

软件与系统安全分析

拒绝服务攻击及其防御技术

国家计算机网络入侵防范中心

张玉清

本章内容安排

- **6.1** 拒绝服务攻击概述
- **6.2** 典型拒绝服务攻击技术
- **6.3** 分布式拒绝服务攻击
- **6.4** 拒绝服务攻击的防御
- **6.5** 分布式拒绝服务攻击的防御
- **6.6** 小结



6.1 拒绝服务攻击概述

- 6.1.1 拒绝服务攻击的概念
- 6.1.2 拒绝服务攻击的类型

6.1.1 拒绝服务攻击的概念

- ❑ 拒绝服务（ **Denial of Service**，简称**DoS**），是一种简单的破坏性攻击，通常是利用传输协议中的某个弱点、系统存在的漏洞、或服务的漏洞，对目标系统发起大规模的进攻，用超出目标处理能力的海量数据包消耗可用系统资源、带宽资源等，或造成程序缓冲区溢出错误，致使其无法处理合法用户的正常请求，无法提供正常服务，最终致使网络服务瘫痪，甚至系统死机。
- ❑ 简单的说，拒绝服务攻击就是让攻击目标瘫痪的一种“损人不利己”的攻击手段。

6.1.1 拒绝服务攻击的概念

- ❑ 历史上最著名的拒绝服务攻击服务恐怕要数**Morris**蠕虫事件，**1988年11月**，全球众多连在因特网上的计算机在数小时内无法正常工作，这次事件中遭受攻击的包括**5个**计算机中心和**12个**地区结点，连接着政府、大学、研究所和拥有政府合同的**25万台**计算机。这次病毒事件，使计算机系统直接经济损失达**9600**万美元。
- ❑ 许多知名网站如**Yahoo、eBay、CNN、百度、新浪**等都曾遭受过**DoS**攻击。

6.1.1 拒绝服务攻击的概念

□ 国内首例拒绝服务案

2007年5月，北京市市公安局网监处接到了网络游戏运营商联众公司的报案，该公司托管在北京、上海、石家庄的多台服务器遭受到**200万个**不同程度的大流量**DDOS**拒绝服务攻击包，长达近一个月，公司经济损失达数百万元。



6.1.1 拒绝服务攻击的概念

- 国家域名解析节点遭受到的有史以来规模最大的拒绝服务攻击（来源：**CNNIC**）
- **2013年8月25日凌晨**，**.cn**根域名遭受有史以来最大规模**DDos**攻击，**.cn**的根域授权**DNS**全线故障，造成大量以**.cn**和**.com.cn**结尾的域名无法访问。直到当日凌晨**4**点左右才有部分恢复。距本次事故发生一个月之后，本月**24**号，**CNNIC**和工信部终于揪出了本次攻击事件的始作俑者——一名来自山东青岛的黑客。

6.1.1 拒绝服务攻击的概念

- ❑ 拒绝服务攻击可能是蓄意的，也可能是偶然的。
- ❑ 当未被授权的用户过量使用资源时，攻击是蓄意的；当合法用户无意地操作而使得资源不可用时，则是偶然的。
- ❑ 应该对两种拒绝服务攻击都采取预防措施。但是拒绝服务攻击问题也一直得不到合理的解决，究其原因是因为这是由于网络协议本身的安全缺陷造成的。

6.1.2 拒绝服务攻击的类型

- ❑ 最常见的**DoS**攻击是利用合理的服务请求来占用过多的服务资源，致使服务超载，无法响应其他的请求。
- ❑ 这些服务资源包括网络带宽、文件系统空间容量、开放的进程、向内的连接等。
- ❑ 这种攻击会导致资源的匮乏，无论计算机的处理速度多么快，内存容量多么大，互联网带宽多么大都无法避免这种攻击带来的后果。

6.1.2 拒绝服务攻击的类型

从实施**DoS**攻击所用的思路来看，**DoS**攻击可以分为：

□ 滥用合理的服务请求

- 过度地请求系统的正常服务，占用过多服务资源，致使系统超载。这些服务资源通常包括网络带宽、文件系统空间容量、开放的进程或者连接数等

□ 制造高流量无用数据

- 恶意地制造和发送大量各种随机无用的数据包，用这种高流量的无用数据占据网络带宽，造成网络拥塞

□ 利用传输协议缺陷

- 构造畸形的数据包并发送，导致目标主机无法处理，出现错误或崩溃，而拒绝服务

□ 利用服务程序的漏洞

- 针对主机上的服务程序的特定漏洞，发送一些有针对性的特殊格式的数据，导致服务处理错误而拒绝服务

6.1.2 拒绝服务攻击的类型

按漏洞利用方式分类，**DoS**攻击可以分为：

□ 特定资源消耗类

- 主要利用TCP/IP协议栈、操作系统或应用程序设计上的缺陷，通过构造并发送特定类型的数据包，使目标系统的协议栈空间饱和、操作系统或应用程序资源耗尽或崩溃，从而达到DoS的目的。

□ 暴力攻击类

- 依靠发送大量的数据包占据目标系统有限的网络带宽或应用程序处理能力来达到攻击的目的。通常暴力攻击需要比特定资源消耗攻击使用更大的数据流量才能达到目的。

6.1.2 拒绝服务攻击的类型

按攻击数据包发送速率变化方式，**DoS**攻击可分为：

□ 固定速率

□ 可变速率

- 根据数据包发送速率变化模式，又可以分为震荡变化型和持续增加型。
- 震荡变化型变速率发送方式间歇性地发送数据包，使入侵检测系统难以发现持续的异常。
- 持续增加型变速率发送方式可以使攻击目标的性能缓慢下降，并可以误导基于学习的检测系统产生错误的检测规则。

6.1.2 拒绝服务攻击的类型

按攻击可能产生的影响，**DoS**攻击可以分为：

□ 系统或程序崩溃类

- 根据可恢复的程度，系统或程序崩溃类又可以分为：自我恢复类、人工恢复类、不可恢复类等。
- 自我恢复类是指当攻击停止后系统功能可自动恢复正常。人工恢复类是指系统或服务程序需要人工重新启动才能恢复。不可恢复类是指攻击给目标系统的硬件设备、文件系统等造成了不可修复性的损坏。

□ 服务降级类

- 系统对外提供服务的服务下降

6.2 典型拒绝服务攻击技术

- 6.2.1 Ping of Death
- 6.2.2 泪滴 (Teardrop)
- 6.2.3 IP欺骗DoS攻击
- 6.2.4 UDP洪水
- 6.2.5 SYN洪水
- 6.2.6 Land攻击
- 6.2.7 Smurf攻击
- 6.2.8 Fraggle攻击
- 6.2.9 电子邮件炸弹
- 6.2.10 畸形消息攻击
- 6.2.11 Slashdot effect
- 6.2.12 WinNuke攻击

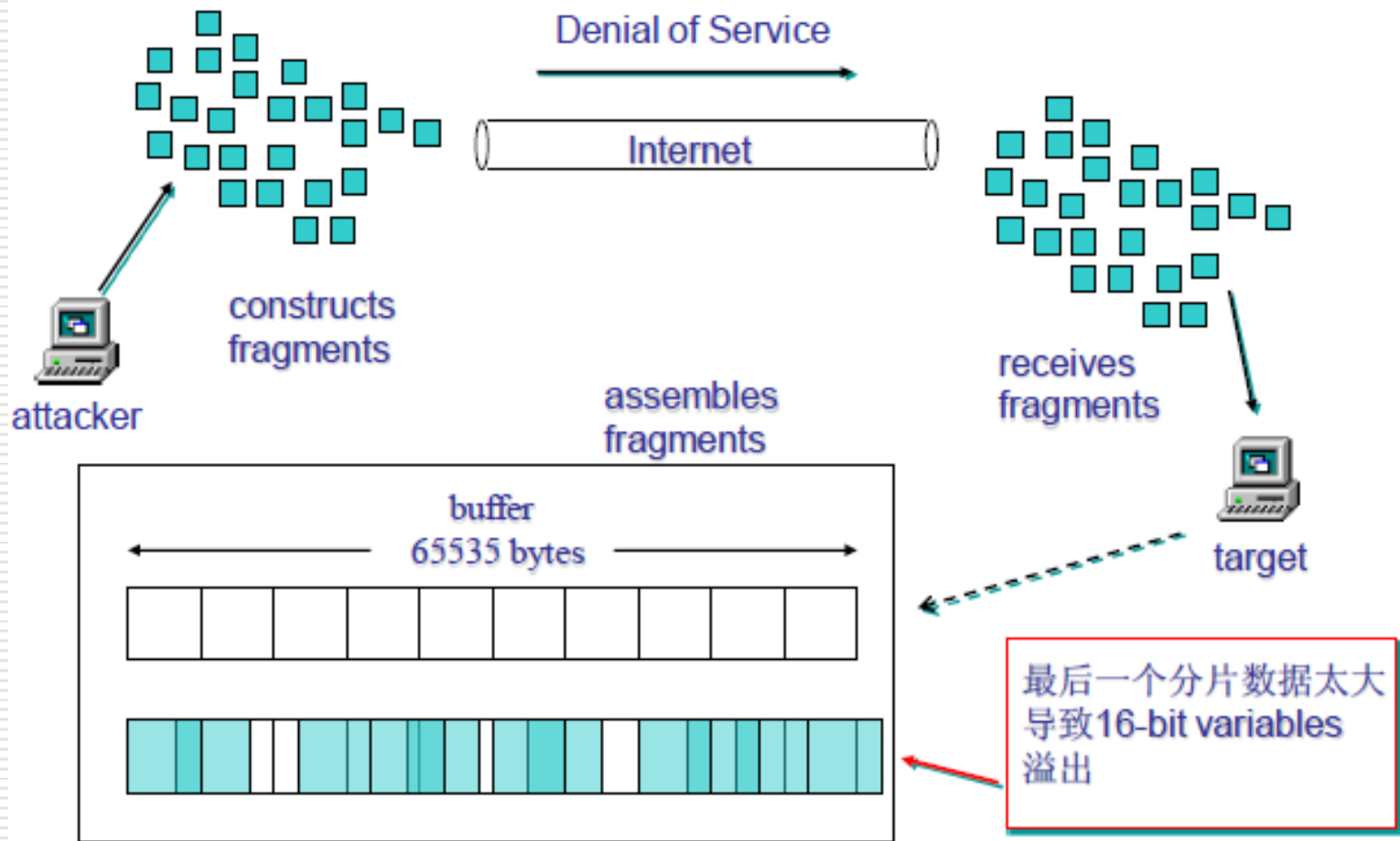
6.2.1 Ping of Death

- ❑ **Ping**是一个非常著名的程序，这个程序的目的是为了测试另一台主机是否可达。现在所有的操作系统上几乎都有这个程序，它已经成为系统的一部分。
- ❑ **Ping**程序的目的是为了查看网络上的主机是否处于活动状态。
- ❑ 通过发送一份**ICMP**回显请求报文给目的主机，并等待返回**ICMP**回显应答，根据回显应答的内容判断目的主机的状况。

6.2.1 Ping of Death

- ❑ **Ping**之所以会造成伤害是源于早期操作系统在处理**ICMP**协议数据包存在漏洞。
- ❑ **ICMP**协议的报文长度是固定的，大小为**64KB**，早期很多操作系统在接收**ICMP**数据报文的时候，只开辟**64KB**的缓存区用于存放接收到的数据包。
- ❑ 一旦发送过来的**ICMP**数据包的实际尺寸超过**64KB(65536B)**，操作系统将收到的数据报文向缓存区填写时，报文长度大于**64KB**，就会产生一个缓存溢出，结果将导致**TCP/IP**协议堆栈的崩溃，造成主机的重新启动或是死机。

6.2.1 Ping of Death



6.2.1 Ping of Death

- **Ping**程序有一个“-l”参数可指定发送数据包的尺寸，因此，使用**Ping**这个常用小程序就可以简单地实现这种攻击。例如通过这样一个命令：

Ping -l 65540 192.168.1.140

- 如果对方主机存在这样一个漏洞，就会形成一次拒绝服务攻击。这种攻击被称为“死亡之**Ping**”。

6.2.1 Ping of Death

- 现在的操作系统都已对这一漏洞进行了修补。对可发送的数据包大小进行了限制。
- 在**Windows 7**操作系统中输入这样的命令：

Ping -l 65535 192.168.1.140

系统返回这样的信息：

Bad value for option -l, valid range is from 0 to 65500.

```
Microsoft Windows [版本 6.1.7601]  
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。  
  
C:\Users\Administrator>ping -l 65540  
选项 -l 的值有错误, 有效范围从 0 到 65500。
```

6.2.1 Ping of Death

□ Ping Of Death攻击的攻击特征、检测方法和反攻击方法总结如下：

- **攻击特征**：该攻击数据包大于65535个字节。由于部分操作系统接收到长度大于65535字节的数据包时，就会造成内存溢出、系统崩溃、重启、内核失败等后果，从而达到攻击的目的。
- **检测方法**：判断数据包的大小是否大于65535个字节。
- **反攻击方法**：使用新的补丁程序，当收到大于65535个字节的数据包时，丢弃该数据包，并进行系统审计。

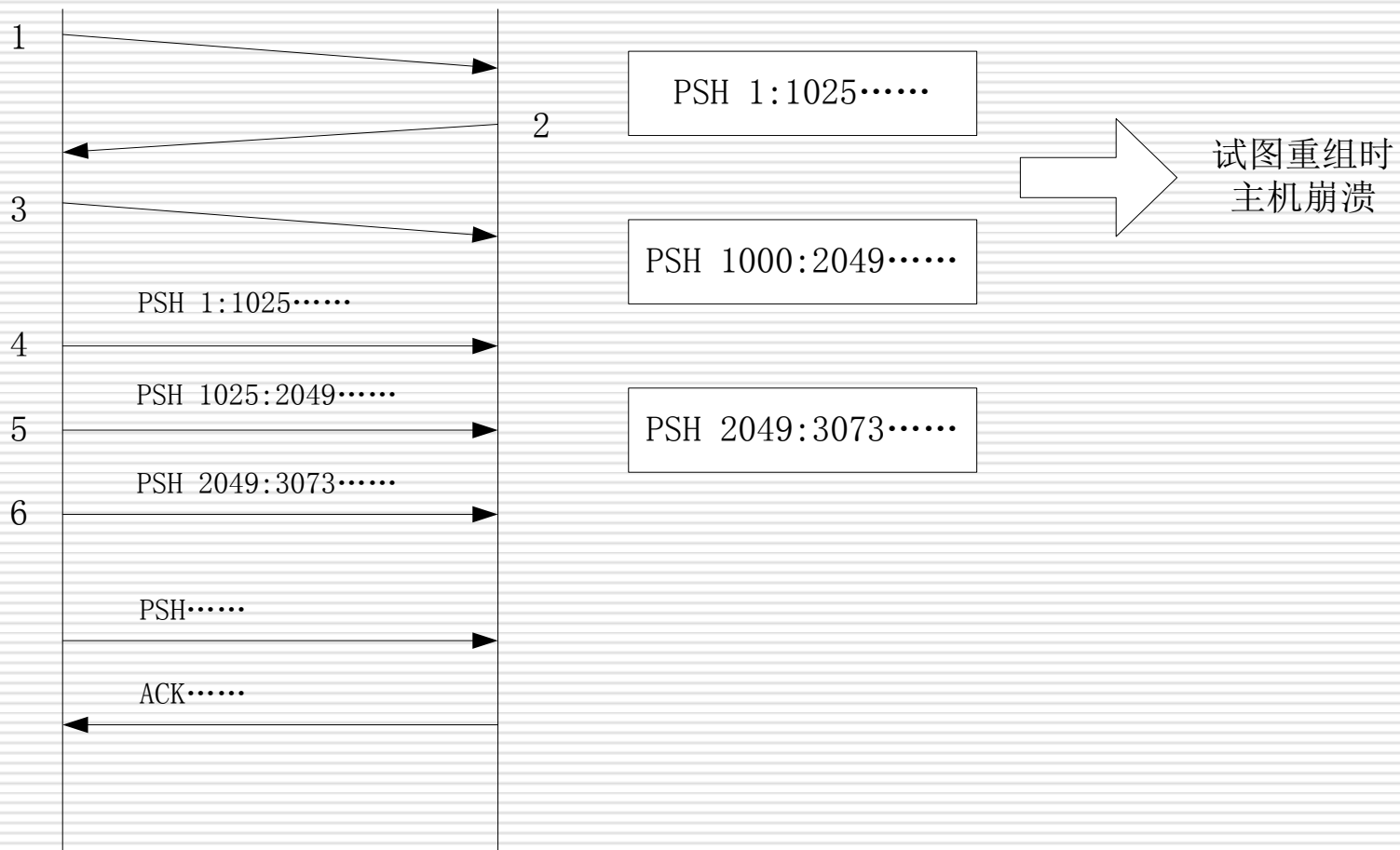
6.2.2 泪滴（Teardrop）

- “泪滴”也被称为分片攻击，它是一种典型的利用**TCP/IP**协议的问题进行拒绝服务攻击的方式，由于第一个实现这种攻击的程序名称为**Teardrop**，所以这种攻击也被称为“泪滴”。

6.2.2 泪滴（Teardrop）

- 两台计算机在进行通信时，如果传输的数据量较大，无法在一个数据报文中传输完成，就会将数据拆分成多个分片，传送到目的计算机后再到堆栈中进行重组，这一过程称为“分片”。
- 为了能在到达目标主机后进行数据重组，**IP**包的**TCP**首部中包含有信息（分片识别号、偏移量、数据长度、标志位）说明该分段是原数据的哪一段，这样，目标主机在收到数据后，就能根据首部中的信息将各分片重新组合还原为数据。

例子



例子(2)

- ❑ 如上图所示，从客户机向服务器发送一个数据报文无法发送完成的数据，这些数据会被分片发送。
- ❑ 报文**1**、**2**、**3**是**TCP**连接的三次握手过程，接着**4**、**5**、**6**客户机向服务器发送三个报文，在这三个数据报文首部信息中，有每个报文的分片信息。

例子(3)

□ 这就是报文重组的信息：

- PSH 1:1025(1024) ack 1, win 4096
- PSH 1025:2049(1024) ack 1, win 4096
- PSH 2049:3073(1024) ack 1, win 4096

□ 在这个报文中，可以看到在第**4**、**5**、**6**这三个报文中，第**4**个发送的数据报文中是原数据的第**1**~**1025**字节内容，第**5**个发送的报文包含的是第**1025**~**2048**字节，第**6**个数据报文是第**2049**~**3073**个字节，接着后面是继续发送的分片和服务器的确认。当这些分片数据被发送到目标主机后，目标主机就能够根据报文中的信息将分片重组，还原出数据。

例子(4)

- 如果入侵者伪造数据报文，向服务器发送含有重叠偏移信息的分段包到目标主机，例如如下所列的分片信息：
 - PSH 1:1025(1024) ack1, win4096
 - PSH 1000:2049(1024) ack1, win4096
 - PSH 2049:3073(1024) ack1, win4096
- 这样的信息被目的主机收到后，在堆栈中重组时，由于畸形分片的存在，会导致重组出错，这个错误并不仅仅是影响到重组的数据，由于协议重组算法，会导致内存错误，引起协议栈的崩溃。

6.2.2 泪滴（teardrop）

□ 泪滴攻击的攻击特征、检测方法和反攻击方法总结如下：

- **攻击特征**：Teardrop工作原理是向被攻击者发送多个分片的IP包，某些操作系统收到含有重叠偏移的伪造分片数据包时将会出现系统崩溃、重启等现象。
- **检测方法**：对接收到的分片数据包进行分析，计算数据包的片偏移量（**Offset**）是否有误。
- **反攻击方法**：添加系统补丁程序，丢弃收到的病态分片数据包并对这种攻击进行审计。

6.2.3 IP欺骗DoS攻击

- 这种攻击利用**RST**位来实现。
- 假设现在有一个合法用户(**61.61.61.61**)已经同服务器建立了正常的连接，攻击者构造攻击的**TCP**数据，伪装自己的**IP**为**61.61.61.61**，并向服务器发送一个带有**RST**位的**TCP**数据段。服务器接收到这样的数据后，认为**61.61.61.61**发送的连接有错误，就会清空缓冲区中建立好的连接。
- 这时，如果合法用户**61.61.61.61**再发送合法数据，服务器就已经没有这样的连接了，该用户就必须重新开始建立连接。

6.2.3 IP欺骗DoS攻击

- 攻击时，攻击者会伪造大量的**IP**地址，向目标发送**RST**数据，使服务器不对合法用户服务，从而实现了受害服务器的拒绝服务攻击。

6.2.4 UDP洪水

- ❑ **UDP洪水（UDP flood）** 主要是利用主机能自动进行回复的服务（例如使用**UDP**协议的**chargen**服务和**echo**服务）来进行攻击。
- ❑ 很多提供**WWW**和**Mail**等服务设备通常是使用**Unix**的服务器，它们默认打开一些被黑客恶意利用的**UDP**服务。如**echo**服务会显示接收到的每一个数据包，而原本作为测试功能的**chargen**服务会在收到每一个数据包时随机反馈一些字符。

6.2.4 UDP洪水

- 当我们向**echo**服务的端口发送一个数据时，**echo**服务会将同样的数据返回给发送方，而**chargen**服务则会随机返回字符。
- 当两个或两个以上系统存在这样的服务时，攻击者利用其中一台主机向另一台主机的**echo**或者**chargen**服务端口发送数据，**echo**和**chargen**服务会自动进行回复，这样开启**echo**和**chargen**服务的主机就会相互回复数据。
- 由于这种做法使一方的输出成为另一方的输入，两台主机间会形成大量的**UDP**数据包。当多个系统之间互相产生**UDP**数据包时，最终将导致整个网络瘫痪。

UDP洪水实例（UDP-Flood）

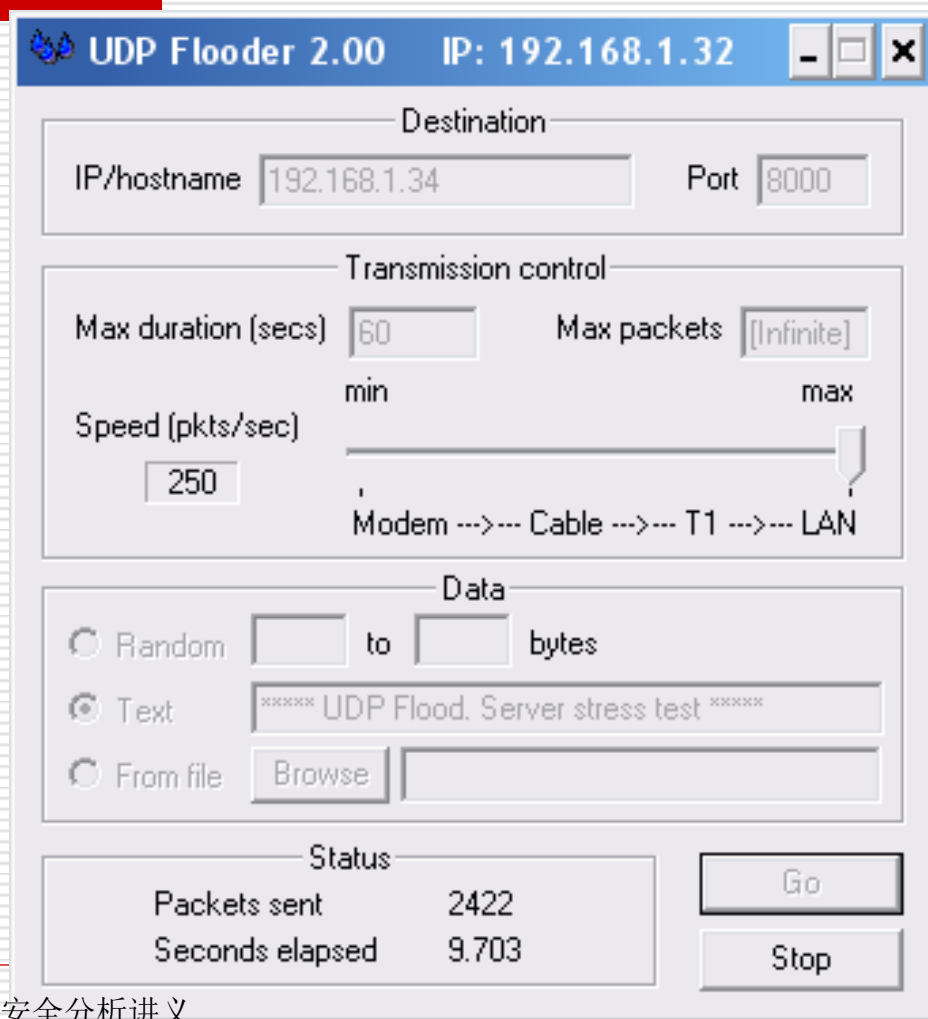
- ❑ **IP/hostname和port:** 输入目标主机的**IP**地址和端口号;
- ❑ **Max duration:** 设定最长的攻击时间;
- ❑ **Speed:** 设置**UDP**包发送速度;
- ❑ **Data:** 指定发送的**UDP**数据包中包含的内容。

The screenshot shows the 'UDP Flooder 2.00' application window. The title bar includes the application name and the local IP address 'IP: 192.168.1.32'. The interface is divided into several sections: 'Destination' with fields for 'IP/hostname' and 'Port'; 'Transmission control' with fields for 'Max duration (secs)', 'Max packets', and a 'Speed (pkts/sec)' slider set to '1' with a 'min' to 'max' scale; 'Data' with radio buttons for 'Random', 'Text', and 'From file'. The 'Text' option is selected, showing the content '***** UDP Flood. Server stress test *****'. The 'From file' option has a 'Browse' button. At the bottom, a 'Status' section shows 'Packets sent' and 'Seconds elapsed', both currently at '0'. There are 'Go' and 'Stop' buttons on the right side.

Status	
Packets sent	0
Seconds elapsed	0

UDP洪水实例(2)

- 对局域网网内的一台计算机
192.168.1.34
发起**UDP Flood**
攻击，发包速率为**250PPS**。



UDP洪水实例(3)

- ❑ 在被攻击的计算机**192.168.1.34**上打开**Sniffer**工具，可以捕捉由攻击者计算机发到本机的**UDP**数据包，可以看到内容为“******* UDP Flood. Server stress test *******”的大量**UDP**数据包，如下页图所示。
- ❑ 如果加大发包速率和增加攻击机的数量，则目标主机的处理能力将会明显下降。

(Untitled) - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: + Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
5249	61.601445	192.168.1.34	192.168.1.32	ICMP	Destination unreachable
5250	61.601456	192.168.1.32	192.168.1.34	UDP	Source port: 3972 Destination port: 8000
5251	61.601463	192.168.1.34	192.168.1.32	ICMP	Destination unreachable
5252	61.601468	192.168.1.32	192.168.1.34	UDP	Source port: 3972 Destination port: 8000
5253	61.601476	192.168.1.34	192.168.1.32	ICMP	Destination unreachable
5254	61.616987	192.168.1.32	192.168.1.34	UDP	Source port: 3972 Destination port: 8000
5255	61.617034	192.168.1.34	192.168.1.32	ICMP	Destination unreachable
5256	61.617048	192.168.1.32	192.168.1.34	UDP	Source port: 3972 Destination port: 8000
5257	61.617055	192.168.1.34	192.168.1.32	ICMP	Destination unreachable
5258	61.617061	192.168.1.32	192.168.1.34	UDP	Source port: 3972 Destination port: 8000
5259	61.617068	192.168.1.34	192.168.1.32	ICMP	Destination unreachable
5260	61.617074	192.168.1.32	192.168.1.34	UDP	Source port: 3972 Destination port: 8000

Frame 5258 (83 bytes on wire, 83 bytes captured)

Ethernet II, Src: 00:e0:4c:ed:36:0e, Dst: 00:01:6c:a2:18:a2

Internet Protocol, Src Addr: 192.168.1.32 (192.168.1.32), Dst Addr: 192.168.1.34 (192.168.1.34)

User Datagram Protocol, Src Port: 3972 (3972), Dst Port: 8000 (8000)

Data (41 bytes)

```

0000  00 01 6c a2 18 a2 00 e0 4c ed 36 0e 08 00 45 00  ....L.b...E.
0010  00 45 15 b4 00 00 80 11 a1 61 c0 a8 01 20 c0 a8  .E.....a...
0020  01 22 0f 84 1f 40 00 31 f9 5c 2a 2a 2a 2a 2a 20  ."...@.1.\*****
0030  55 44 50 20 46 6c 6f 6f 64 2e 20 53 65 72 76 65  UDP Flood. Serve
0040  72 20 73 74 72 65 73 73 20 74 65 73 74 20 2a 2a  r stress test **
0050  2a 2a 00                                     **.
  
```

File: (Untitled) 1156 KB 00:01:16 Drops: 0 P: 12803 D: 12803 M: 0

6.2.5 SYN洪水

- ❑ **SYN Flood**是当前最流行的拒绝服务攻击方式之一，这是一种利用**TCP**协议缺陷，发送大量伪造的**TCP**连接请求，使被攻击方资源耗尽(**CPU**满负荷或内存不足)的攻击方式。
- ❑ **SYN Flood**是利用**TCP**连接的三次握手过程的特性实现的。

6.2.5 SYN洪水

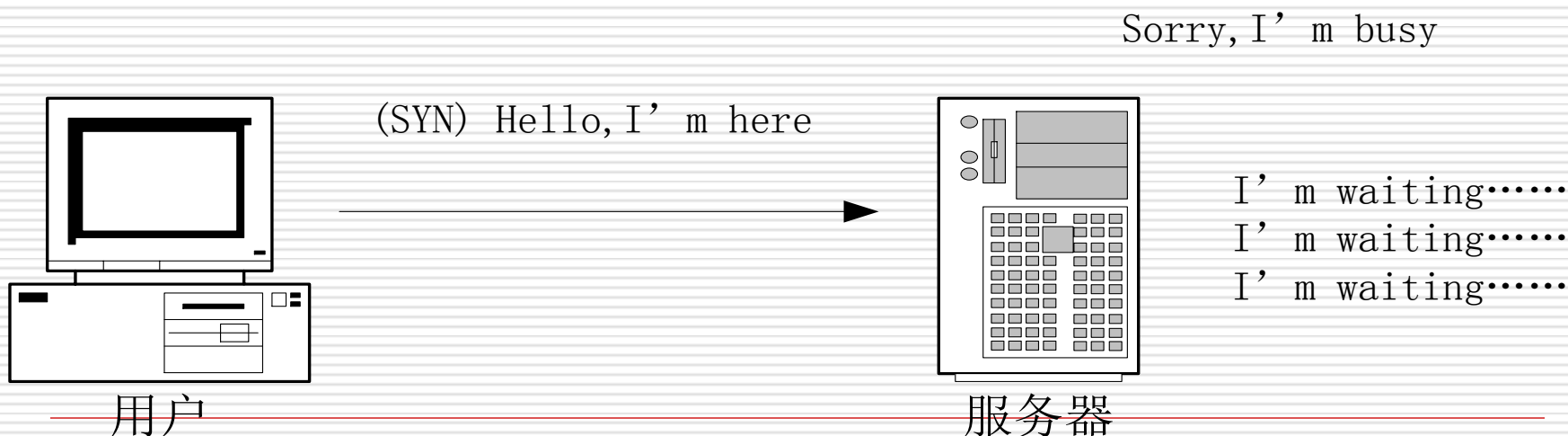
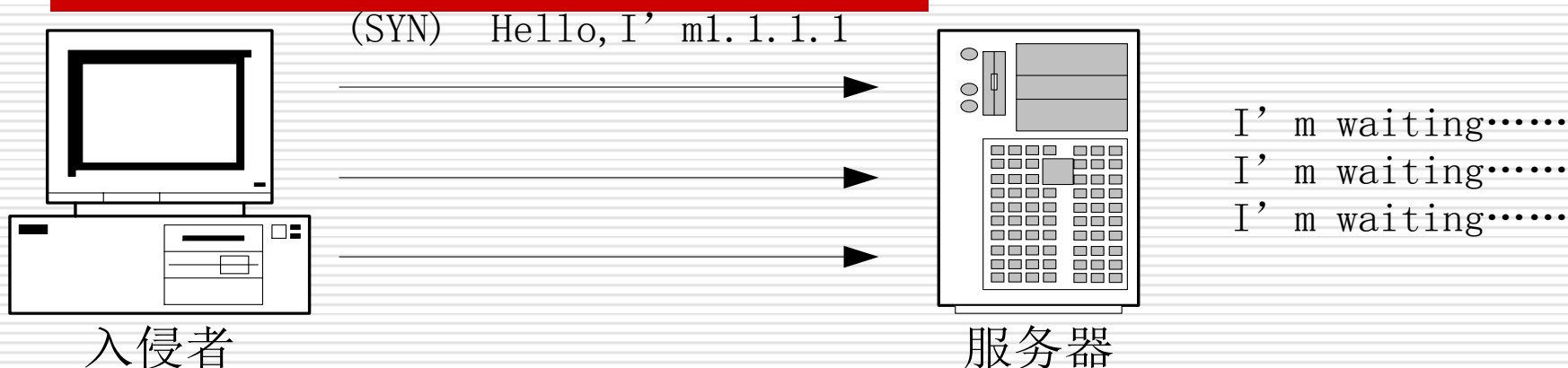
- 在**TCP**连接的三次握手过程中，假设一个客户端向服务器发送了**SYN**报文后突然死机或掉线，那么服务器在发出**SYN/ACK**应答报文后是无法收到客户端的**ACK**报文的，这种情况下服务器端一般会重试，并等待一段时间后丢弃这个未完成的连接。这段时间的长度我们称为**SYN Timeout**。一般来说这个时间是分钟的数量级。
- 一个用户出现异常导致服务器的一个线程等待**1**分钟并不是什么很大的问题，但如果有一个恶意的攻击者大量模拟这种情况(伪造**IP**地址)，服务器端将为了维护一个非常大的半连接列表而消耗非常多的资源。

6.2.5 SYN洪水

- ❑ 即使是简单的保存并遍历半连接列表也会消耗非常多的**CPU**时间和内存，何况还要不断对这个列表中的**IP**进行**SYN+ACK**的重试。
- ❑ 实际上如果服务器的**TCP/IP**栈不够强大，最后的结果往往是堆栈溢出崩溃——即使服务器端的系统足够强大，服务器端也将忙于处理攻击者伪造的**TCP**连接请求而无暇理睬客户的正常请求，此时从正常客户的角度来看，服务器失去响应，这种情况就称作：服务器端受到了**SYN Flood**攻击(**SYN洪水攻击**)。

SYN“洪水”攻击示意图

I' m ready, I' m
waiting for you

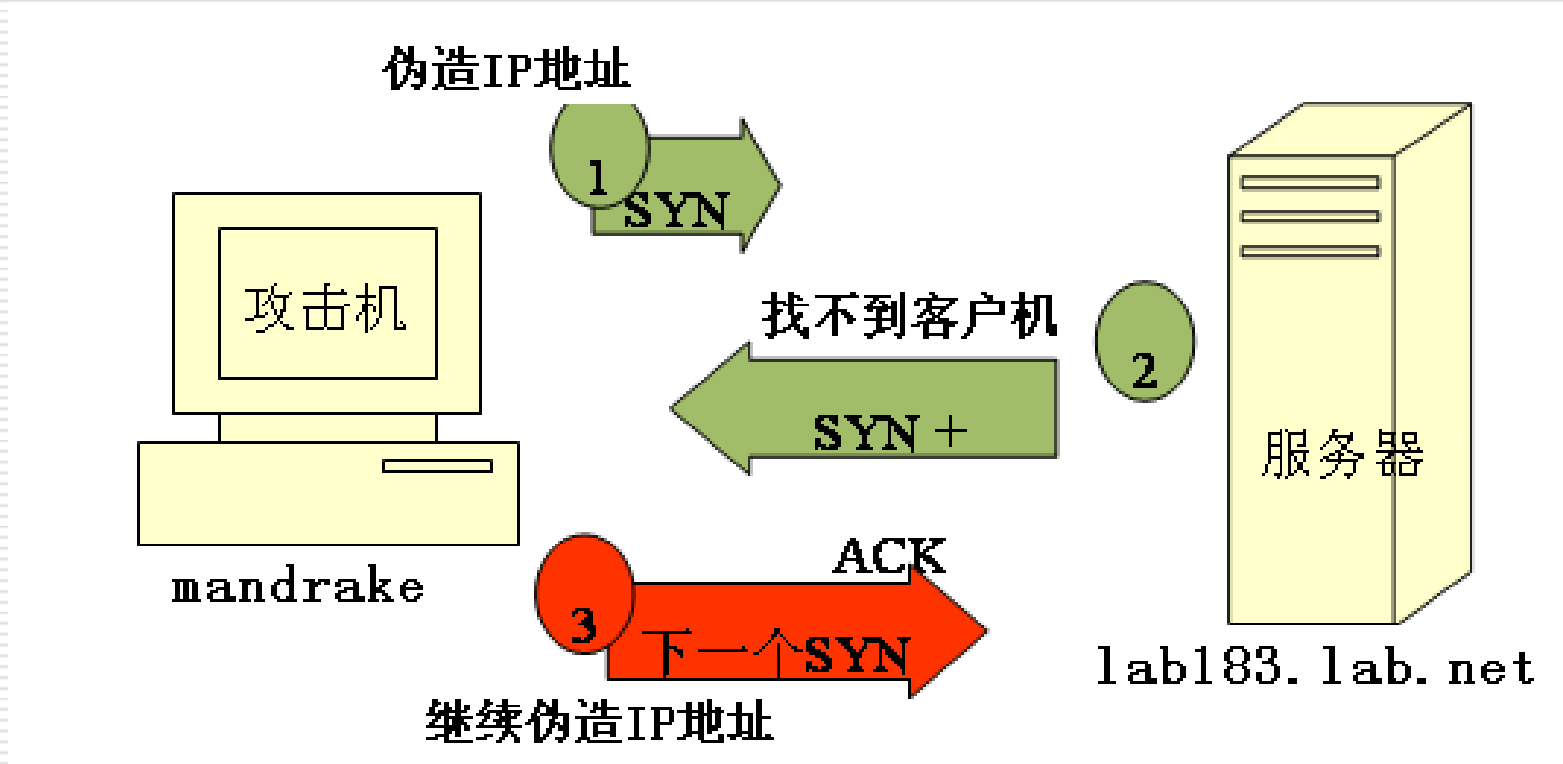


SYN“洪水”攻击实例

- 局域网环境，有一台攻击机
（**PIII667/128/mandrake**），被攻击的是一台**Solaris 8.0**的主机，网络设备是**Cisco**的百兆交换机。
- 后面将显示在**Solaris**上进行**snoop**抓包的记录。
- 注：**snoop**与**tcpdump**等网络监听工具一样，是一个网络抓包与分析工具。

SYN“洪水”攻击实例(2)

攻击示意图:



SYN“洪水”攻击实例(3)

- ❑ 攻击机开始发包，**DoS**开始了...，突然间**Solaris**主机上的**snoop**窗口开始飞速地翻页，显示出接到数量巨大的**Syn**请求。这时的屏幕就好象是时速**300**公里的列车上的一扇车窗。
- ❑ **Syn Flood**攻击时的**snoop**输出结果如下页图所示。

1	0.000000	16.198.89.0	192.168.6.13	TCP	38594 > 59983	[SYN, URG]	Seq=0 Ack=0 win=26220
2	0.000019	169.103.0.0	192.168.6.13	TCP	10330 > 34272	[SYN, URG]	Seq=0 Ack=0 win=36414
3	0.000032	91.169.213.0	192.168.6.13	TCP	27322 > 21290	[SYN, URG]	Seq=0 Ack=0 win=4488,
4	0.000044	184.114.18.0	192.168.6.13	TCP	50727 > 21119	[SYN, URG]	Seq=0 Ack=0 win=21852
5	0.000057	207.18.89.0	192.168.6.13	TCP	29015 > 57863	[SYN, URG]	Seq=0 Ack=0 win=52368
6	0.000070	84.178.6.0	192.168.6.13	TCP	34308 > 19117	[SYN, URG]	Seq=0 Ack=0 win=39343
7	0.000279	165.176.212.0	192.168.6.13	TCP	10361 > 21828	[SYN, URG]	Seq=0 Ack=0 win=9215,
8	0.000523	34.5.85.0	192.168.6.13	TCP	34292 > 1026	[SYN, URG]	Seq=0 Ack=0 win=29890,
9	0.000540	185.89.119.0	192.168.6.13	TCP	55906 > 37830	[SYN, URG]	Seq=0 Ack=0 win=3555,
10	0.000553	36.91.145.0	192.168.6.13	TCP	5322 > 1535	[SYN, URG]	Seq=0 Ack=0 win=13664,
11	0.005188	198.80.143.0	192.168.6.13	TCP	54383 > 3147	[SYN, URG]	Seq=0 Ack=0 win=28765,
12	0.005235	183.193.69.0	192.168.6.13	TCP	25358 > 37470	[SYN, URG]	Seq=0 Ack=0 win=16868
13	0.005250	67.137.225.0	192.168.6.13	TCP	10978 > 6775	[SYN, URG]	Seq=0 Ack=0 win=55038,
14	0.005263	79.147.85.0	192.168.6.13	TCP	19356 > 40745	[SYN, URG]	Seq=0 Ack=0 win=62824
15	0.005780	67.8.4.0	192.168.6.13	TCP	64388 > 41889	[SYN, URG]	Seq=0 Ack=0 win=31778
16	0.005806	46.13.153.0	192.168.6.13	TCP	33607 > 7096	[SYN, URG]	Seq=0 Ack=0 win=37195,
17	0.005851	12.116.83.0	192.168.6.13	TCP	50443 > 62841	[SYN, URG]	Seq=0 Ack=0 win=42192
18	0.005863	239.73.237.0	192.168.6.13	TCP	39099 > 16940	[SYN, URG]	Seq=0 Ack=0 win=125,
19	0.005876	97.50.47.0	192.168.6.13	TCP	58840 > 4928	[SYN, URG]	Seq=0 Ack=0 win=25909,
20	0.005938	31.170.243.0	192.168.6.13	TCP	2834 > 44656	[SYN, URG]	Seq=0 Ack=0 win=14553,
21	0.005953	98.164.203.0	192.168.6.13	TCP	29192 > 15241	[SYN, URG]	Seq=0 Ack=0 win=6508,
22	0.005965	38.20.211.0	192.168.6.13	TCP	24047 > 36982	[SYN, URG]	Seq=0 Ack=0 win=47132
23	0.005977	176.185.161.0	192.168.6.13	TCP	52465 > 17444	[SYN, URG]	Seq=0 Ack=0 win=1478,
24	0.005991	10.95.136.0	192.168.6.13	TCP	3860 > 46681	[SYN, URG]	Seq=0 Ack=0 win=5268,
25	0.006004	215.66.6.0	192.168.6.13	TCP	34969 > 32543	[SYN, URG]	Seq=0 Ack=0 win=42379
26	0.006048	241.79.143.0	192.168.6.13	TCP	57553 > 7634	[SYN, URG]	Seq=0 Ack=0 win=31490,
27	0.006064	213.191.131.0	192.168.6.13	TCP	29956 > 1500	[SYN, URG]	Seq=0 Ack=0 win=16678,
28	0.006076	54.246.235.0	192.168.6.13	TCP	22637 > 20564	[SYN, URG]	Seq=0 Ack=0 win=60443
29	0.006088	165.237.132.0	192.168.6.13	TCP	29941 > 44743	[SYN, URG]	Seq=0 Ack=0 win=34505
30	0.006101	117.17.59.0	192.168.6.13	TCP	49069 > 21347	[SYN, URG]	Seq=0 Ack=0 win=8876,
31	0.006113	0.255.108.0	192.168.6.13	TCP	32982 > 20743	[SYN, URG]	Seq=0 Ack=0 win=20633
32	0.006123	189.57.227.0	192.168.6.13	TCP	21567 > 4089	[SYN, URG]	Seq=0 Ack=0 win=52334,

Sniffer发现针对某一主机的大量的TCP SYN报文

SYN“洪水”攻击实例(4)

- ❑ 此时，目标主机再也收不到刚才那些正常的网络包，只有**DoS**包。
- ❑ 大家注意一下，这里所有的**Syn Flood**攻击包的源地址都是伪造的，给追查工作带来很大困难。
- ❑ 这时在被攻击主机上积累了多少**Syn**的半连接呢？用**netstat**来看一下：
netstat -an | grep SYN。
结果如下页图所示。

...

...

192.168.0.183.9	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.13	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.19	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.21	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.22	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.23	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.25	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.37	127.0.0.79.1801	0	0 24656	0 SYN_RCVD
192.168.0.183.53	127.0.0.79.1801	0	0 24656	0 SYN_RCVD.....

SYN“洪水”攻击实例(5)

其中**SYN_RCVD**表示当前未完成的**TCP SYN**队列，统计一下（**wc**是文件内容统计命令，**-l**选项表示统计行数）：

```
# netstat -an | grep SYN | wc -l  
5273
```

```
# netstat -an | grep SYN | wc -l  
5154
```

```
# netstat -an | grep SYN | wc -l  
5267
```

.....

共有五千多个**Syn**的半连接存储在内存中。这时候被攻击机已经不能响应新的服务请求了，系统运行非常慢，也无法**ping**通。而这只是在攻击发起后仅仅**70**秒钟左右时的情况。

SYN“洪水”的防御

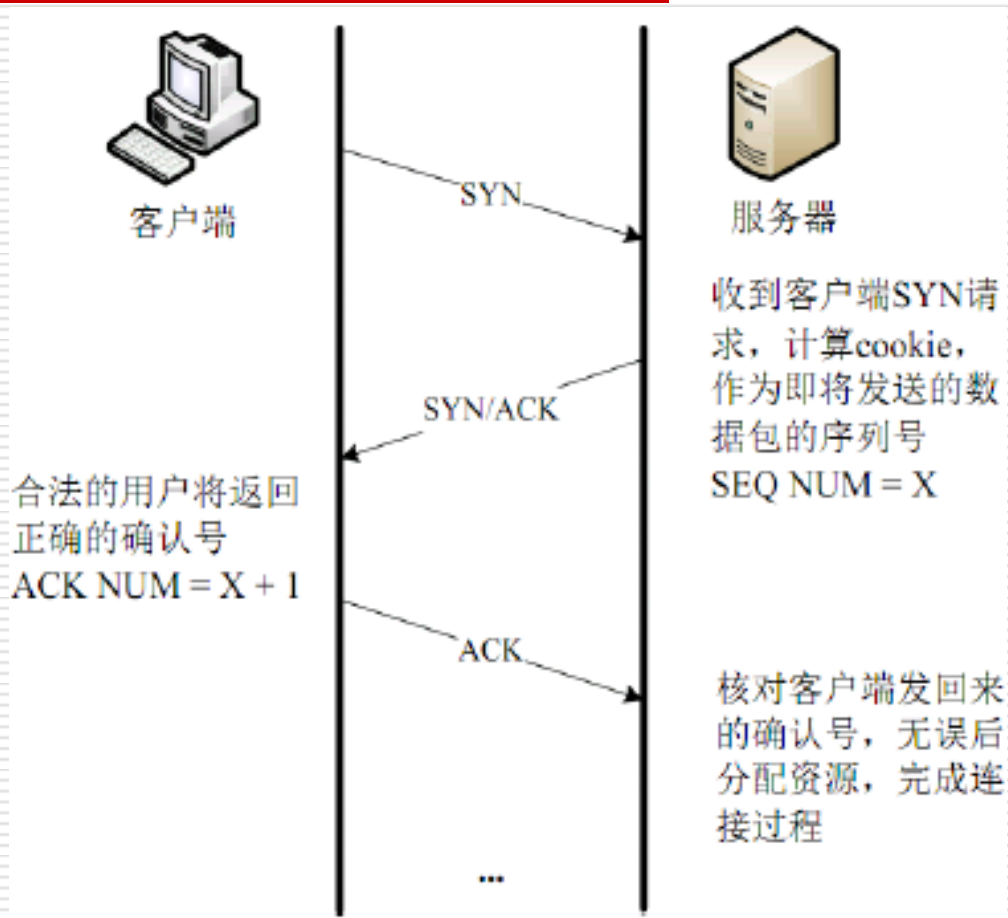
□ **SYN洪水攻击**比较难以防御，以下是几种解决方法：

- 缩短SYN Timeout时间
- 设置SYN Cookie
- 负反馈策略
- 退让策略
- 分布式DNS负载均衡
- 防火墙

缩短**SYN Timeout**时间

- 由于**SYN Flood**攻击的效果取决于服务器上保持的**SYN**半连接数，这个值=**SYN**攻击的频度 \times **SYN Timeout**，所以通过缩短从接收到**SYN**报文到确定这个报文无效并丢弃该连接的时间，可以成倍的降低服务器的负荷。

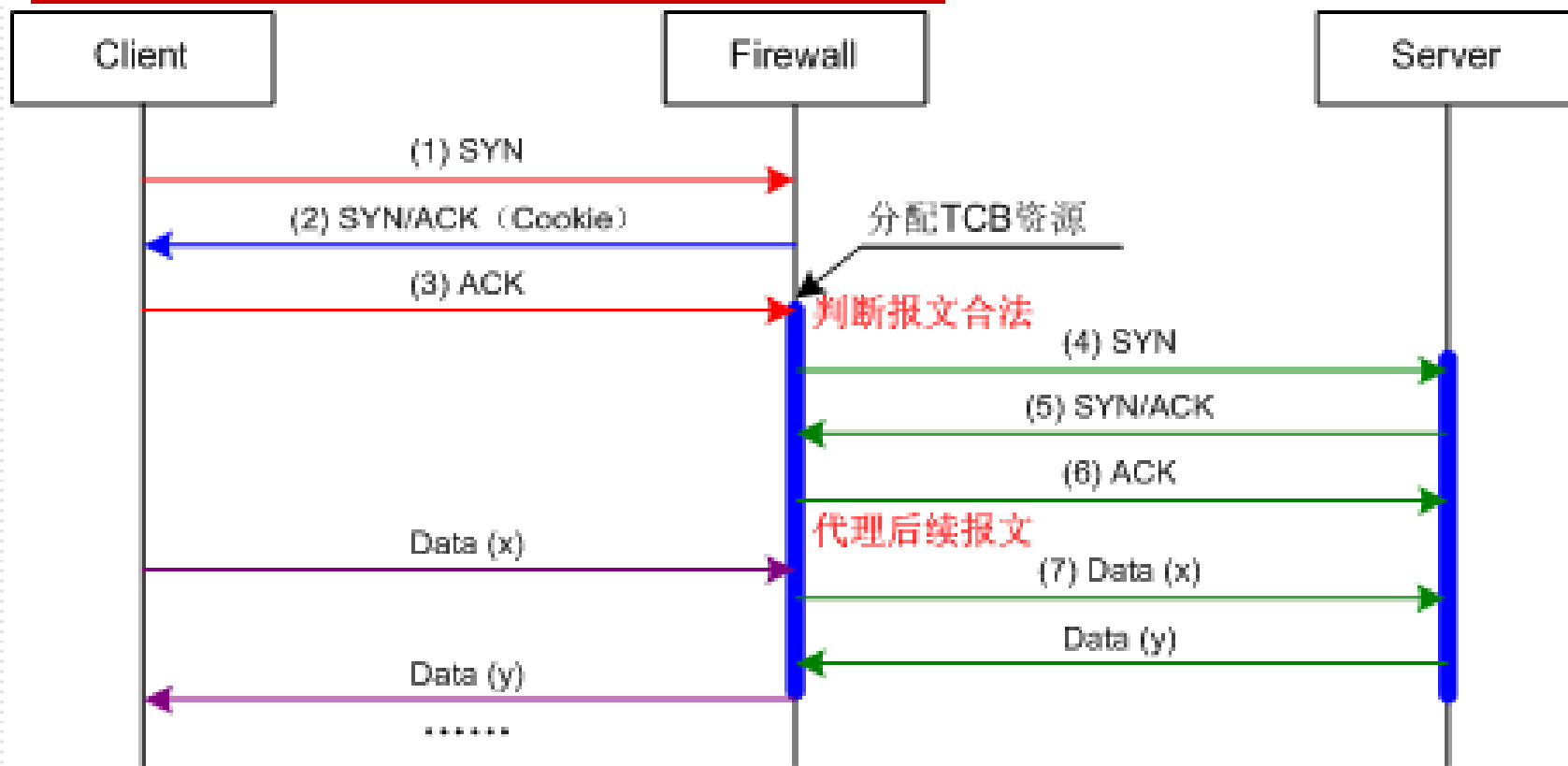
设置SYN Cookie



设置SYN Cookie

- ❑ 当服务器**S**接收到一个**SYN**包时，返回**SYN/ACK**包，其**ACK**序列号是经过加密的，由源地址，源端口，目标地址，目标端口和一个加密种子计算得出。然后服务器释放所有的状态。
- ❑ 如果一个**ACK**包从客户端**C**返回，服务器将重新计算来判断它是不是上个**SYN/ACK**的返回包。如果是，**S**就可以直接进入**TCP**连接状态并打开连接，否则直接丢弃。这样，**S**就可以避免守候半开放连接了。

设置SYN Cookie



防火墙使用SYN Cookie防范SYN Flood

设置SYN Cookie

- ❑ 防火墙的**SYN Cookie**技术利用**SYN/ACK**报文携带的认证信息，对握手协商的**ACK**报文进行了认证，从而避免了防火墙过早分配**TCB**资源，可以有效防范**SYN Flood**攻击。
- ❑ 在防范**SYN Flood**攻击的过程中，防火墙作为虚拟的服务器与客户端交互，同时也作为虚拟的客户端与服务器交互，在为服务器过滤掉恶意连接报文的同时保证了常规业务的正常运行。

负反馈策略

- 正常情况下，**OS**对**TCP**连接的一些重要参数有一个常规的设置：**SYN Timeout**时间、**SYN-ACK**的重试次数、**SYN**报文从路由器到系统再到**Winsock**的延时等等。
- 这个常规设置针对系统优化，可以给用户提供服务；一旦服务器受到攻击，**SYN Half link** 的数量超过系统中**TCP**活动**Half link**最大连接数的设置，系统将会认为自己受到了**SYN Flood**攻击，并将根据攻击的判断情况作出反应：减短**SYN Timeout**时间、减少**SYN-ACK**的重试次数、自动对缓冲区中的报文进行延时等等措施，力图将攻击危害减到最低。

退让策略

- ❑ 退让策略是基于**SYN Flood**攻击代码的一个缺陷：**SYN Flood**一旦攻击开始，将不会再进行域名解析。
- ❑ 切入点：假设一台服务器在受到**SYN Flood**攻击后迅速更换自己的**IP**地址，那么攻击者仍在不断攻击的只是一个空的**IP**地址，并没有任何主机，而防御方只要将**DNS**解析更改到新的**IP**地址就能在很短的时间内恢复用户通过域名进行的正常访问。
- ❑ 为了迷惑攻击者，我们甚至可以放置一台“牺牲”服务器让攻击者满足于攻击的“效果”。

分布式**DNS**负载均衡

- ❑ 在众多的负载均衡架构中，基于**DNS**解析的负载均衡本身就拥有对**SYN Flood**的免疫力。
- ❑ 基于**DNS**解析的负载均衡能将用户的请求分配到不同**IP**的服务器主机上，攻击者攻击的永远只是其中一台服务器，一来这样增加了攻击者的成本，二来过多的**DNS**请求可以帮助我们追查攻击者的真正踪迹。

6.2.6 Land攻击

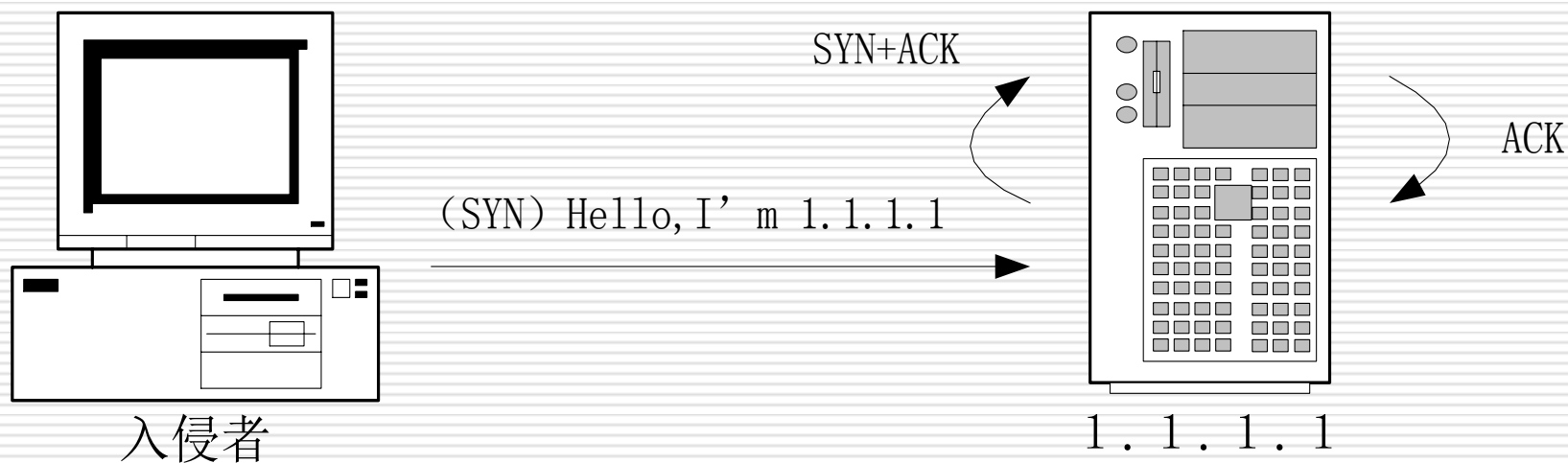
- ❑ **Land**是因特网上最常见的拒绝服务攻击类型，它是由著名黑客组织**rootshell**发现的。
- ❑ 原理很简单，向目标机发送大量的源地址和目标地址相同的包，造成目标机解析**Land**包时占用大量的系统资源，从而使网络功能完全瘫痪。

6.2.6 Land攻击

- ❑ **Land**攻击也是利用**TCP**的三次握手过程的缺陷进行攻击。
- ❑ **Land**攻击是向目标主机发送一个特殊的**SYN**包，包中的源地址和目标地址都是目标主机的地址。目标主机收到这样的连接请求时会向自己发送**SYN/ACK**数据包，结果导致目标主机向自己发回**ACK**数据包并创建一个连接。
- ❑ 大量的这样的数据包将使目标主机建立很多无效的连接，系统资源被大量的占用。

6.2.6 Land攻击

□ Land攻击示意图:



6.2.6 Land攻击

□ Land攻击可简要概括如下：

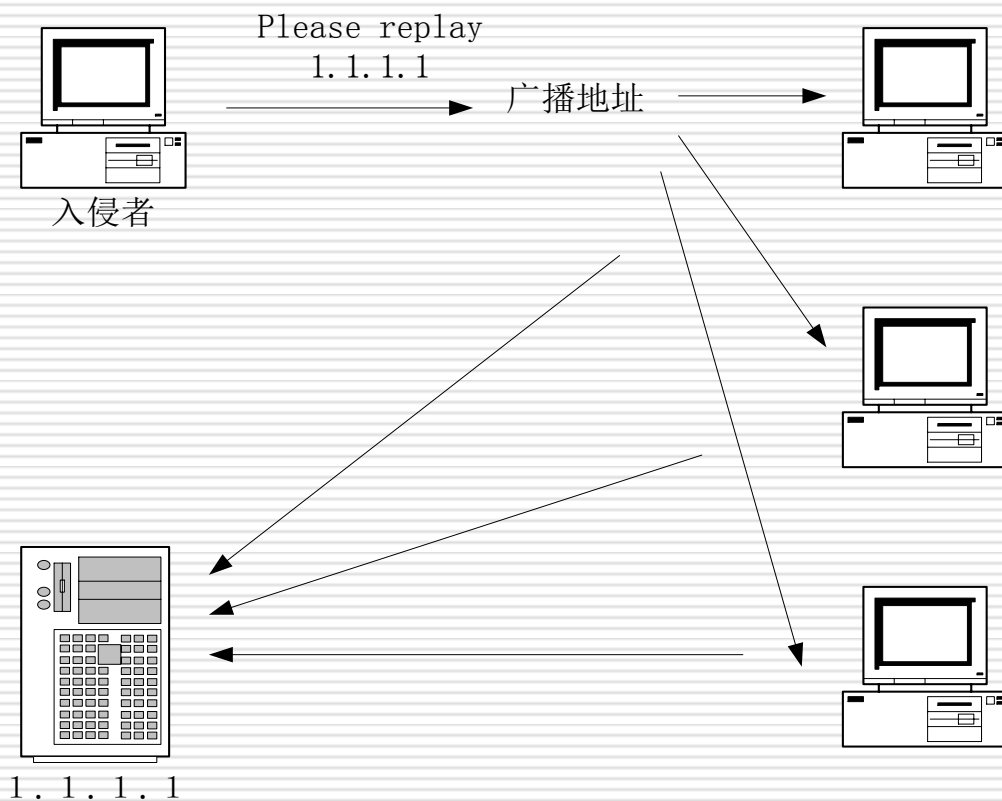
- **攻击特征：**用于Land攻击的数据包中的源地址和目标地址是相同的。操作系统接收到这类数据包时，不知道该如何处理堆栈中的这种情况，或者循环发送和接收该数据包，消耗大量的系统资源，从而有可能造成系统崩溃或死机等现象。
- **检测方法：**判断网络数据包的源/目标地址是否相同。
- **反攻击方法：**适当配置防火墙设备或过滤路由器的过滤规则可以防止这种攻击行为，并对这种攻击进行审计。

6.2.7 Smurf攻击

- ❑ **Smurf**攻击是利用**IP**欺骗和**ICMP**回应包引起目标主机网络阻塞，实现**DoS**攻击。
- ❑ **Smurf**攻击原理：在构造数据包时将源地址设置为被攻击主机的地址，而将目的地址设置为广播地址，于是，大量的**ICMP echo**回应包被发送给被攻击主机，使其因网络阻塞而无法提供服务。
- ❑ 比**Ping of Death**洪水的流量高出**1**或**2**个数量级。

6.2.7 Smurf攻击

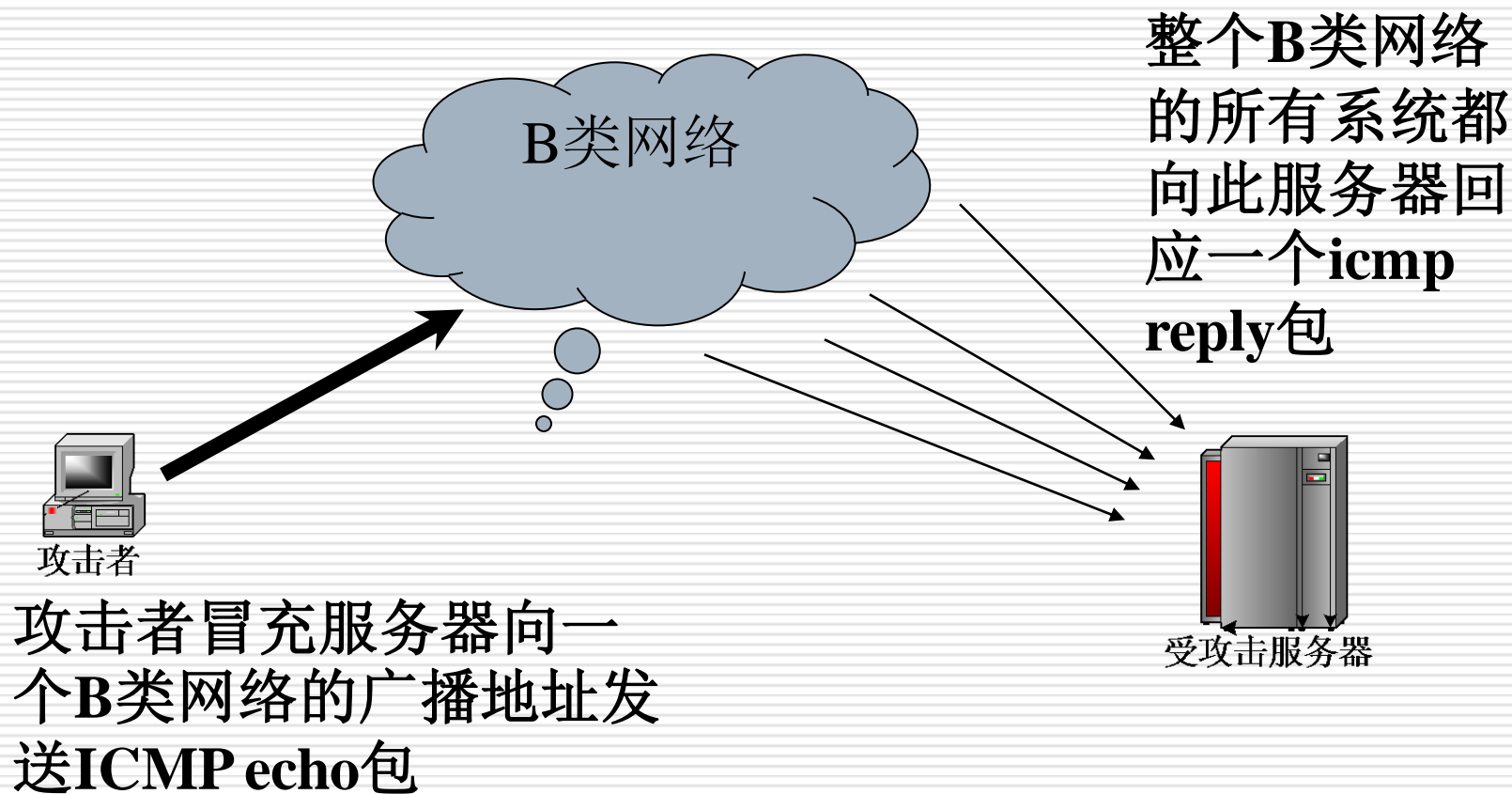
□ Smurf攻击示意图:



6.2.7 Smurf攻击

- ❑ 如上例所示，入侵者的主机发送了一个数据包，而目标主机就收到了三个回复数据包。
- ❑ 如果目标网络是一个很大的以太网，有**200**台主机，那么在这种情况下，入侵者每发送一个**ICMP**数据包，目标主机就会收到**200**个数据包，因此目标主机很快就会被大量的回复信息吞没，无法处理其他的任何网络传输。
- ❑ 这种攻击不仅影响目标主机，还能影响目标主机的整个网络系统。

Smurf攻击例子



6.2.8 Fraggle攻击

- **Fraggle**攻击原理与**Smurf**一样，也是采用向广播地址发送数据包，利用广播地址的特性将攻击放大以使目标主机拒绝服务。
- 不同的是，**Fraggle**使用的是**UDP**应答消息而非**ICMP**。

6.2.9 电子邮件炸弹

- ❑ 电子邮件炸弹是最古老的匿名攻击之一，由于这种攻击方式简单易用，互联网上也很容易找到这些发送匿名邮件的工具，并且入侵者只需要知道对方的电子邮件地址就可以进行攻击了。
- ❑ 传统的电子邮件炸弹只是简单的往你的邮箱里发送大量的邮件，入侵者的目的是要用垃圾邮件填满你的邮箱后，正常的邮件就会因空间不够而被服务器拒收。

6.2.9 电子邮件炸弹

- ❑ 如果用户的邮箱使用空间不受限制，那么电子邮件炸弹攻击就有可能影响到服务器的正常工作了。
- ❑ 最有可能的情况是入侵者不断发送大量的电子邮件，由于用户的邮箱空间不受限制，服务器会接收全部的邮件并保存在硬盘上。大量到来的邮件将不断吞噬服务器上的硬盘空间，最终将耗尽服务器上的所有硬盘空间，使得服务器无法再对外服务。
- ❑ 还有一种可能是通过设置一台机器不断地大量向同一地址发送电子邮件，入侵者能够耗尽接收者网络的带宽。

6.2.9 电子邮件炸弹

- ❑ 电子邮件是通过**SMTP**协议进行发送的，最初的**SMTP**协议服务是不需要进行身份认证的，在发送电子邮件的过程中不对用户进行身份认证。
- ❑ **SMTP**不会进行认证，邮件的发送人可以伪造任何邮件地址，甚至可以不写发件人的信息。这就是能发送匿名邮件的原因。
- ❑ 针对**SMTP**的问题，新的**SMTP**协议规范新增了**2**个命令，对发送邮件的发件人进行身份认证，在一定程度上降低了匿名电子邮件的风险。

6.2.10 畸形消息攻击

- ❑ 畸形消息攻击是一种有针对性的攻击方式，它利用目标主机或者特定服务存在的安全漏洞进行攻击。
- ❑ 目前无论是**Windows**、**Unix**、**Linux**等各类操作系统上的许多服务都存在安全漏洞，由于这些服务在处理信息之前没有进行适当的错误校验，所以一旦收到畸形的信息就有可能崩溃。

6.2.10 畸形消息攻击

- 例如，在**IIS 5**没有安装相应的修补包以及没有相应的安全措施时，向**IIS 5**服务器递交如下的**URL**会导致**IIS 5**停止服务：

http://testIP/...[25kb of '.']...ida

而向**IIS 5**递交如下的**HTTP**请求会导致**IIS**系统的崩溃，需要重新启动才能恢复：

“GET /.....[3k]..... .htr HTTP/1.0”

- 这两者都是向服务器提交正常情况下不会出现请求，导致服务器处理错误而崩溃，是典型的畸形消息攻击。

6.2.11 Slashdot effect

- ❑ **Slashdot effect**来自Slashdot.org这个网站，这曾是十分知名而且浏览人数十分庞大的**IT**、电子、娱乐网站，也是**blog**网站的开宗始祖之一。由于**Slashdot.org**的知名度和浏览人数的影响，在**Slashdot.org**上的文章中放入的网站链接，有可能一瞬间被点入上千次，甚至上万次，造成这个被链接的网站承受不住突然增加的连接请求，出现响应变慢、崩溃、拒绝服务。这种现象就称为**Slashdot effect**，这种瞬间产生的大量进入某网站的动作，也称作**Slashdotting**。

6.2.11 Slashdot effect

- ❑ 这种攻击手法使**web**服务器或其他类型的服务器由于大量的网络传输而过载，一般这些网络流量是针对某一个页面或一个链接而产生的。
- ❑ 当然这种现象也会在访问量较大的网站上正常的发生，但一定要把这些正常现象和攻击区分开来。
- ❑ 如果您的服务器突然变得拥挤不堪，甚至无法响应再多的请求时，您应当仔细检查一下这个资源匮乏的现象，确认在**10000**次点击里全都是合法用户进行的，还是由**5000**个合法用户和一个点击了**5000**次的攻击者进行的。

6.2.12 WinNuke攻击

- ❑ **WinNuke**攻击又称“带外传输攻击”，它的特征是攻击目标端口，被攻击的目标端口通常是**139、138、137、113、53**。
- ❑ **TCP**传输协议中使用带外数据（**Out of Band, OOB**数据）通道来传送一些比较特殊（如比较紧急）的数据。在紧急模式下，发送的每个**TCP**数据包都包含**URG**标志和**16**位**URG**指针，直至将要发送的带外数据发送完为止。**16**位**URG**指针指向包内数据段的某个字节数据，表示从第一字节到指针所指字节的数据就是紧急数据，不进入接收缓冲就直接交给上层进程。

6.2.12 WinNuke攻击

- **WinNuke**攻击就是制造特殊的这种报文，但这些攻击报文与正常携带**OOB**数据报文不同的是：其指针字段与数据的实际位置不符，即存在重合，这样**WINDOWS**操作系统在处理这些数据的时候，就会崩溃。

6.2.12 WinNuke攻击

- 攻击者将这样的特殊**TCP**带外数据报文发送给已建立连接的主机的**NetBIOS**端口**139**，导致主机崩溃后，会显示下面的信息：

An exception OE has occurred at 0028:[address] in VxD MSTCP(01)+ 000041AE. This was called from 0028:[address] in VxD NDIS(01)+ 00008660.It may be possible to continue normally.

Press any key to attempt to continue.

Press CTRL+ALT+DEL to restart your computer.

You will lose any unsaved information in all applications.

Press any key to continue

6.2.12 WinNuke攻击

□ WinNuke攻击的特征、检测方法和反攻击方法概括如下：

- **攻击特征：**WinNuke攻击又称带外传输攻击，它的特征是被攻击的目标端口通常是139、138、137、113、53，而且URG位设为“1”，即紧急模式。
- **检测方法：**判断数据包目标端口是否为139、138、137等，并判断URG位是否为“1”。
- **反攻击方法：**适当配置防火墙设备或过滤路由器就可以防止这种攻击手段（丢弃该数据包），并对这种攻击进行审计（记录事件发生的时间，源主机和目标主机的MAC地址和IP地址）

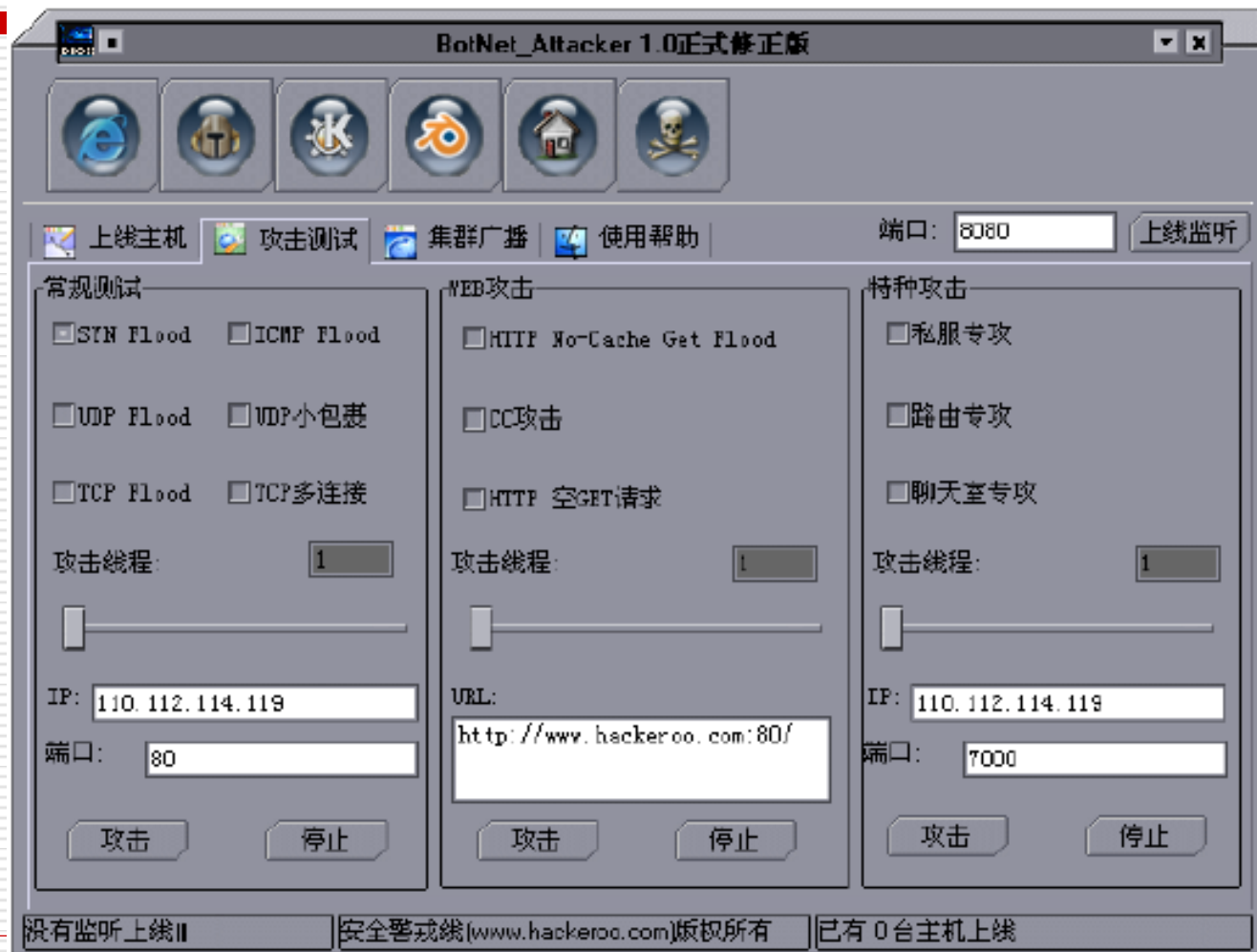
Dos常用攻击工具

独裁者



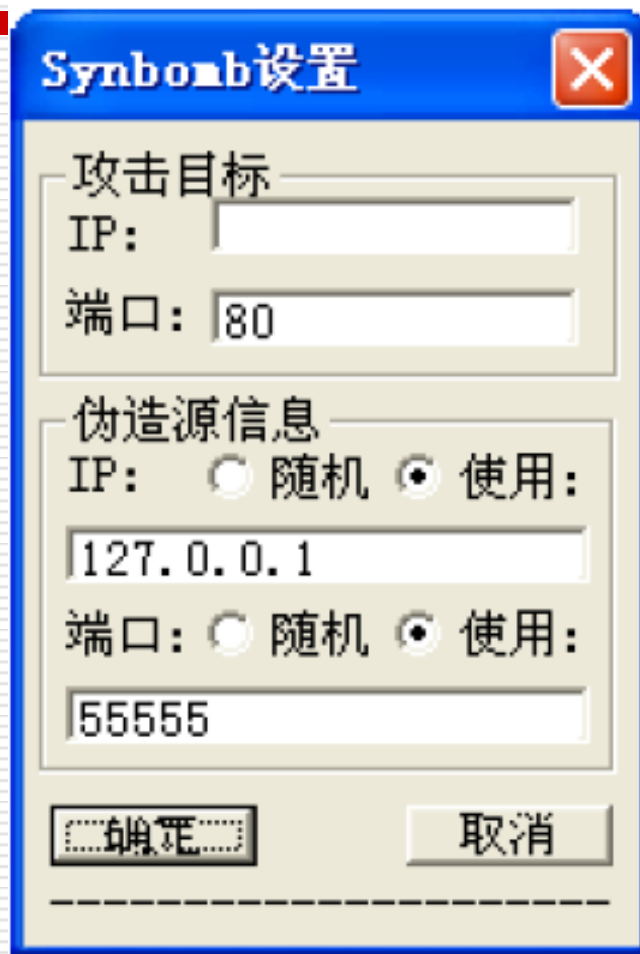
Dos常用攻击工具

NetBot_Attacker



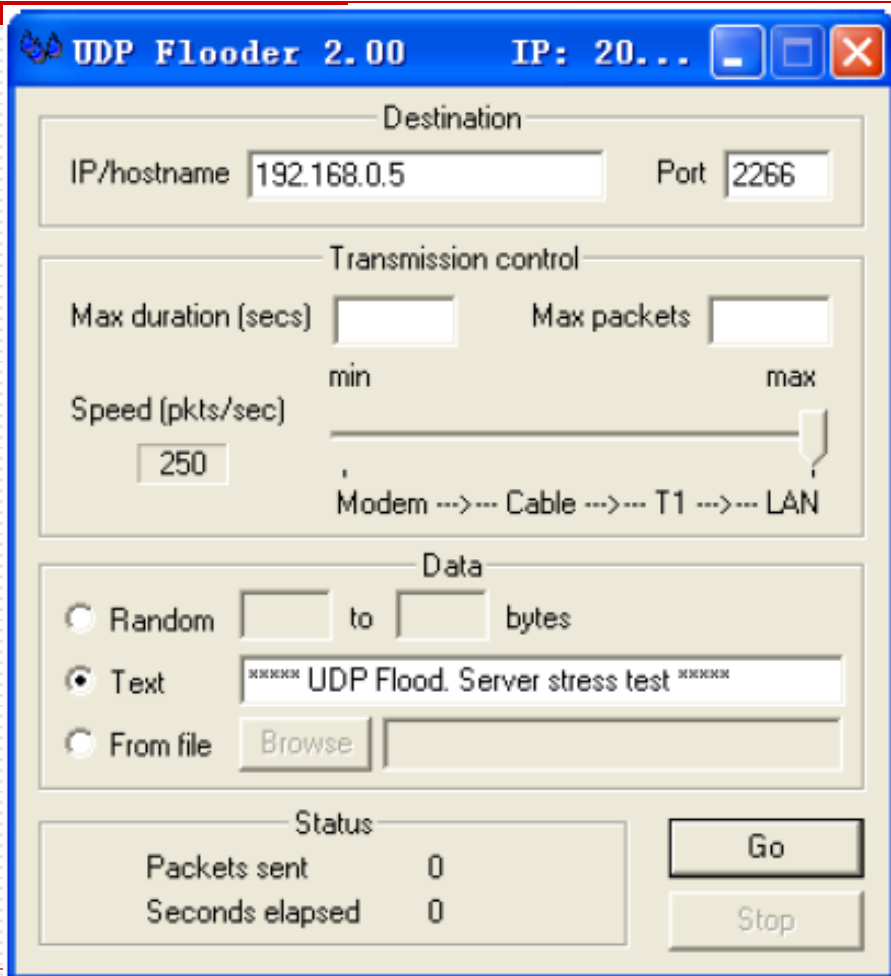
Dos常用攻击工具

Synbomb



Dos常用攻击工具

UDP Flooder



6.3 分布式拒绝服务攻击

- **6.3.1** 分布式拒绝服务攻击简介
- **6.3.2** 分布式拒绝服务攻击造成的影响
- **6.3.3** 分布式拒绝服务攻击工具

6.3.1 分布式拒绝服务攻击简介

- ❑ **分布式拒绝服务DDoS (Distributed Denial of Service)**攻击指借助于客户/服务器技术，将多个计算机联合起来作为攻击平台，对一个或多个目标发动**DoS**攻击，从而成倍地提高拒绝服务攻击的威力。
- ❑ 可以使得分散在互联网各处的机器共同完成对一台主机攻击的操作，从而使主机看起来好象是遭到了不同位置的许多主机的攻击。
- ❑ 这些分散的机器可以分别进行不同类型的攻击。

6.3.1 分布式拒绝服务攻击简介

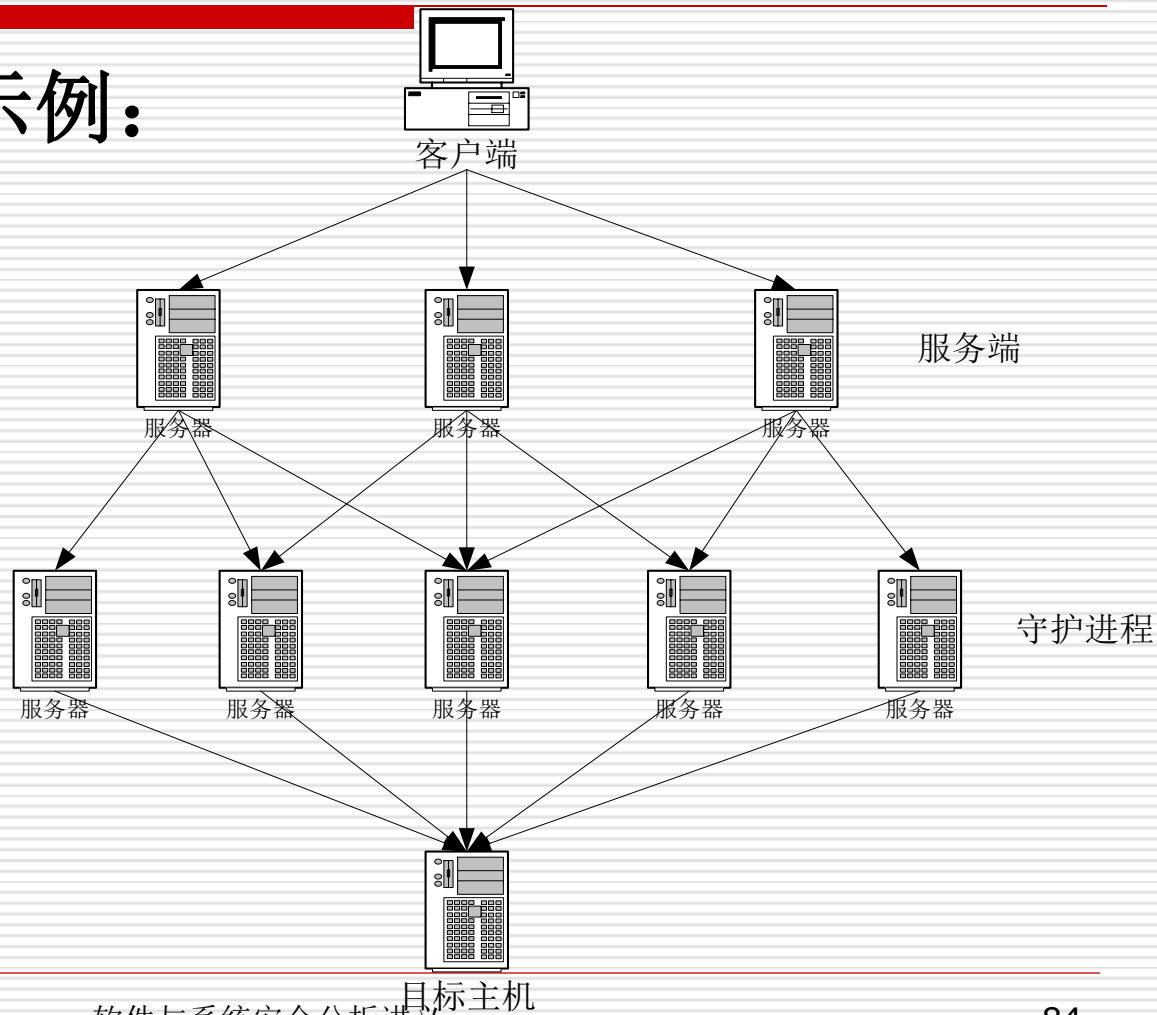
- ❑ 在进行分布式拒绝服务攻击前，入侵者必须先控制大量的无关主机，并在这些机器上安装进行拒绝服务攻击的软件。
- ❑ 互联网上充斥着安全措施较差的主机，这些主机存在系统漏洞或配置上的错误，可能是一些没有足够安全技术力量的小站点或者一些企业的服务器，入侵者轻易就能进入这些系统。
- ❑ 由于攻击者来自于范围广泛的**IP**地址，而且来自每台主机的少量的数据包有可能从入侵检测系统的眼皮下溜掉，这就使得防御变得困难。

6.3.1 分布式拒绝服务攻击简介

- 分布式拒绝服务攻击的软件一般分为**客户端、服务端与守护程序**，这些程序可以使协调分散在互联网各处的机器共同完成对一台主机攻击的操作，从而使主机遭到来自不同地方的许多主机的攻击。
- 客户端：也称攻击控制台，它是发起攻击的主机
- 服务端：也称攻击服务器，它接受客户端发来的控制命令
- 守护程序：也称攻击器、攻击代理，它直接（如**SYN Flooding**）或者间接（如反射式**DDoS**）与攻击目标进行通信

6.3.1 分布式拒绝服务攻击简介

□ DDoS攻击示例：



6.3.1 分布式拒绝服务攻击简介

- ❑ 入侵者通过客户端软件向服务端软件发出攻击指令，服务端在接收到攻击指令后，控制守护进程向目标主机发动攻击。
- ❑ 采用三层结构的做法是确保入侵者的安全，一旦客户端发出指令后，客户端就能断开连接，由服务端指挥守护进程攻击。客户端连接和发送指令的时间很短，隐蔽性极强。

6.3.1 分布式拒绝服务攻击简介

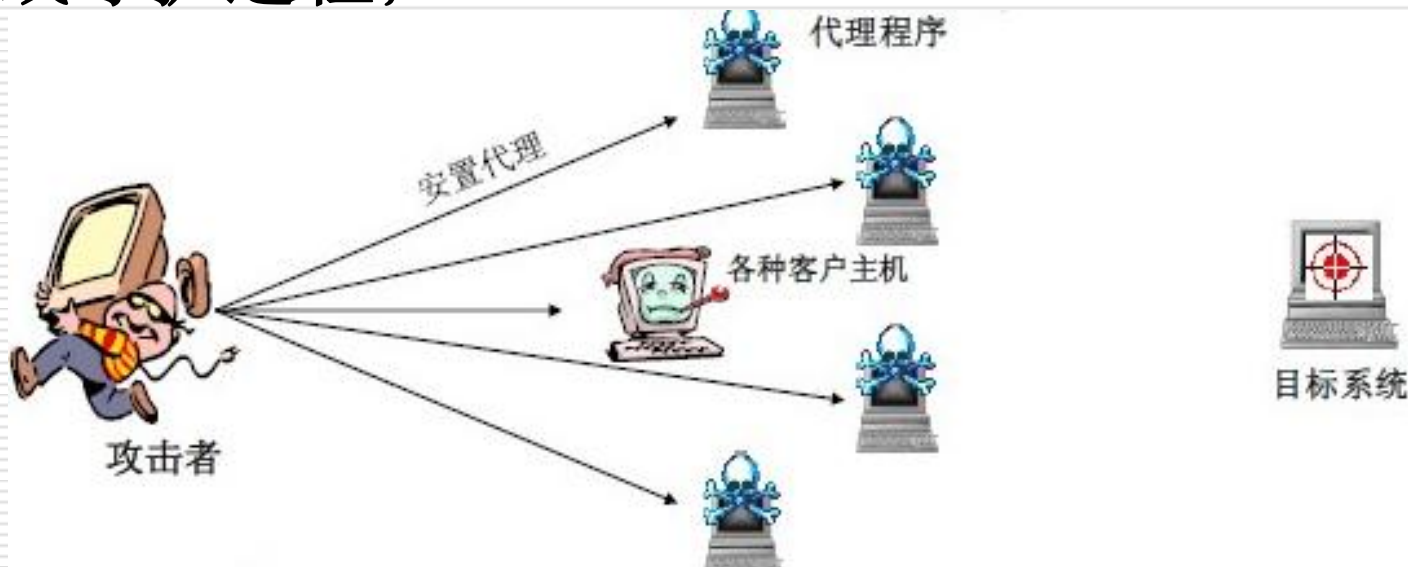
- ❑ 入侵者先控制多台无关主机，在上面安装守护进程与服务端程序。
- ❑ 当需要攻击时，入侵者从客户端连接到安装了服务端软件的主机上，发出攻击指令，服务端软件指挥守护进程同时向目标主机发动拒绝服务攻击。
- ❑ 目前流行的分布式拒绝服务攻击软件一般没有专用的客户端软件，使用**telnet**进行连接和传送控制命令。

6.3.1 分布式拒绝服务攻击简介

- 通常情况下，服务端与守护进程（代理程序）间并不是一一对应的关系，而是多对多的关系。也就是说，一个安装了守护进程的主机可以被多个服务端所控制，一个服务端软件也同时控制多个守护进程。

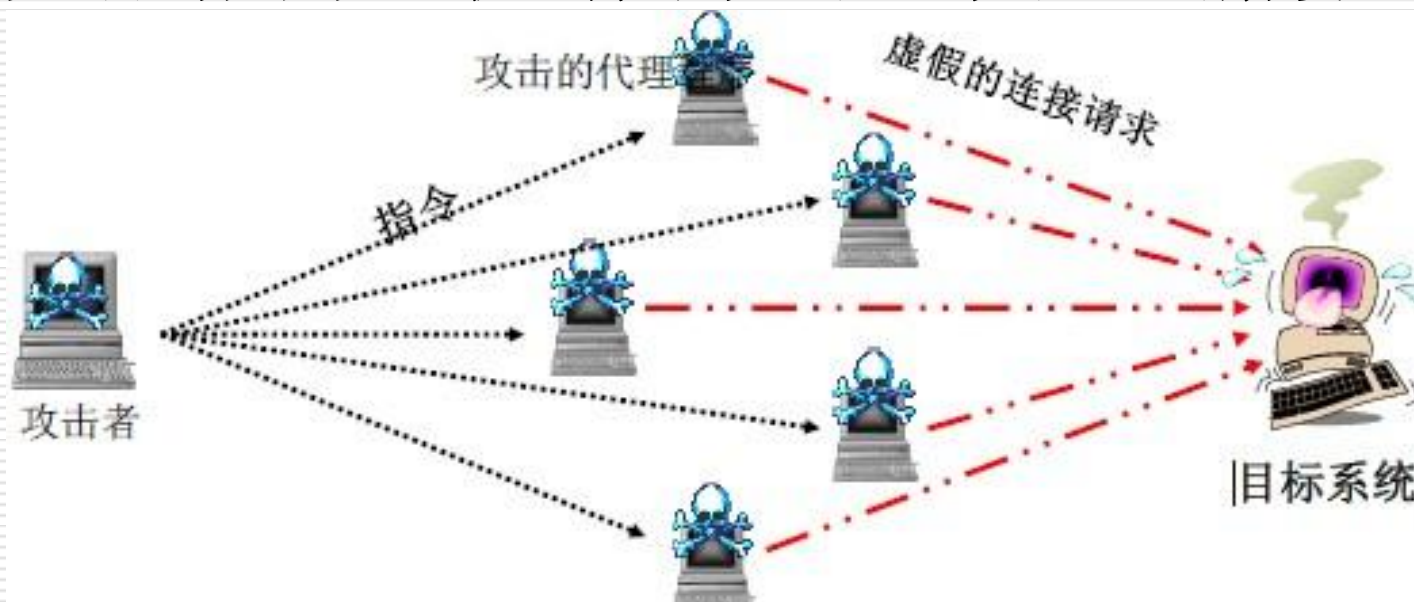
DDoS攻击时序

- 攻击准备：1) 探测扫描大量主机以寻找可入侵主机；2) 入侵有安全漏洞的主机并获取控制权，在每台被入侵主机中安装攻击所用的代理程序或守护进程；



DDoS攻击时序

- 发起攻击：**3)**向安装有代理程序的主控端主机发出命令，由它们来控制代理主机上的守护进程进行协同入侵。**4)**包括虚假的连接请求在内的大量残缺的数字包攻击目标系统，最终将导致它因通信淤塞而崩溃。



6.3.2 DDoS造成的影响

- 被**DDoS**攻击时的现象
- **DDoS**攻击对**Web**站点的影响

被DDoS攻击时的现象

- ❑ 被攻击主机上有大量等待的**TCP**连接;
- ❑ 网络中充斥着大量的无用的数据包, 源地址为假;
- ❑ 制造高流量无用数据, 造成网络拥塞, 使受害主机无法正常和外界通讯;
- ❑ 利用受害主机提供的服务或传输协议上的缺陷, 反复高速的发出特定的服务请求, 使受害主机无法及时处理所有正常请求;
- ❑ 严重时会造成系统死机。

DDoS攻击对Web站点的影响

- ❑ 当对一个**Web**站点执行 **DDoS** 攻击时，这个站点的一个或多个**Web**服务会接到非常多的请求，最终使它无法再正常使用。
- ❑ 在一个**DDoS**攻击期间，如果有一个不知情的用户发出了正常的页面请求，这个请求会完全失败，或者是页面下载速度变得极其缓慢，看起来就是站点无法使用。

DDoS攻击案例

- **2015年6月15日17点31分**，阿里云某用户反馈业务遭受到大流量攻击，攻击峰值流量峰值**20Gbps**。经过云盾安全专家分析，主要攻击类型为**SYN Flood**、**ACK Flood**、**Connection Flood**（连接耗尽）、**TCP**连接等网络层**DDoS**攻击。在阿里云云盾系统的防护下，攻击在**30s**内被成功拦截。但事情并没有结束。

DDoS攻击案例

- ❑ **18点04分和18点16分**，阿里云再次接到两位用户的反馈，与这前的攻击手法类似，规模更大峰值达到**30Gbps**。
- ❑ 云盾安全人员发现，黑客一直在进行短时间大流量的轮换攻击，每次持续不超过**2分钟**，打打停停，并根据攻击的效果快速切换被攻击的目标。
- ❑ 经客户经理确认，三个用户均为贵金属行业用户。

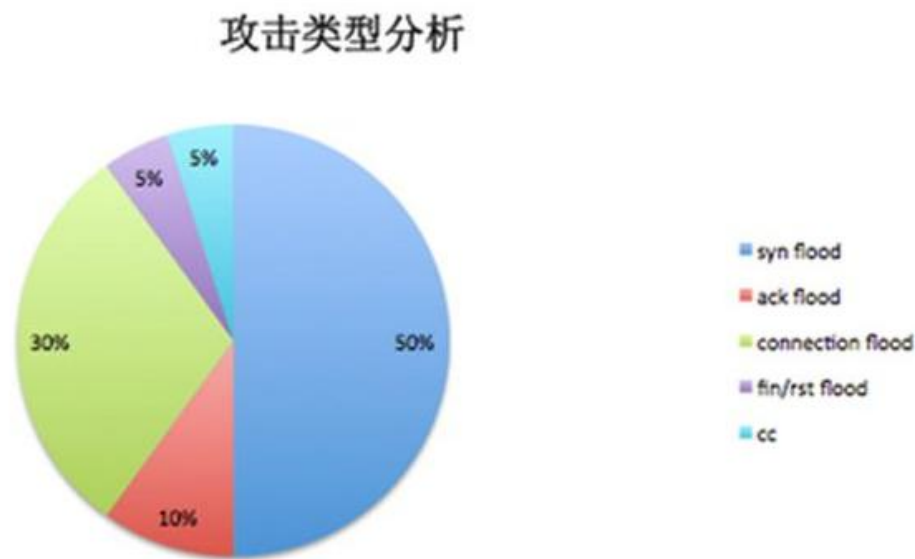
DDoS攻击案例

- 而紧接着在**10到20**分钟之内，又有两个用户反馈遭到类似的**DDoS**攻击，同样均为贵金属行业的企业客户
- 此后的一段时间，攻击者仍不断尝试攻击多个网站，经过分析发现目标均是针对贵金属行业内多家有影响力的网站。

攻击分析

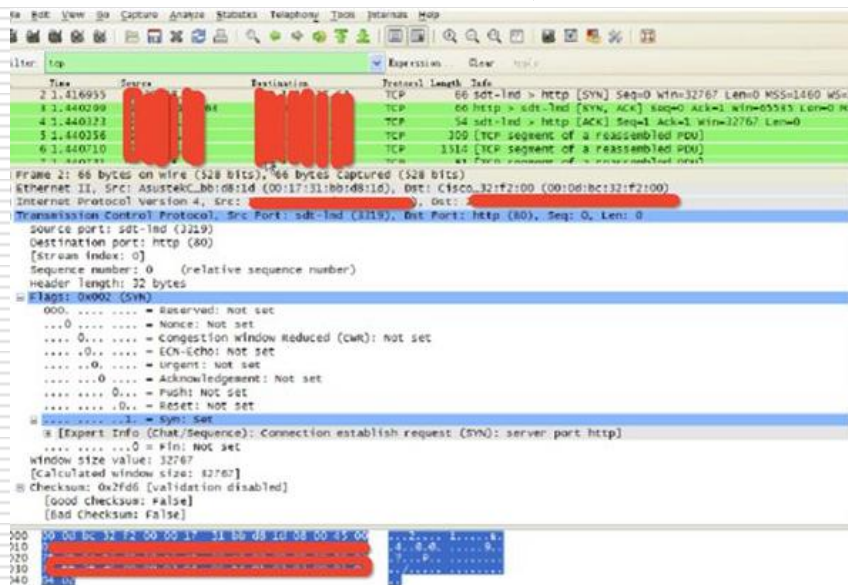
- 黑客在此次攻击事件中主要采用了**SYN Flood**、**TCP Flood**等连接耗尽为策略的网络层**DDoS**攻击。云盾高防在防护此类攻击的时候，主要使用了**SYN**源认证的防护机制和针对**TCP**连接的防护算法，达到了很好的处置效果。

从右图可见，一半的攻击是**SYN Flood**攻击。



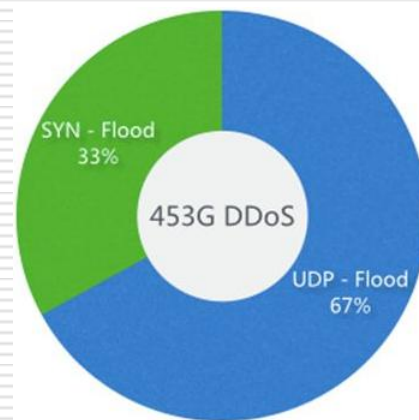
攻击分析

- 通过抓包可看到攻击者在攻击的时候，并没有采用大包的方式，也就是说，黑客并没有增加**SYN**数据包的负载。我们推测，黑客采用非常传统的方式，而没有制造畸形数据包，可能是希望绕过云盾非正常数据包的检测策略。



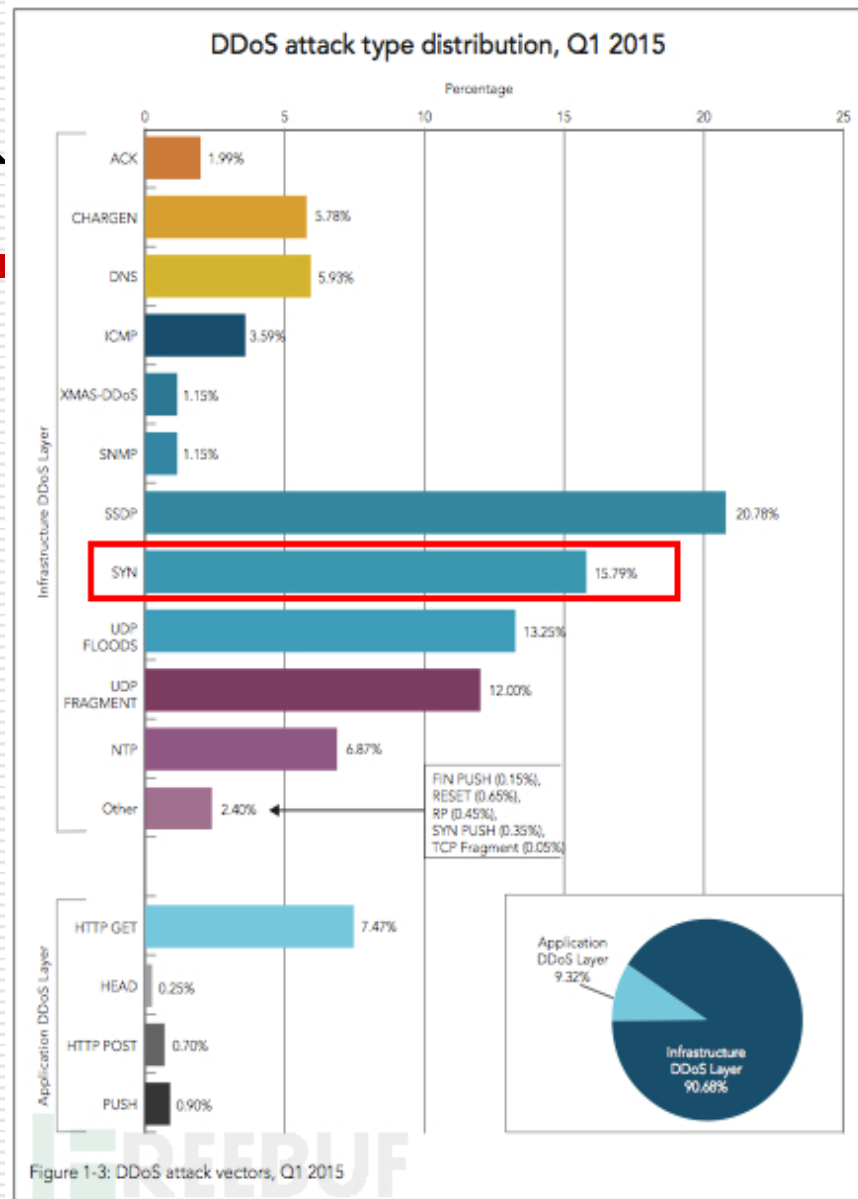
SYN Flood没有过时

- ❑ **SYN Flood**作为最为传统的**DDoS**攻击手段之一。很多安全专家认为**SYN Flood**已经过时，将逐渐被应用层**DDoS**攻击取代。真的是这样吗？
- ❑ 在**2014年12**月底创造互联网历史上最大攻击流量——**453.8G**的**DDoS**攻击事件中，对阿里云某游戏用户的近三分之一的攻击流量是**SYN Flood**。



SYN Flood没有过时

- Akamai在5月份发布的《互联网状态》报告显示，**SYN Flood** 攻击是第二大**DDoS**攻击方式（**15.79%**）。



SYN Flood没有过时

- ❑ 为什么如此“传统”的**DDoS**攻击手段仍被黑客如此大规模的使用呢？
- ❑ 一方面由于这种攻击的成本很低，工具丰富，操作方便；另一方面，**SYN Flood**攻击手法也在不断发展，比如增加**Payload**的大小等方式，以提高攻击的效率。

DDoS攻击态势

1. 大流量攻击呈现增长趋势。

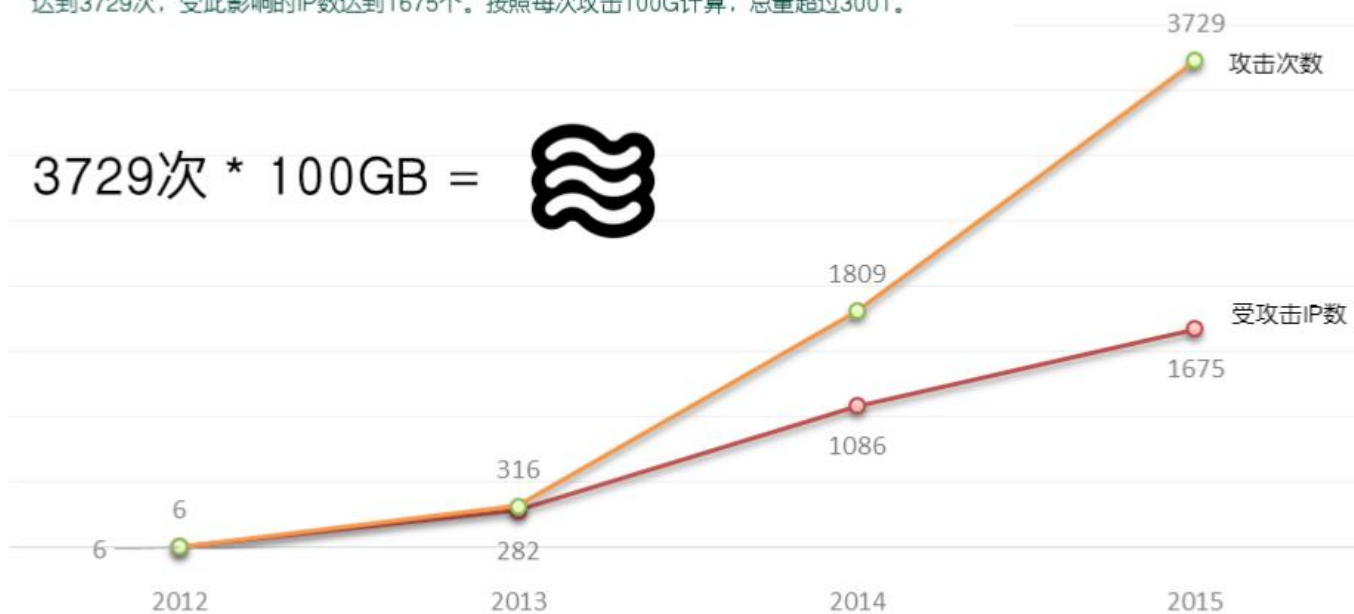
- 在 **2015** 年上半年的**DDoS**攻击中，百**G**以上的攻击频次明显增大。
- 以某运营商为例，**2015** 年受 **100G** 以上流量攻击的**IP** 数增长到**1675** 个，攻击的次数增长到**3729**次，照此计算 **100G** 以上的攻击总量已经超过 **300T**。这个趋势相比 **2014** 年 **IP** 数量有所增长，单 **IP** 受到 **100G** 以上流量攻击的次数也明显上升。而 **100G** 以下的攻击总量要远远超过这个数量。

DDoS攻击态势

100G以上的攻击总量超过300T

近3年来，大流量攻击及频次有明显上升的态势。2015年上半年100G以上的攻击次数达到3729次，受影响的IP数达到1675个。按照每次攻击100G计算，总量超过300T。

3729次 * 100GB =



Source : 2015 H1 NSFOCUS DDoS THREAT REPORT

www.nsfocus.com

DDoS攻击态势

2. 大流量攻击走向云端

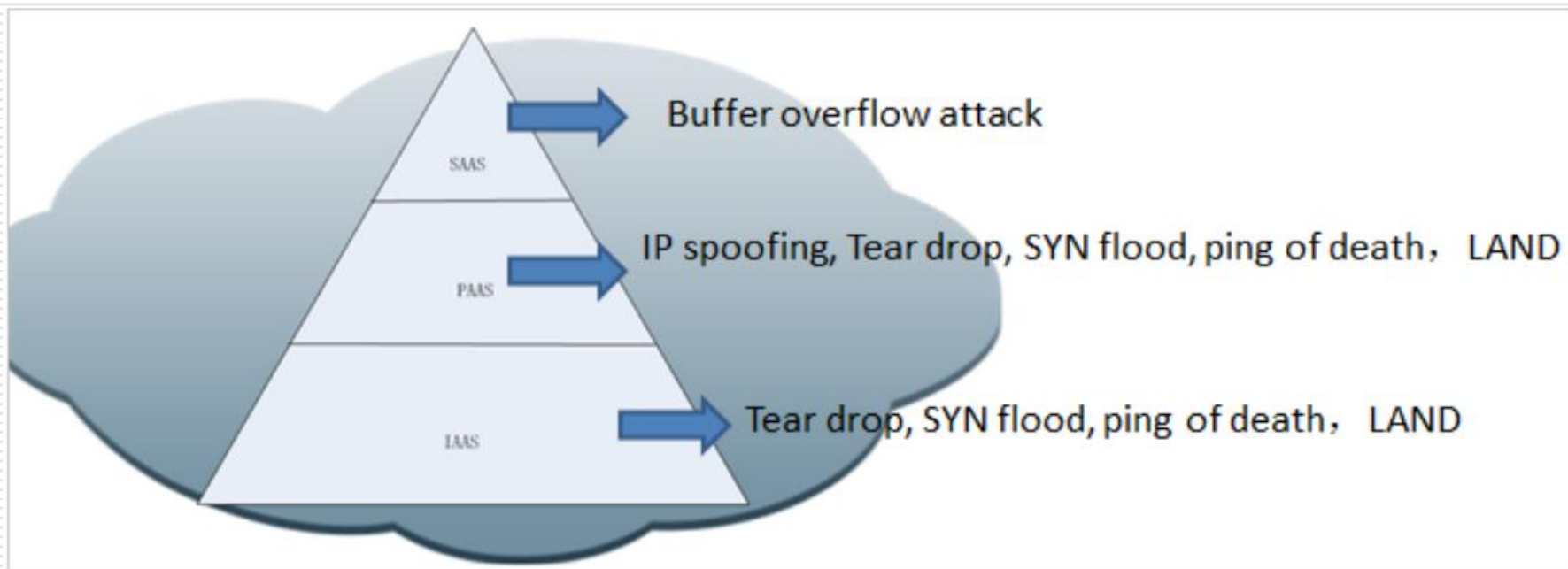
- 在 **DDoS** 大流量攻击兴起的同时，为了抵御风险免受其害，许多用户将其业务向云端迁移，云计算技术的诸多优点使得云服务得以广泛应用。
- 中国信息通信研究院的报告显示，我国公共云服务市场规模大概在 **72** 亿元左右，比去年增长 **47.5%**。中国私有云市场规模也在不断扩张，**2014** 年国内私有云市场规模大概在 **246** 亿人民币左右，增长速度将近 **30%**。

DDoS攻击态势

- 云服务的增多在为用户带来了便利的同时，也在安全方面也带来了两个方面的变化：
 - 客户端轻量化，客户端原本的计算任务，大幅度向云端转移，云端的流量会越来越大，这将会被大流量 **DDoS** 攻击所利用；
 - 环境复杂化，随着业务环境虚拟化，从业务更加灵活多变到运维管理，其中不断产生新的不确定性，都可能为新的 **DDoS** 攻击形式创造机会。

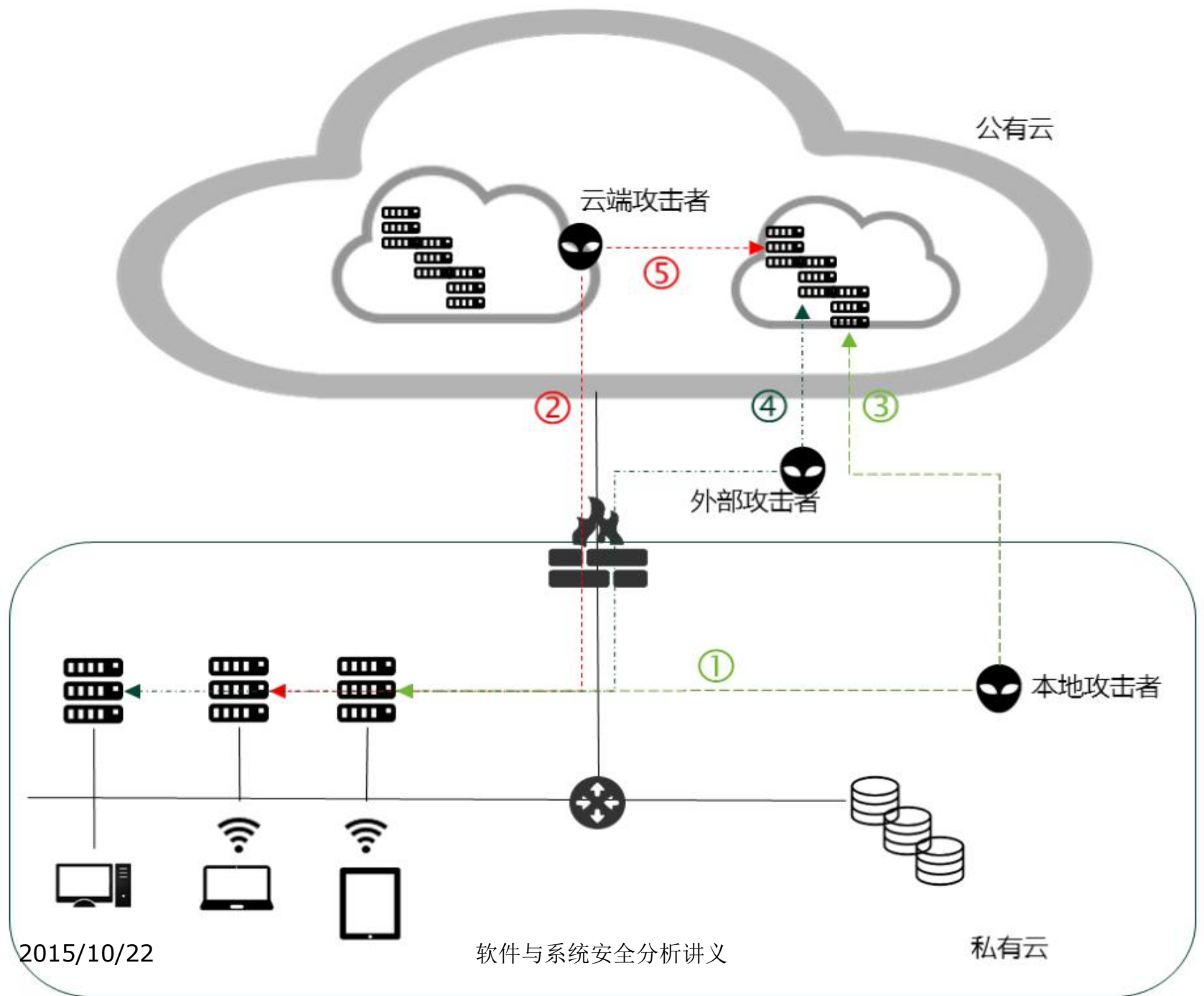
DDoS攻击态势

- 云计算及云服务多种多样，但从基础架构的角度来说，业界比较公认的理论将其分为 **3** 层基础架构。



攻击路径

- 从攻击路径来说，一般情况下按照攻击者所处的位置，可以大致分为三种攻击路径，
 - 从公有云发起的攻击；
 - 从私有云发起的攻击；
 - 从云外部发起的攻击。
- 这三种攻击路径按照攻击目标的不同进行叠加，至少会产生 **6** 种攻击路径。



私有云攻击

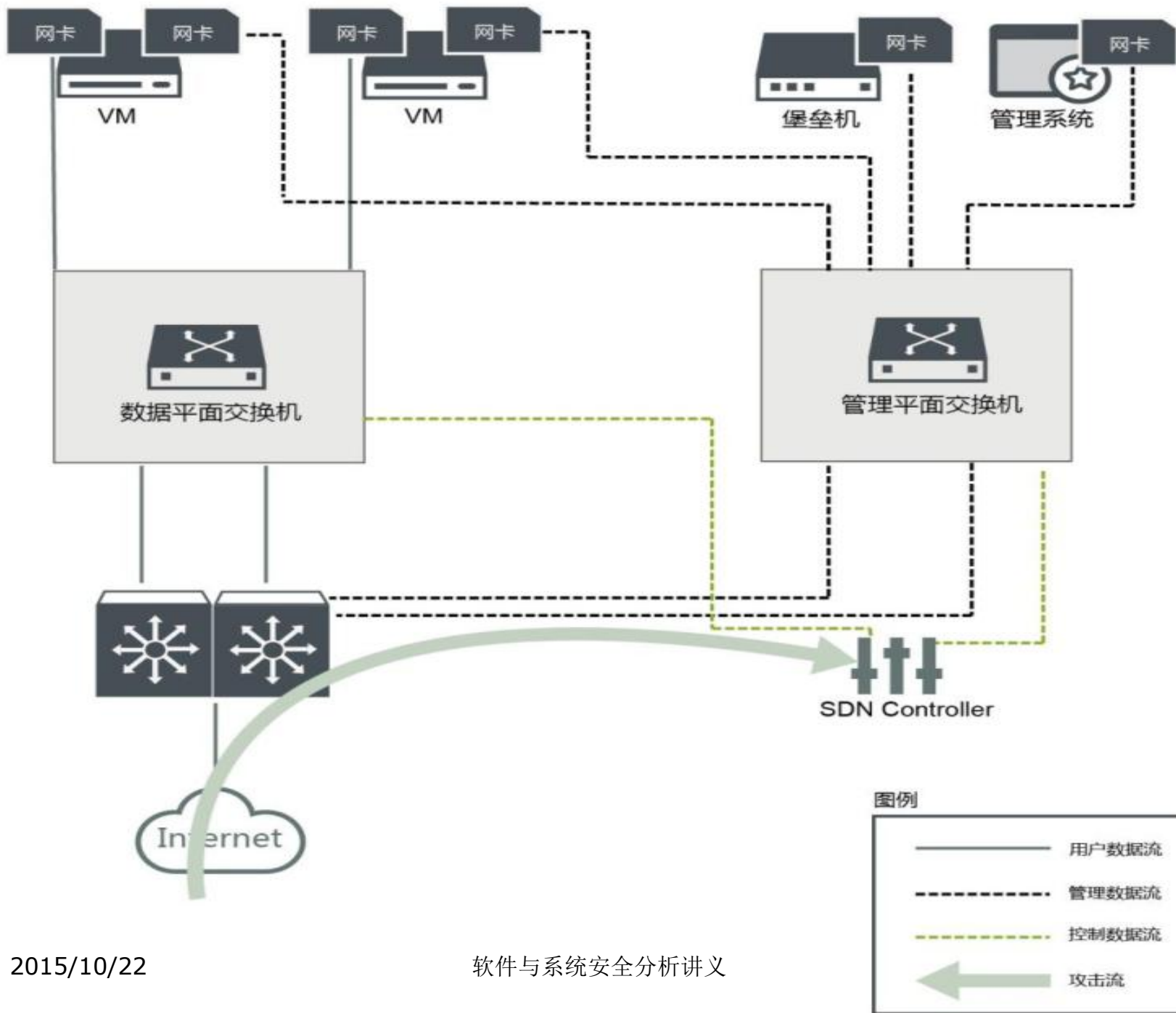
- ❑ 在企业私有云场景中，需要考虑的不仅仅是来自外部②的攻击，也需要防御内部的攻击，这里的关键在于云是否有配置对内部流量的清洗，内部的流量是否经过清洗检测设备，也就是①类型的攻击需要进行防御。
- ❑ 另外，对于内部向外的攻击③，也可能存在。

公有云攻击

- ❑ 在公有云的场景中，对于③④这样的攻击类型较为困难，因为云资源动态分配和动态拓扑，可以有效的进行流量负载平衡，且云服务商的流量清洗机制可以有效缓解这种攻击类型，这也就是很多的业务都愿意迁往云端的原因。
- ❑ 但有一类攻击类型虽然在目前阶段很少看到，但较为隐蔽，后续随着云端服务的竞争，很容易出现这种攻击形式。比如⑤。这种攻击形式，攻击者可能从相同的物理网络甚至同一物理机上发起攻击，此时流量将在本地虚拟交换机或者 **Hypervisor** 上处理，并不会经过外部的流量检测及 **DDoS** 防护设备。这种风险成为现在已知的云端 **DDoS** 风险。

SDN 控制器成风险

- 在云计算体系架构中有一项关键性技术 **SDN**（软件定义网络，Software Defined Network），它为网络的计算资源和存储资源的动态分配提供了易管理的机制，业界已经有厂商利用 **SDN** 技术实现**DDoS**防护的案例，但恰恰是这一点使其成为云内的安全薄弱环节。
- 至少会有两种可能发生的攻击形式：
 - **SDN**控制器被 **DDoS** 攻击后，造成数据调度的混乱，使其管理的网络大面积受到影响乃至瘫痪；
 - **SDN**控制器为了保持自己可用，将所有数据流导向被攻击的虚拟主机，最终造成虚拟主机被攻击直至瘫痪。

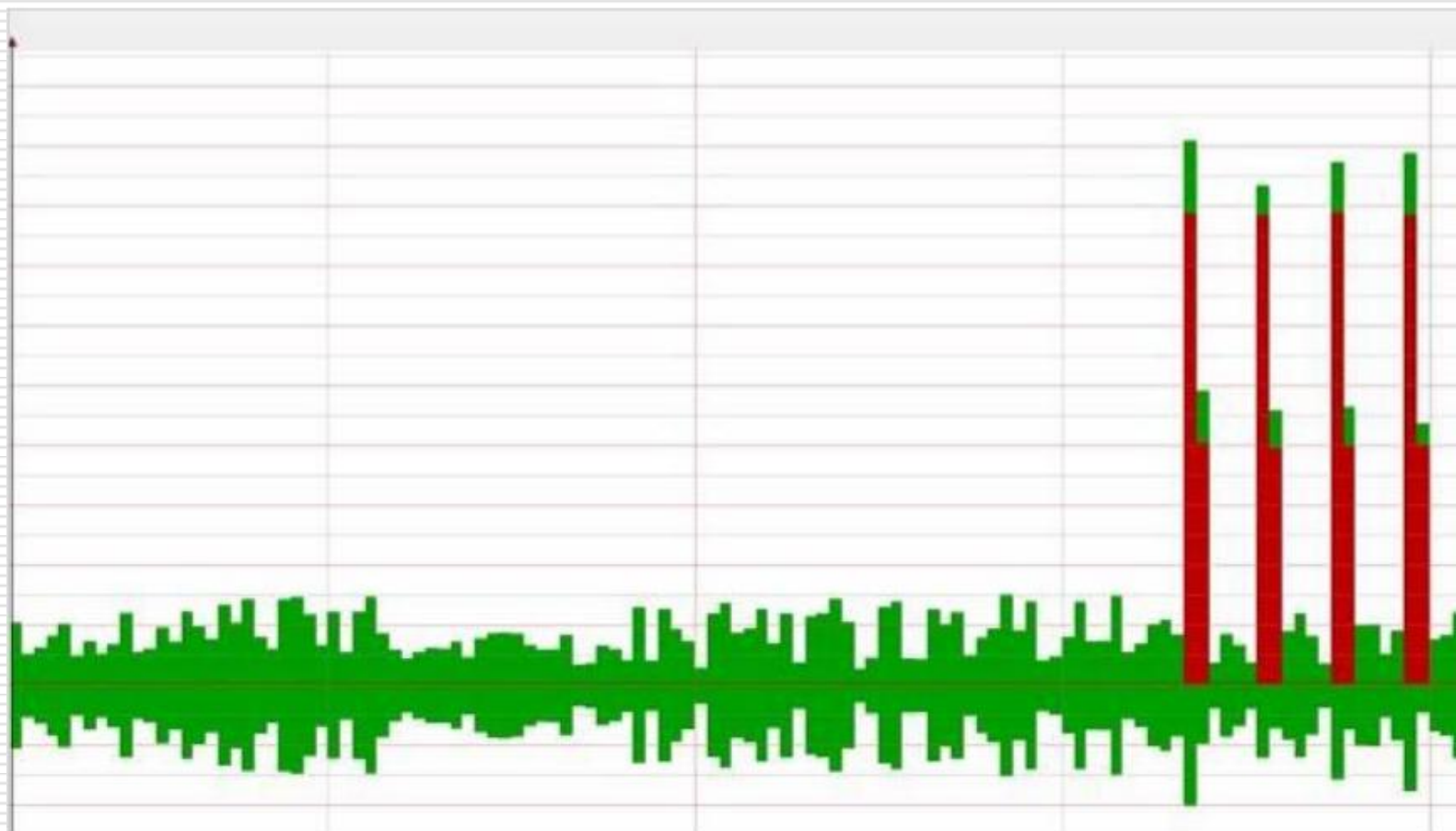


DDoS攻击态势

3. 小流量“快”攻击变身脉冲攻击

- 虽然大流量攻击乃至云端攻击会越来越多的出现，但这并不意味着小流量攻击就消失了。相反，在一些行业中，小流量攻击有着特殊的目的，与大流量（百 **G** 以上）及超大流量（**500G** 或 更高）相比：
 - 这些攻击因为其流量小，不会引起业界的关注；
 - 这些小流量隐藏在大流量其中，难以辨识；
 - 有些攻击时长小到防护设备难以捕获，很难完整呈现其攻击过程。
- 这些特点决定了小流量攻击不仅不会被攻击者抛弃。**2015** 年上半年，**0-30** 分钟时长的攻击环比上涨 **4.52%**，**1** 分钟以下的攻击占总量的 **42.74%**。

脉冲攻击的片段



DDoS攻击态势

4. 小流量“慢”攻击业务逻辑

- 在众多小流量攻击案例中，针对业务逻辑设计问题的慢速攻击也具有代表性。
- 不同于游戏领域的小而快，这种攻击类型的小流量慢速攻击由于间隔时间较长，从协议、流量、逻辑上来看也没有明显异常，但是却针对协议的弱点或者应用逻辑上的弱点，故意延长通信的时间、占用连接的资源、增加服务器的处理过程，进行资源消耗，使目标的 **CPU** 资源、内存资源、连接池等耗尽，最终产生拒绝服务。

6.3.3 DDoS的工具

- ❑ **TFN2K**
- ❑ **Trinoo**
- ❑ **Stacheldraht**
- ❑ 其他拒绝服务攻击工具

TFN2K—介绍

- ❑ **TFN(Tribe Flood Network)**是德国著名黑客**Mixer**编写的分布式拒绝服务攻击的攻击工具，它是一个典型的分布式拒绝服务攻击的工具。
- ❑ **TFN**由服务端程序和守护程序组成，能实施**ICMP flood**、**SYN flood**、**UDP flood**和**Smurf**等多种拒绝服务攻击。

TFN2K--特点

- ❑ **TFN2K**的选择面宽，**Solaris**、**Linux**、**Windows NT/2000**上都能运行。
- ❑ **TFN2K**的另一个特点是服务端控制守护进程发动攻击时，可以定制通信使用的协议，**TFN2K**目前可以使用的**TCP**、**UDP**、**ICMP**三种协议中的任何一种。
- ❑ 服务端向守护进程发送的控制指令，守护进程是不会进行回复。由于这一特点，网络中的**TFN2K**的隐蔽性更强，检测更加困难，因为服务端可以将命令的数据报文的源地址信息进行伪造。

TFN2K--特点(2)

- ❑ **TFN2K**所有命令都经过了**CAST-256**算法（**RFC2612**）加密。加密关键字在程序编译时定义，并作为**TFN2K**客户端程序的口令。并且所有加密数据在发送前都被编码（**Base64**）成可打印的**ASCII**字符。**TFN2K**守护程序接收数据包并解密数据。
- ❑ 为保护自身，守护进程还能通过修改进程名方式来欺骗管理员，掩饰自己的真正身份。
- ❑ 总之，**TFN2K**采用的**单向通信、随机使用通信协议、通信数据加密**等多种技术以保护自身，使得实时检测**TFN2K**更加困难。

TFN2K—检测

- **TFN2K**有一个独特的设计，在每一个数据包后面填充了**16个零(0x00)**，这样做的目的是为了使数据包的长度不固定，欺骗某些防火墙或者入侵检测系统。
- 然而，这项独特的设计也成为了**TFN2K**的弱点。
- **TFN2K**的数据包后面填充的零（**0x00**）在经过**Base64**编码后就变成了**A(0x41)**。这样，尾部的数据包就成为了**TFN2K**的特征。当然这并不是说检测到尾部有**0x41**的数据包就认为网络存在**TFN2K**，不过，如果在网络中大量捕获到这种类型的数据包的时候，管理员就该好好检查网络中的主机了。

TFN2K—检测(2)

- 另一种对**TFN2K**的检测的方法是采用病毒检测的通用做法，采用特征码。
- 虽然**TFN2K**服务端和守护进程的文件名可以随意修改，但是程序中必然存在不会改变的特征字符串，这个不会改变的字符串就是程序的特征码，检查系统中是否存在有这样特征码的程序就能发现系统中存在的**TFN2K**。

TFN2K—防御

□ TFN2K的抵御方法有：

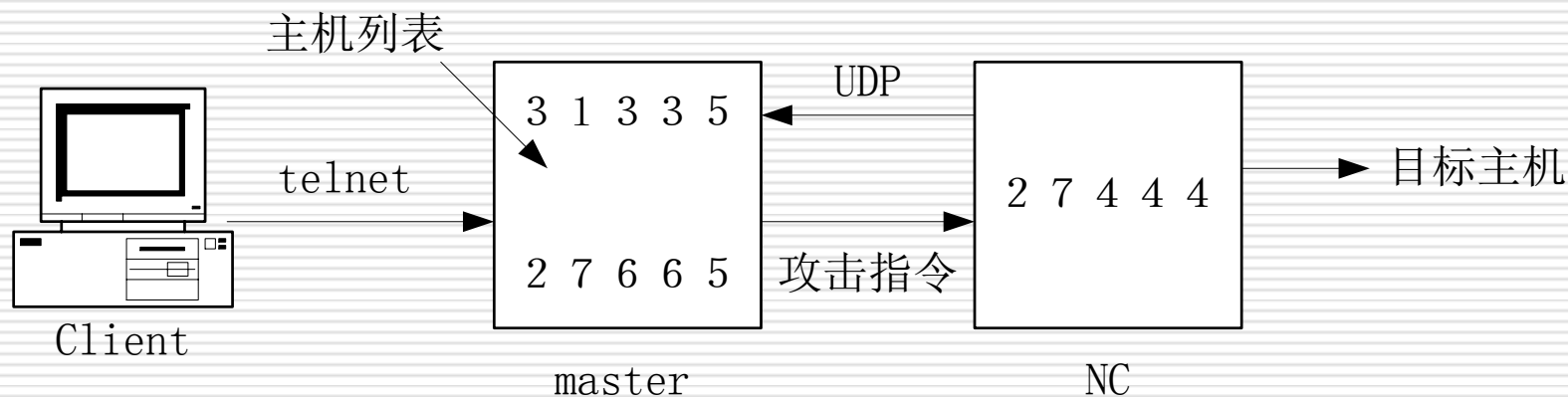
- 加固系统和网络，以防系统被当做DDoS主机。
- 在边界路由器上设置出口过滤，这样做的原因是或许不是所有的TFN2K源地址都用内部网络地址进行伪装。
- 请求上游供应商配置入口过滤。

Trinoo—介绍

- **Trinoo**也是一种比较常见的分布式拒绝服务攻击，**Trinoo**与**TFN2K**相比，虽然在很多方面都略逊一筹，但从总体上来说，**Trinoo**还是一个非常不错的分布式拒绝服务攻击工具。

Trinoo—组成

- **Trinoo**是一个典型的分布式拒绝服务攻击软件，由两部分组成，服务端和守护进程，而没有专门的客户端软件，客户端软件可以使用通用的如**Telnet**来代替。如图：



Trinoo—工作原理

- ❑ **Trinoo**的守护进程**NC**在编译时就将安装有服务程序的主机**IP**地址包含在内，这样，守护进程**NC**一旦运行起来，就会自动检测本机的**IP**地址，并将本机的**IP**地址发送到预先知道的服务器的**31335**端口（服务器开启**31335UDP**端口接收守护进程）。
- ❑ 同时，守护进程也在本机打开一个**27444**的**UDP**端口等待服务器端过来的命令。
- ❑ **Trinoo**的服务器端在收到守护进程发回来的**IP**地址后，就明白已有一个守护进程准备完毕，可以发送指令命令了。
- ❑ 主服务器会一直记录并维护一个已激活守护程序的主机清单。

Trinoo—设计特色

- ❑ **Trinoo**的所有连接都需要口令，连接的口令是编译时就指定的，缺省情况下，服务端连接守护进程的口令是“**144adsl**”，而客户端连接到服务端的口令是“**betaalmostdone**”。不过口令在进行验证时是明文进行传送的。
- ❑ **Trinoo**另一个比较有特色的设计是，当客户端连接到服务端时，如果还有其他的连接建立，**Trinoo**会将一个包含连接**IP**地址的报警信息发送到已连接的主机。这样，入侵者在控制服务端发动攻击时，还能掌握系统上的用户动向，确保**Trinoo**客户端的安全。

Trinoo--基本特性及建议的抵御策略

- ❑ 在**master**程序（服务端）与代理程序（守护程序）的所有通讯中，**Trinoo**都使用了**UDP**协议。入侵检测软件能够寻找使用**UDP**协议的数据流(类型**17**)。
- ❑ **Trinoo master**程序的监听端口是**27655**，攻击者一般借助**telnet**通过**TCP**连接到**master**程序所在计算机。入侵检测软件能够搜索到使用**TCP** (类型**6**)并连接到端口**27655**上的数据流。

Trinoo--基本特性及建议的抵御策略

- ❑ 所有从**master**程序到代理程序的通讯都包含字符串"**I44**", 并且被引导到代理的**UDP** 端口**27444**。入侵检测软件检查到**UDP** 端口**27444**的连接, 如果有包含字符串**I44**的信息包被发送过去, 那么接受这个信息包的计算机可能就是**DDoS**代理。
- ❑ **Master**和代理之间的通讯受到口令的保护, 但是口令不是以加密格式发送的, 因此它可以被“嗅探”到并被检测出来。使用这个口令以及**Dave Dittrich**编写的**Trinot**脚本, 要准确地验证出**Trinoo**代理的存在是很可能的。

Stacheldraht

- ❑ **Stacheldraht** 也是一个分布式拒绝服务攻击，它很多方面类似于**Trinoo**和**TFN**，能发动**ICMP Flood**、**SYN Flood**、**UDP Flood**和**Smurf**等多种攻击。它的主要特色是能进行自动更新。

Stacheldraht(2)

- ❑ **Stacheldraht**跟**TFN**和**trinoo**一样也是基于客户机/服务器模式，其中**Master**程序与潜在的成千个代理程序进行通讯。在发动攻击时，入侵者与**master**程序进行连接。
- ❑ **Stacheldraht**增加了以下新功能：攻击者与**master**程序之间的通讯是加密的，以及使用**rcp (remote copy, 远程复制)**技术对代理程序进行更新。

Stacheldraht(3)

- ❑ **Stacheldraht**同**TFN**一样，可以并行发动数不胜数的**DoS**攻击，类型多种多样，而且还可建立带有伪装源**IP**地址的信息包。
- ❑ **Stacheldraht**所发动的攻击包括**UDP**洪水、**TCP SYN**洪水、**ICMP**回应洪水攻击。

Stacheldraht DDoS攻击的特征及防御

1) 在发动**Stacheldraht**攻击时，攻击者访问**master**程序，向它发送一个或多个攻击目标的**IP**地址。**Master**程序再继续与所有代理程序进行通讯，指示它们发动攻击。

Stacheldraht master程序与代理程序之间的通讯主要是由**ICMP** 回音和回音应答信息包来完成的。配置路由器或入侵检测系统，不允许一切**ICMP**回音和回音应答信息包进入网络，这样可以挫败**Stacheldraht**代理。但是这样会影响所有要使用这些功能的**Internet**程序，例如**ping**。

Stacheldraht DDoS攻击的特征及防御 (2)

2) 代理程序要读取一个包含有效**master**程序的**IP**地址列表。代理会试图与列表上所有的**master**程序进行联系。如果联系成功，代理程序就会进行一个测试，以确定它被安装到的系统是否会允许它改变“伪造”信息包的源地址。

通过配置入侵检测系统或使用嗅探器来搜寻它们的签名信息，可以探测出这两个行为。

Stacheldraht DDoS攻击的特征及防御 (3)

2.1) 代理会向每个**master**发送一个**ICMP**回音应答信息包，其中有一个**ID**域包含值**666**，一个数据域包含字符串“**skillz**”。如果**master**收到了这个信息包，它会以一个包含值**667**的**ID**域和一个包含字符串“**ficken**”的数据域来应答。代理和**master**通过交换这些信息包来实现周期性的基本接触。

通过对这些信息包的监控，可以探测出**Stacheldraht**。

Stacheldraht DDoS攻击的特征及防御 (4)

2.2) 一旦代理找到了一个有效**master**程序，它会向**master**发送一个**ICMP**信息包，其中有一个伪造的源地址，这是在执行一个伪造测试。这个假地址是“**3.3.3.3**”。如果**master**收到了这个伪造地址，在它的应答中，用**ICMP**信息包数据域中的“**spoofworks**”字符串来确认伪造的源地址是奏效的。

通过监控这些值，也可以将**Stacheldraht**检测出来。

Stacheldraht DDoS攻击的特征及防御 (5)

3) Stacheldraht代理并不检查 ICMP 回音应答信息包来自哪里，因此就有可能伪造 ICMP 信息包将其排除。

4) Stacheldraht代理程序与TFN 和 trinoo一样，都可以用一个C程序 DDoS_scan来探测。

其他拒绝服务攻击工具

- ☐ **Trinity**
- ☐ **Shaft**
- ☐ **MStream**

Trinity

- ❑ **Trinity**也是一个能对受害人的站点进行多种类型的“洪水”攻击的工具，能发动**UDP、Fragment、SYN、RST、ACK**以及其他的一些“洪水”攻击。它的特点是可以**通过IRC(Internet Relay Chat,网上实时聊天)或者AOL(American On Line)的ICQ来传递信息**。**Trinity**使用的主通信端口是**6667**，并且它还运行一个后台程序监听**TCP 33270**端口。

Shaft

- **Shaft**分布式拒绝服务攻击的网络结构非常类似**Trinoo**，这个攻击工具没有什么特殊的功能，唯一与其他工具不同的是它所有的**TCP**数据包序列号都是**0x28374839**。

MStream

- **MStream**使用虚假的**ACK**标志**TCP**数据包进行攻击。传输没有经过加密。主控端有口令保护。它有一个与其他工具不同的特点就是在这个程序提供所有连接的用户成功或失败的信息。

6.4 拒绝服务攻击的防御

- 防御的困难之处
- 防御方法

防御的困难之处

□ 不容易定位攻击者的位置

- Internet上绝大多数网络都不限制源地址，也就是伪造源地址非常容易
- 很难溯源找到攻击控制端的位置
- 各种反射式攻击，无法定位源攻击者

□ 完全阻止是不可能的，但是适当的防范工作可以减少被攻击的机会

防御方法

- ❑ 有效完善的设计
- ❑ 带宽限制
- ❑ 及时给系统安装补丁
- ❑ 运行尽可能少的服务
- ❑ 只允许必要的通信
- ❑ 封锁敌意**IP**地址

有效完善的设计

- ❑ 一个站点越完善，它的状况会越好。如果公司有一个运行关键任务的**Web**站点，用户必须连接到**Internet**，但是与路由器之间只有一条单一的连接，服务器运行在一台单一的计算机上，这样的设计就不是完善的。
- ❑ 这种情况下，攻击者对路由器或服务器进行**DoS**攻击，使运行关键任务的应用程序被迫离线。理想情况下，公司不仅要有多条与**Internet**的连接，最好有不同地理区域的连接。
- ❑ 公司的服务位置越分散，**IP**地址越分散，攻击同时寻找与定位所有计算机的难度就越大。

带宽限制

- ❑ 当**DoS**攻击发生时，针对单个协议的攻击会损耗公司的带宽，以致拒绝合法用户的服务。例如，如果攻击者向端口**25**发送洪水般的数据，攻击者会消耗掉所有带宽，所以试图连接端口**80**的用户被拒绝服务。
- ❑ 一种防范方法是限制基于协议的带宽。例如，端口**25**只能使用**25%**的带宽，端口**80**只能使用**50%**的带宽。

及时给系统安装补丁

- 当新的**DoS**攻击出现并攻击计算机时，厂商一般会很快确定问题并发布补丁。如果一个公司关注最新的补丁，同时及时安装，这样被**DoS**攻击的机会就会减少。
- 记住：这些措施并不能阻止**DoS**攻击耗尽公司的资源。还有在安装补丁之前，先要对其进行测试。即使厂商声明它可以弥补**DoS**漏洞，这并不意味着不会产生新的问题。

运行尽可能少的服务

- 运行尽可能少的服务可以减少被攻击成功的机会。
- 如果一台计算机开了**20**个端口，这就使得攻击者可以在大的范围内尝试对每个端口进行不同的攻击。相反，如果系统只开了两个端口，这就限制了攻击者攻击站点的攻击类型。
- 另外，当运行的服务和开放的端口都很少时，管理员可以容易的设置安全，因为要监听和担心的事情都很少了。

只允许必要的通信

- ❑ 这一防御机制与上一个标准“运行尽可能少的服务”很相似，不过它侧重于周边环境，主要是防火墙和路由器。关键是要不仅要对系统实施最少权限原则，对网络也要实施最少权限原则。确保防火墙只允许必要的通信出入网络。
- ❑ 许多公司只过滤进入通信，而对向外的通信不采取任何措施。这两种通信都应该过滤。

封锁敌意**IP**地址

- 当一个公司知道自己受到攻击时，应该马上确定发起攻击的**IP**地址，并在其外部路由器上封锁此**IP**地址。这样做的问题是，即使在外外部路由器上封锁了这些**IP**地址，路由器仍然会因为数据量太多而堵塞，导致合法用户被拒绝对其他系统或网络的访问。
- 因此，一旦公司受到攻击应立刻通知其**ISP**和上游提供商封锁敌意数据包。因为**ISP**拥有较大的带宽和多点的访问，如果他们封锁了敌意通信，仍然可以保持合法用户的通信，也可以恢复遭受攻击公司的连接。

6.5 分布式拒绝服务攻击的防御

- **6.5.1** 分布式拒绝服务攻击的监测
- **6.5.2** 分布式拒绝服务攻击的防御
- **6.5.3** 拒绝服务监控系统的的设计

6.5.1 分布式拒绝服务攻击的监测

- ❑ 许多人或工具在监测分布式拒绝服务攻击时常犯的错误是只搜索那些**DDoS**工具的缺省特征字符串、缺省端口、缺省口令等。
- ❑ 人们必须着重观察分析**DDoS**网络通讯的普遍特征，不管是明显的，还是模糊的。
- ❑ 使用网络入侵监测系统（**NIDS**），根据异常现象在网络入侵监测系统上建立相应规则，能够较准确地监测出**DDoS**攻击。

6.5.1 分布式拒绝服务攻击的监测

- ❑ 实际上，**DDoS**的唯一检测方式是：异常的网络交通流量。
- ❑ 下面将分别介绍**5**种异常模式及相应的解决办法。

异常现象1

□ 大量的**DNS PTR**查询请求

- 根据分析，攻击者在进行DDoS攻击前总要解析目标的主机名。**BIND**域名服务器能够记录这些请求。由于每台攻击服务器在进行一个攻击前会发出**PTR**反向查询请求，也就是说在DDoS攻击前域名服务器会接收到大量的反向解析目标IP主机名的**PTR**查询请求。

异常现象2

□ 超出网络正常工作时的极限通讯流量

- 当DDoS攻击一个站点时，会出现明显超出该网络正常工作时的极限通讯流量的现象。现在的技术能够分别对不同的源地址计算出对应的极限值。当明显超出此极限值时就表明存在DDoS攻击的通讯。
- 因此可以在主干路由器端建立ACL访问控制规则以监测和过滤这些通讯。

异常现象3

□ 特大型的**ICMP**和**UDP**数据包。

- 正常的**UDP**会话一般都使用小的**UDP**包，通常有效数据内容不超过**10**字节。正常的**ICMP**消息也不会超过**64**到**128**字节。
- 那些明显大得多的数据包很有可能就是控制信息通讯用的，主要含有加密后的目标地址和一些命令选项。一旦捕获到（没有经过伪造的）控制信息通讯，**DDoS**服务器的位置就无所遁形了，因为控制信息通讯数据包的目标地址是没有伪造的。

异常现象4

- 不属于正常连接通讯的**TCP**和**UDP**数据包。
 - 隐蔽的DDoS工具随机使用多种通讯协议通过基于无连接通道发送数据。优秀的防火墙和路由规则能够发现这些数据包。另外，那些连接到高于**1024**而且不属于常用网络服务的目标端口的数据包也是非常值得怀疑的。

异常现象5

- 数据段内容只包含文字和数字字符（例如，没有空格、标点和控制字符）的数据包。
 - 这往往是数据经过BASE64编码后而只会含有base64字符集字符的特征。TFN2K发送的控制信息数据包就是这种类型的数据包。TFN2K（及其变种）的特征模式是在数据段中有一串A字符（AAA.....），这是经过调整数据段大小和加密算法后的结果。如果没有使用BASE64编码，对于使用了加密算法数据包，这个连续的字符就是“\0”

6.5.2 分布式拒绝服务攻击的防御

□ 虽然还没有有很好的措施来彻底解决分布式拒绝服务攻击问题，但下面有一些措施能降低系统受到拒绝服务攻击的危害：

- 优化网络和路由结构
- 保护网络及主机系统安全
- 安装入侵检测系统
- 与ISP服务商合作
- 使用扫描工具

优化网络和路由结构

- 理想情况下，提供的服务不仅要有多条与 **Internet** 的连接，而且最好有不同地理区域的连接。这样服务器 **IP** 地址越分散，攻击者定位目标的难度就越大，当问题发生时，所有的通信都可以被重新路由，可以大大降低其影响。

保护网络及主机系统安全

- 本质上，如果攻击者无法获得网络的访问权，无法攻克一台主机，他就无法在系统上安装**DDoS**服务器。要使一个系统成为服务器，首先要以某种手段攻克它。
- 如果周边环境不会被突破，系统能够保持安全，就不会被用于攻击其他系统。
- 对所有可能成为目标的主机都进行优化，禁止不必要的服务，可以减少被攻击的机会。要注意保护主机系统的安全，避免其被攻击者用作傀儡主机，充当**DDoS**的间接受害者。

安装入侵检测系统

- 能否尽可能快地探测到攻击是非常关键的。以 **DDoS** 的角度来看，单位越快探测到系统被入侵或服务器被用来进行攻击，该单位的网络状况越好。
- 借助于入侵检测系统（**IDS**）可以完成这一工作。

安装入侵检测系统

- 有两种常用的**IDS**:基于网络的和基于主机的。
 - 基于网络的**IDS**是网络上被动的设备，负责嗅探通过给定网段的所有数据包。通过查看数据包，查找显示可能的攻击的签名并对可疑行为发出警报。
 - 基于主机的**IDS**运行在一台独立的服务器上，并经常查看审计日志以查找可能的攻击信息。

安装入侵检测系统

- 正如有两种类型**IDS**一样，也有两种构建**IDS**的技术：样式匹配和不规则探测。
 - 样式匹配技术有一个关于已知攻击特征的数据库。当它找到与给定样式相同的数据包时就发出警报。
 - 不规则探测系统决定什么是网络的正常通信，任何不符合这一规则的通信都被标为可疑的。
- 可以想象，基于不规则探测的系统实现起来十分困难，因为对于一个公司正常的通信对于另一个公司则是不正常的。因此大多数入侵检测系统都是基于样式匹配技术的。

与**ISP**合作

- 这一点非常重要。**DDoS**攻击非常重要的一个特点是洪水般的网络流量，耗用了大量带宽，单凭自己管理网络，是无法对付这些攻击的。当受到攻击时，与**ISP**协商，确定发起攻击的**IP**地址，请求**ISP**实施正确的路由访问控制策略，封锁来自敌意**IP**地址的数据包，减轻网络负担，防止网络拥塞，保护带宽和内部网络。

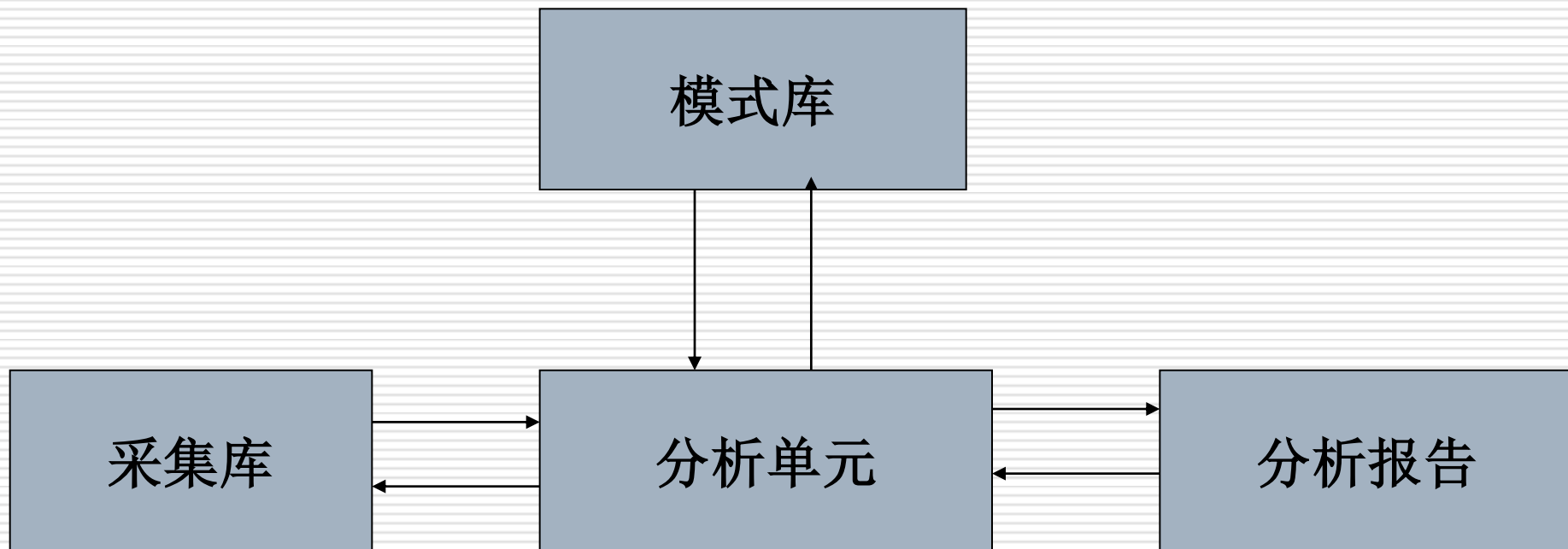
使用扫描工具

- 由于许多公司网络安全措施都进行得很慢，它们的网络可能已经被攻克并用作了**DDoS**服务器，因此要扫描这些网络查找**DDoS**服务器并尽可能的把它们从系统中关闭删除。
- 一些工具可以做到这些，而且大多数商业的漏洞扫描程序都能检测到系统是否被用作**DDoS**服务器。

6.5.3 拒绝服务监控系统的的设计

- 利用总结出的若干拒绝服务攻击数据包的特征，可以进行拒绝服务监控系统的设计，建立网络通信中异常现象的模式库，把实时采集网络数据包与模式库进行模式匹配，得到监控结果。
- 下页为监控系统的结构图。

6.5.3 拒绝服务监控系统的的设计



6.5.3 拒绝服务监控系统的设计

- 上图中，采集器用来收集网络通信信息，并向分析单元提供分析所需的数据，同时还能接收分析单元的指令，进一步采集分析单元所需的特定信息。
- 分析单元可以采用人工神经网络对网络采集信息和模式库进行模式匹配得出分析报告，同时定期进行自学习，更新模式库，及时跟踪并反映拒绝服务攻击模式的最新动态。

6.6 小结

- 无论是**DoS**还是**DDoS**攻击，其目的是使受害主机或网络无法及时接收并处理外界请求，表现为：
 - 制造大流量无用数据，造成通往被攻击主机的网络拥塞，使被攻击主机无法正常和外界通信。
 - 利用被攻击主机提供服务或传输协议上处理重复连接的缺陷，反复高频的发出攻击性的重复服务请求，使被攻击主机无法及时处理其它正常的请求。
 - 利用被攻击主机所提供服务程序或传输协议的本身的实现缺陷，反复发送畸形的攻击数据引发系统错误的分配大量系统资源，使主机处于挂起状态。
- **DoS/DDoS**攻击是很有效的攻击方式，必须注意防范这种攻击。

谢谢各位!