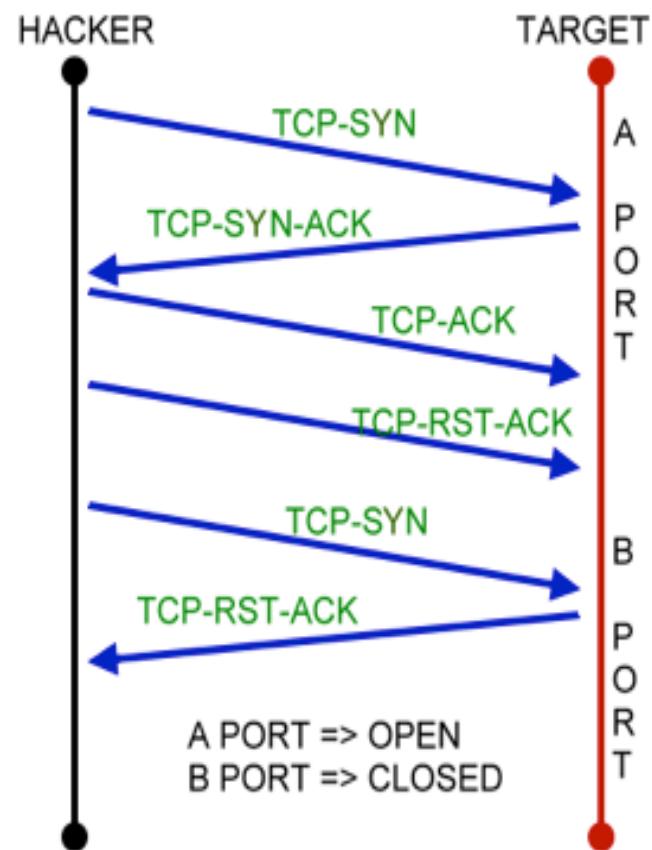
- 标准的TCP 全连接扫描方式
- 非root用户也可以执行





-sT指令扫描介绍

- 标准的TCP 全连接扫描方式
- 非root用户也可以执行
- 对扫描的PORT发送TCP-SYN 封包
 - 有回应 TCP SYN-ACK封包
 - 没回应 TCP SYN-RST 封包





- 隐蔽扫描 (stealth scanning)
 - 是指攻击者不愿意在扫描时其信息被记录在目标系统上而采用的扫描方式
- 隐蔽扫描原理
 - 进行SYN扫描，开放的端口都ACK|SYN响应
 - 攻击者发送RST代替ACK
 - 关闭的端口对最初的SYN信号的响应也会是RST



Nmap使用方法——隐蔽扫描(2/2)

- 举例："-sS"命令发送一个SYN扫描探测主机

```
root@wzy-desktop: ~
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@wzy-desktop:~# nmap -sS 222.31.66.218

Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 15:39 CST
Interesting ports on 222.31.66.218:
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
2049/tcp  open  nfs
8000/tcp  open  http-alt
8081/tcp  open  blackice-icecap
9999/tcp  open  abyss
MAC Address: 5C:F3:FC:49:43:E8 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.32 seconds
```



Nmap使用方法——隐蔽扫描(2/2)

- 举例："-sS"命令发送一个SYN扫描探测主机

```
root@wzy-desktop: ~
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@wzy-desktop:~# nmap -sS 222.31.66.218

Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 15:39 CST
Interesting ports on 222.31.66.218:
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
2049/tcp  open  nfs
8000/tcp  open  http-alt
8081/tcp  open  blackice-icecap
9999/tcp  open  abyss
MAC Address: 5C:F3:FC:49:43:E8 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.32 seconds
```

三次握手不完全，很多主机都不会记录这样的探测



高级选项功能(1/3)

- 操作系统的识别

- 方法,TCP/IP上的指纹带有-O选项决定远程操作系统的类型.这可以和一个端口扫描结合使用,但不能和ping扫描结合使用

- 举例,-O选项探测目标主机的操作系统



高级选项功能(1/3)

- 操作系统的识别

一方法,TCP/IP上的指纹带有-O选项决定远程操作系统的类型.这可以和一个端口扫描结合使用,但不能和ping扫描结合使用

—举例,-O选项探测目标主机的操作系统

```
root@wzy-desktop:~# nmap -sS -O 222.31.66.218
Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-22 15:55 CST
Interesting ports on 222.31.66.218:
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
2049/tcp  open  nfs
8000/tcp  open  http-alt
8081/tcp  open  blackice-icecap
9999/tcp  open  abyss
MAC Address: 5C:F3:FC:49:43:E8 (Unknown)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.24 - 2.6.28
Network Distance: 1 hop
```



高级选项功能(2/3)

- 防火墙/IDS躲避和哄骗

- 使用原因

- 检验网络安全状态最有效的方法之一是尝试哄骗网络

- 所有主流的IDS都包含了检测Nmap扫描的规则

- 实现方法

- 使用-D选项，这是一种带有诱骗模式的扫描，在远程主机的连接记录里会记下所有你所指定的诱骗性的地址

注意：你用来诱骗的主机必须是开放的



高级选项功能(3/3)

- 举例：
 - 在扫描时使用诱骗技术避免引起管理员的注意
 - nmap -sS 10.16.8.18 -D 10.6.8.29
 - 10.16.8.18——扫描目标设备/计算机
 - 10.16.8.29——诱骗地址



高级选项功能(3/3)

- 举例：
 - 在扫描时使用诱骗技术避免引起管理员的注意
 - nmap -sS 10.16.8.18 -D 10.6.8.29
 - 10.16.8.18——扫描目标设备/计算机
 - 10.16.8.29——诱骗地址

目标主机的安全日志中只会显示诱骗地址



Nmap常见应用(1/2)

- 获得远程主机的端口信息和识别主机操作系统
 - nmap -sS -P0 -sV -O <target>
 - P0 选项允许你关闭ICMP ping
 - sV 选项启用版本检测
 - O 表示试图识别远程操作系统
- 获得网络中所有存活的主机
 - nmap -sP 222.31.66.128/25
 - 针对特定子网



Nmap常见应用(2/2)

- 获取指定开放端口的服务器列表
 - nmap -sT -p 80 -oG – 222.31.66.* | grep open
 - 改变-p的参数可指定端口
- ping一个范围内的IP地址
 - nmap -sP 222.31.66.190-210
 - namp接受多种类型的地址符号
- 寻找一个给定子网中未使用的ip
 - nmap -T4 -sP 192.168.2.0/24 && grep ‘00:00:00:00:00’ /proc/net/arp



Nmap命令选项小结(1/3)

序号	扫描原理	命令选项	说明
1	ARP扫描	<code>-sP --script=sniffer-detect</code>	需要root权限
2	ICMP echo扫描	<code>-sP -PE <host_ip></code>	<ul style="list-style-type: none">需要root权限“D圭二仁也行于如可计此太松测
3	ICMP sweep扫描	<code>-sP -PE <ip_block></code>	需要root权限
4	ICMP Broadcast扫描	<code>-sP -PE <广播/组播IP地址></code>	需要root权限
5	ICMP non-echo扫描	<code>-sP -PP/-PM</code>	<ul style="list-style-type: none">需要root权限<code>-PP timestamp ICMP</code>请求
6	TCP Connect扫描	<code>-sT -p <port_number></code>	
7	UDP扫描	<code>-sU -p <port_number></code>	
8	TCP SYN扫描	<code>-sS <host_ip></code>	需要root权限
9	TCP FIN扫描	<code>-sF <host_ip></code>	需要root权限



Nmap命令选项小结(2/3)

序号	扫描原理	命令选项	说明
10	TCP Xmas扫描	<code>-sX <host_ip></code>	需要root权限
11	TCP Null扫描	<code>-sN <host_ip></code>	需要root权限
12	ACK扫描	<code>-sA <host_ip></code>	需要root权限
13	IDLE扫描	<code>-sI <zombie host[:probeport]> <host_ip></code>	需要root权限
14	栈指纹OS识别	<code>-O</code>	需要root权限
15	TCP间接扫描	<code>-S <spoof ip> <host_ip></code>	<ul style="list-style-type: none">需要root权限手工嗅探<spoof>
16	IP分片扫描	<code>-f; --mtu <mtu></code>	<ul style="list-style-type: none">需要root权限数据包分片



Nmap命令选项小结(3/3)

序号	扫描原理	命令选项	说明
17	自定义TCP报文头部	--scanflags <flags>	需要root权限
18	自定义IP报文头部	--ip-options <options>	需要root权限
19	伪造源MAC地址	--spoof-mac <mac_address/prefix/vendor name>	需要root权限
20	构造虚假TCP/UDP校验和	--badsum	需要root权限



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



本章内容提要

- 网络扫描与信息收集
- 网络扫描原理
- 网络扫描工具
- 实验讲解



实验讲解

- 实验一：局域网主机扫描
- 实验二：局域网拓扑发现



实验一：局域网主机扫描

- 实验目的
- 实验工具
- 实验步骤
- 实验分析



实验目的和实验工具

- 通过Nmap对局域网主机进行扫描，了解常用的Nmap指令
- Nmap（网络映射器）
——一款开源代码的网络探测和安全审核工具



实验步骤(1/3)

- 查看一下自己的IP地址

—ifconfig | grep inet

```
wzy@wzy-desktop: ~
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
wzy@wzy-desktop:$ ifconfig |grep inet
inet 地址:222.31.66.248 广播:222.31.66.255 掩码:255.255.255.128
inet6 地址: 2001:250:217:307:f24d:a2ff:feec:63df/64 Scope:Global
inet6 地址: fe80::f24d:a2ff:feec:63df/64 Scope:Link
inet 地址:127.0.0.1 掩码:255.0.0.0
inet6 地址: ::1/128 Scope:Host
inet 地址:10.6.8.74 点对点:10.6.8.73 掩码:255.255.255.255
```

- 扫描内网存活的主机,选择目标主机

—nmap -sP -A 4 222.31.66.128/25

```
wzy@wzy-desktop: ~$ sudo nmap -sP 222.31.66.200-218
Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-26 21:00 CST
Host 222.31.66.200 is up (0.00023s latency).
MAC Address: 00:13:72:3D:03:7E (Dell)
Host 222.31.66.207 is up (0.00023s latency).
MAC Address: 00:12:3F:78:15:CF (Dell)
```



实验步骤(2/3)

- 扫描该目标主机常见服务
—nmap 222.31.66.200

```
wzy@wzy-desktop: ~
文件(E) 编辑(E) 查看(V) 终端(T) 帮助(H)
wzy@wzy-desktop:~$ nmap 222.31.66.200

Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-26 21:08 CST
Interesting ports on 222.31.66.200:
Not shown: 990 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
444/tcp   open  snpp
445/tcp   open  microsoft-ds
990/tcp   open  ftps
3389/tcp  open  ms-term-serv
8008/tcp  open  http

Nmap done: 1 IP address (1 host up) scanned in 4.86 seconds
```



实验步骤(3/3)

- 扫描目标主机的操作系统

—nmap -O 222.31.66.200

```
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows XP|2003
OS details: Microsoft Windows XP SP2 or Server 2003 SP2
Network Distance: 1 hop
```

- 查看目标主机各个服务详细版本信息

—nmap -sV 222.31.66.200

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	Serv-U ftptd 7.0
22/tcp	open	ssh	(protocol 2.0)
80/tcp	open	http	Microsoft IIS webserver 6.0
139/tcp	open	netbios-ssn	
443/tcp	open	ssl/https?	
444/tcp	open	ssl/http	Apache httpd
445/tcp	open	microsoft-ds	Microsoft Windows 2003 microsoft-ds
990/tcp	open	ftp	Serv-U ftptd 7.0
3389/tcp	open	microsoft-rdp	Microsoft Terminal Service
8008/tcp	open	http?	



实验分析

- Nmap是一个网络探测和安全扫描程序，系统管理者和个人可以使用这个软件扫描大型网络，获取目标主机运行和提高的服务信息
- 尽量在root用户权限下使用Nmap
- Nmap运行通常会得到被扫描主机端口的列表



实验二：局域网拓扑发现

- 实验目的
- 实验工具
- 实验步骤
- 实验分析



实验目的和实验工具

- 通过ZenMap工具的使用了解整个局域网的拓扑结构
- ZenMap是经典安全扫描工具Nmap的一个官方图形界面版本，是一个跨平台的开源应用
 - 支持使用traceroute和ping命令
 - 可以保存经常使用测试文件的配置
 - 可以绘制拓扑映射图
 - 支持对不同的扫描结果进行比较
 - 拥有大量默认的扫描设置可供选择
 - 可以对扫描结果进行搜索



实验步骤(1/3)

- 扫描同一局域网内的主机

The screenshot shows the Zenmap interface. The top bar has tabs for Scan, Tools, Profile, and Help. The main area has fields for Target (222.31.66.128/25) and Profile (Intense scan). Below these are two text input fields: 'Command' containing the Nmap command and 'Nmap Output' which displays the scan log.

Command: nmap -T4 -A -v -PE -PS22,25,80 -PA21,23,80,3389 222.31.66.128/25

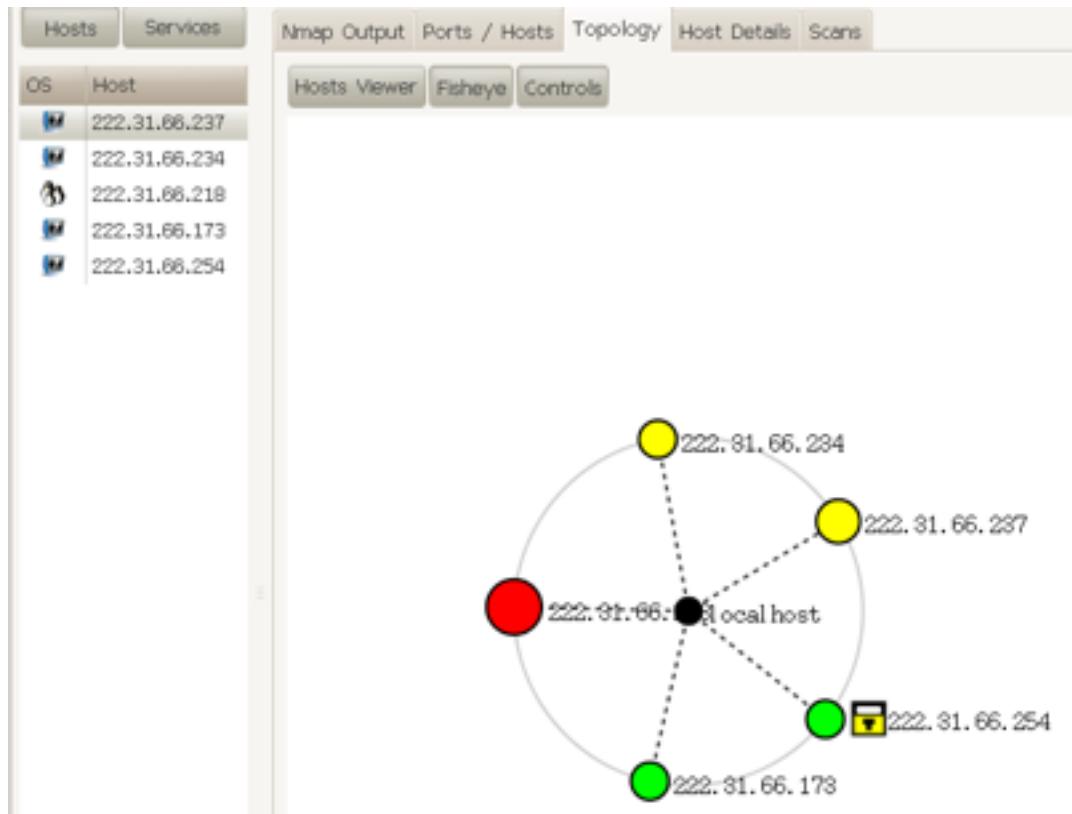
Nmap Output:

```
nmap -T4 -A -v -PE -PS22,25,80 -PA21,23,80,3...  
Starting Nmap 5.00 ( http://nmap.org ) at 2011-09-27  
18:19 CST  
NSE: Loaded 30 scripts for scanning.  
Initiating ARP Ping Scan at 18:19  
Scanning 120 hosts [1 port/host]  
Completed ARP Ping Scan at 18:19, 0.45s elapsed (120  
total hosts)  
Initiating Parallel DNS resolution of 120 hosts. at 18:19  
Completed Parallel DNS resolution of 120 hosts. at 18:19,  
0.11s elapsed  
Initiating Parallel DNS resolution of 1 host. at 18:19  
Completed Parallel DNS resolution of 1 host. at 18:19,  
0.06s elapsed  
Initiating SYN Stealth Scan at 18:19  
Scanning 56 hosts [1000 ports/host]  
Discovered open port 80/tcp on 222.31.66.157  
Discovered open port 80/tcp on 222.31.66.188  
Discovered open port 80/tcp on 222.31.66.200  
Discovered open port 80/tcp on 222.31.66.224  
Discovered open port 80/tcp on 222.31.66.215
```



实验步骤(2/3)

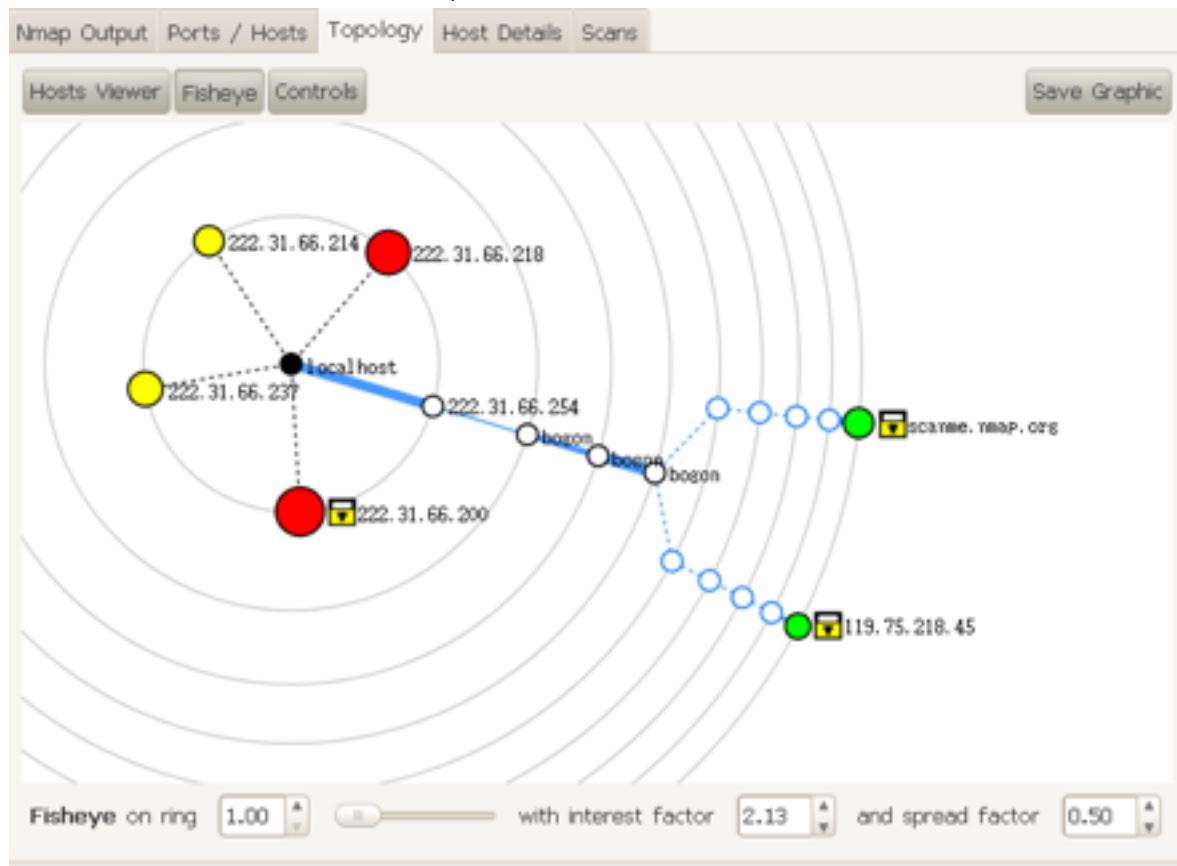
- 扫描同一局域网内的几个主机，观察局域网内的拓扑结构





实验步骤(3/3)

- 扫描外部IP地址，例如scanme.nmap.org,查看整体拓扑结构图





实验分析

- 在 Zenmap 的 Command 输入框中加入 --traceroute 参数，可以在 Topology 选项卡中可以查看从本机到 targethost 的路由图。点击 Controls 按钮以后弹出的对话框中可以根据需要显示网络拓扑节点上主机的详细信息
- 通过拓扑图，我们可以明确局域网的整个拓扑结构，明确我们的网关IP，以及到达外部IP经过的拓扑结构



参考文献

- ① O. Arkin, Network scanning techniques. PubliCom, 1999.
<http://exploitworld.pc-freak.net/info/arkin%20network%20scanning%20techniques.pdf>
- ② <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>
- ③ <http://en.wikipedia.org/wiki/Winsock>
- ④ http://en.wikipedia.org/wiki/Berkeley_sockets
- ⑤ Nmap 官方指南 <http://nmap.org/man/zh/>
- ⑥ Mastering the Nmap Scripting Engine by Fyodor and David Fifield <http://insecure.org/presentations/BHDC10/>



课后思考题

- 通过本章网络扫描基本原理的学习，试推测
 - 应用程序版本信息扫描原理
 - 网络漏洞扫描原理
- 网络扫描知识库的构建方法有哪些？