# 拒绝服务与分布式拒绝服务-DOS/DDOS

## 拒绝服务攻击概念介绍

### 拒绝服务攻击概念说明

* **DoS拒绝服务攻击概念介绍**

利用程序漏洞或一对一资源耗尽的方法对服务端发起的攻击。

* **DDoS分布式拒绝服务攻击概念介绍**

一对一的攻击方式完全拼各自的资源，攻击效果比较差；

多对一的攻击汇聚资源能力，重点在于量大，属于资源耗尽型。

### 拒绝服务攻击发展历史

* **从前：**

欠缺技术能力的无赖，ping死你（最难缠的无赖）

* **现在：**

最强大最危险的攻击，攻击方式众多（专业化的要求勒索）；亲身经历：电商网站被勒索、bill gates僵尸程序。贩卖和租用肉鸡已经成为黑产中重要的一部分，最终的办法就是拼资源，投资抗DDoS，或者乖乖交保护费。

提示信息：

匿名者（anonymous）：世界最著名的黑客组织，组织结构宽松，人员来自世界各地。以DDoS攻击著称的无政府主义者-亦正亦邪，攻击恐怖组织也攻击政府宗教机构，近些年来涉足政治斗争，成员露面时均带有Guy Fawkes面具，最早的核心成员来自4chan图片社区，惯常雇佣外围黑客成员发送DDoS攻击。

### 拒绝服务攻击分类介绍

* **DoS网络：**

基于巨量的Flood耗尽目标网络带宽资源，如：ICMP Flood、UDP Flood

* **DoS协议：**

攻击协议漏洞发起的拒绝服务攻击，如：syn flood、ping of death、ARP、DNS、802.11、SSL

* **DoS应用：**

针对应用软件和操作系统漏洞发起的拒绝服务攻击，大量频繁访问消耗系统资源严重的应用（CC）。通常表现为操作系统运行正常，网络流量不大，但服务停止响应，可以是一击毙命的，也可以是耗尽目标资源的。

以上分类并不严谨，不必太过于执着于此



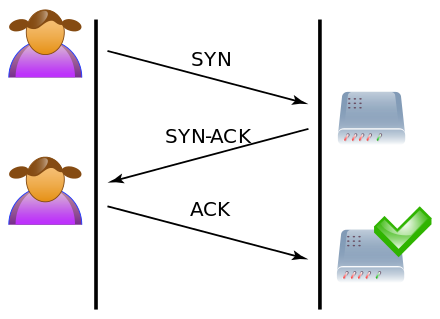
## 拒绝服务攻击协议介绍

### 攻击协议原理介绍说明\_Syn-Flood

SYN Flood (SYN洪水) 是种典型的DoS (Denial of Service，拒绝服务) 攻击。效果就是服务器TCP连接资源耗尽，停止响应正常的TCP连接请求。

说到原理，还得从TCP如何建立连接(Connection)讲起。通信的双方最少得经过3次成功的信息交换才能进入连接全开状态(Full-Open)，行话叫建立TCP连接的3次握手(TCP three-way handshake)。

假设连接发起方是A，连接接受方是B，即B在某个端口（Port）上监听A发出的连接请求。如下图所示，左边是A，右边是B。



A首先发送SYN（Synchronization）消息给B，要求B做好接收数据的准备；B收到后反馈SYN-ACK（Synchronization-Acknowledgement） 消息给A。

这个消息的目的有两个：

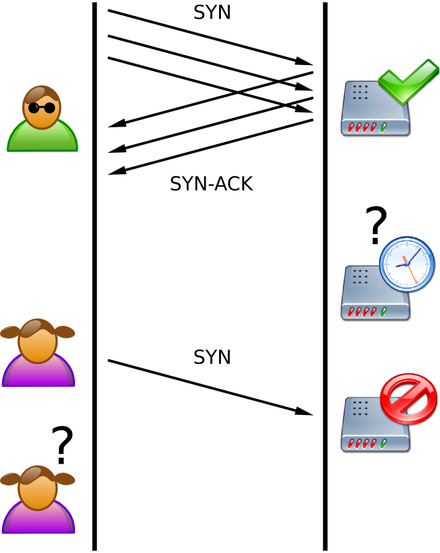
(1) 向A确认已做好接收数据的准备

(2) 同时要求A也做好接收数据的准备

此时B已向A确认好接收状态，并等待A的确认，连接处于半开状态（Half-Open），顾名思义只开了一半；A收到后再次发送ACK(Acknowledgement)消息给B，向B确认也做好了接收数据的准备，至此三次握手完成，“连接”就建立了，实际上只是双方都按对方的要求进入了可以接收消息的状态。以上彼此要求对方确认的“状态”主要是双方将要使用的消息序号(SequenceNum)，TCP为保证消息按发送顺序抵达接收方的上层应用，需要用消息序号来标记消息的发送先后顺序的。

TCP是“双工”(Duplex)连接，同时支持双向通信，也就是双方同时可向对方发送消息，其中SYN和SYN-ACK消息开启了A→B的单向通信通道（B获知了A的消息序号）；SYN-ACK和ACK消息开启了B→A单向通信通道（A获知了B的消息序号）。

以上讨论的是在双方诚实可信，网络正常的理想状况下建立连接。但实际情况是，网络可能不稳定会丢包，使握手消息不能抵达对方，也可能是对方故意不按规矩来，故意延迟或不发送握手确认消息。假设B通过某TCP端口提供服务，B在收到A的SYN消息时，积极的反馈了SYN-ACK消息，使连接进入半开状态，因为B不确定自己发给A的SYN-ACK消息或A反馈的ACK消息是否会丢在半路，所以会给每个待完成的半开连接都设一个Timer，如果超过时间还没有收到A的ACK消息，则重新发送一次SYN-ACK消息给A，直到重试超过一定次数时才会放弃。



做好人是要付出代价的，B为帮助A能顺利连接，需要分配内核资源维护半开连接，那么当B面临海量的大忽悠A时，如上图所示，SYN Flood攻击就形成了。攻击方A可以控制肉鸡向B发送大量SYN消息但不响应ACK消息，或者干脆伪造SYN消息中的Source IP，使B反馈的SYN-ACK消息石沉大海，导致B被大量注定不能完成的半开连接占据，直到资源耗尽，停止响应正常的连接请求。

**利用脚本实现syn\_flood攻击：**

攻击脚本：syn\_flood.py

环境准备：

# 安装Scapy到Debian, Ubuntu或Linux Mint 或kali

apt-get install python-scapy

# 安装Scapy到Fedora或CentOS/RHEL

yum install scapy

说明：在CentOS/RHEL上，你首先需要启用EPEL仓库

# 这条规则的效果是，所有从本机发往目标 IP 地址 192.168.21.117 且带有 TCP RST 标志的数据包都会被丢弃

iptables -A OUTPUT -p tcp --tcp-flags RST RST -d 192.168.21.117 -j DROP

说明：发一个包释放一个连接，这种达不到攻击郊果。要构成攻击效果可以通过iptables限止发送RST包。

Iptables -L -v查看防火墙规则

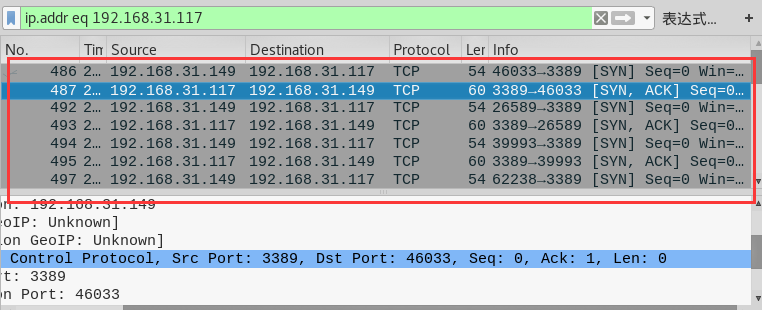
攻击命令：

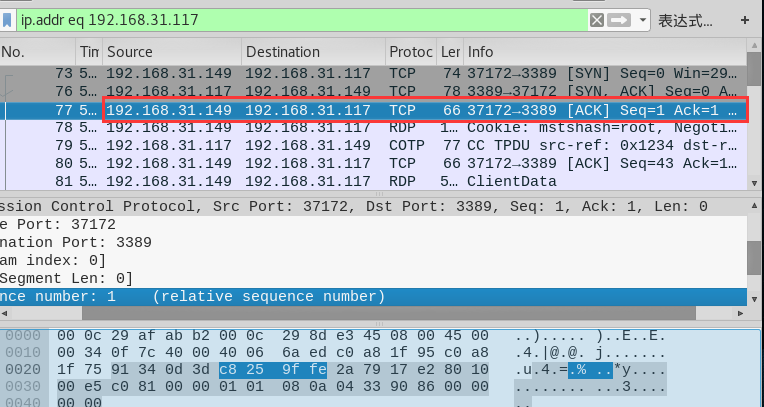
python syn-flodd.py 192.168.110.14 3389 20

**利用软件实现syn\_flood攻击：**

攻击软件：FastSend

**攻击效果抓包截图：**





Win10靶机可以使用Netstat -an来查看有谁进行syn的连接

### 攻击协议原理介绍说明\_socketstress

2008年有Jack C.Louis发现，针对TCP服务的拒绝服务攻击：

* 消耗被攻击目标系统资源，与攻击目标建立大量socket链接
* 完成三次握手最后的ACK包window大小为0（客户端不接收数据），从而占据tcp的连接，当占用的连接量多了，服务端就顶不住了，并且攻击者资源消耗小（CPU 内存 带宽）
* 异步攻击，单机可拒绝服务高配资源服务器
* window窗，实现的TCP流控

**利用脚本实现socktress攻击：**

攻击脚本：socktress

环境准备：

# 防火墙需要进行配置，阻止RST包发出

iptables -A OUTPUT -p tcp --tcp-flags RST RST -d 192.168.21.117 -j DROP

说明：发一个包释放一个连接，这种达不到攻击郊果。要构成攻击效果可以通过iptables限制发送RST包。

攻击命令：HTTP协议访问

# C攻击脚本

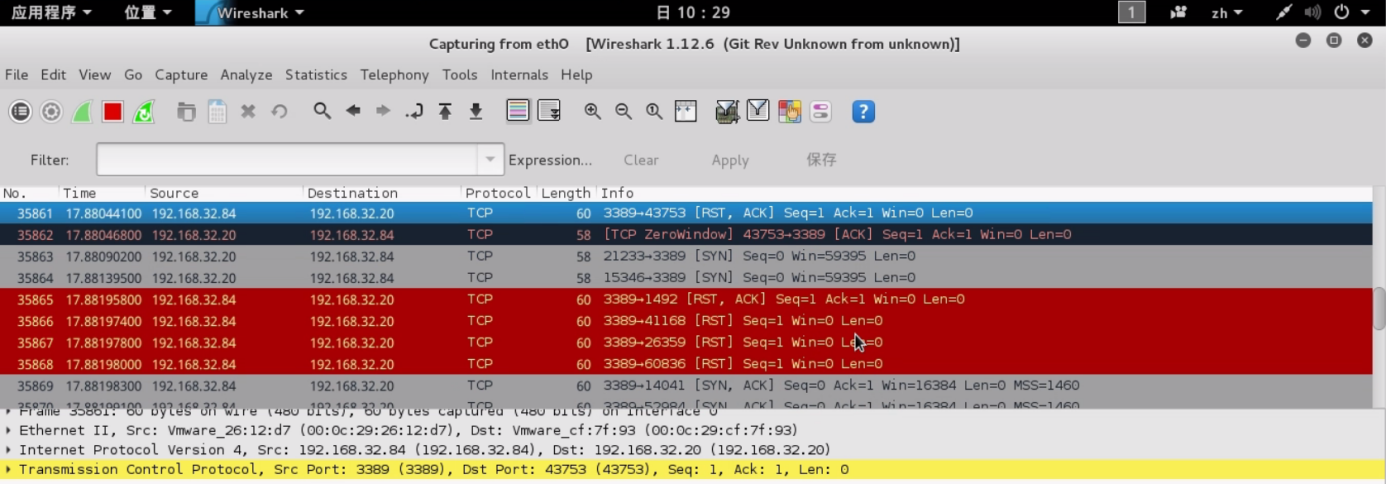
./sockstress 靶机ip:80 eth0

./sockstress 1.1.1.1:80 eth0 -d 100000

./sockstress 1.1.1.1:80 eth0 -p payloads/http

说明：-d是微秒内指定，默认为1000000 改成10之后并发带度更快

**攻击效果抓包截图：**



**攻击防护措施方法：**

直到今天sockstress攻击仍然是一种很有效的DoS攻击方式，甶于建立完整的TCP三步握手，因此使用syn cookie防御无效，根本的防御方法是采用白名单（不实际）

折中对策：限制单位时间内每IP建的TCP连接数

当某个 IP 在 30 秒内向本机 80 端口发起了 10 个或更多的 新连接 请求时，这条规则将丢弃该 IP 发起的所有新连接请求。

iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 30 --hitcount 10 -j DROP

### 攻击协议原理介绍说明\_DNS方法攻击

**产生大流量的攻击方法 DDos**

* 单机的带宽优势
* 巨大单机数量形成的流量汇聚
* 利用协议特性实现放大效果的流量

**DNS协议放大效果**

* 查询请求流量小，但响应流量可能非常巨大
* dig ANY hp.com @202.106.0.20 (流量放大约8倍）

**攻击原理**

* 伪造源地址为被攻击目标地址，向递归域名查询服务器发起查询
* DNS服务器成为流量放大和实施攻击者，大量DNS服务器实现DDoS

### 攻击协议原理介绍说明\_SNMP放大攻击

**简单网络管理协议（**Simple Network Management Protocol**）**

* 服务端UDP 161 / 162
* 管理站（manager /客户端）、被管理设备（agent /服务端）
* 管理信息数据库（MIB）是一个信息存储库，包含管理代理中的有关配置和性能的数据，按照不同分类，包含分属不同组的多个数据对象
* 每一个节点都有一个对象标识符（OID)来唯一的标识
* IETF定义标准的MIB库/厂家自定义MIB库

**攻击原理**

* 请求流量小，查询结果返回流量大
* 结合伪造源地址实现攻击

### 攻击协议原理介绍说明\_NTP放大攻击

**网络时间协议（**Network Time Protocol）

* 保证网络设备时间同步
* 电子设备互相干扰导致时钟差异越来越大
* 影响应用正常运行、日志审计不可信
* 服务端 UDP 123

**攻击原理**

* NTP服务提供monlist (MON\_GETLIST)查询功能，监控NTP服务器的状况
* 客户端查询时，NTP服务器返回最后同步时间的600个客户端IP，每6个IP—个数据包，最多100个数据包（放大约100倍）

### 攻击协议原理介绍说明\_CC攻击

**实现DDOS和伪装攻击：CC(Challenge Collapsar)**

* CC主要是用来攻击页面的
* 可以增加数据库并发访问压力
* 服务端被访问频率越高，占用的系统资源越高

**攻击原理：**

* 攻击者控制某些主机不停地发大量数据包给对方服务器造成服务器资源耗尽，一直到宕机崩溃
* 就是模拟多个用户（多少线程就是多少用户）不停地进行访问那些需要大量数据操作（就是需要大量CPU时间）的页面
* 造成服务器资源的浪费，CPU长时间处于100%，永远都有处理不完的连接直至就网络拥塞，正常的访问被中止

### 攻击协议原理介绍说明\_应用层DoS

**应用服务漏洞**

* 服务代码存在漏洞，遇异常提交数据时程序崩溃
* 应用处理大量并发请求能力有限，被拒绝的是应用或OS

**缓冲区溢出漏洞**

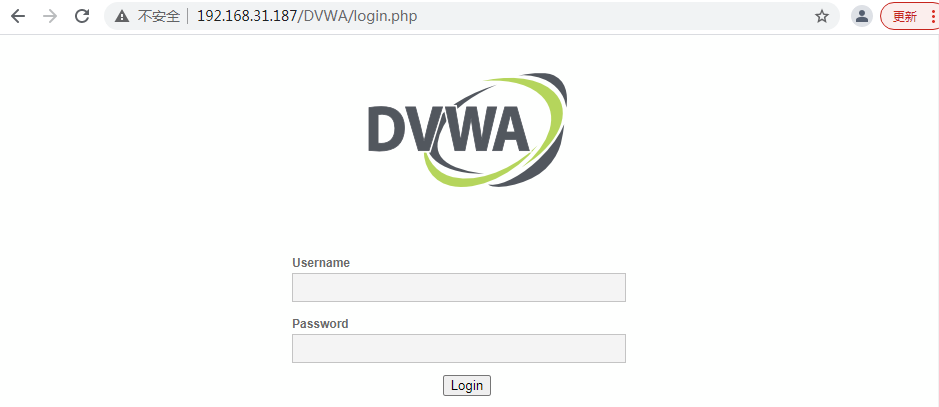
* 向目标函数随机提交数据，特定情况下数据覆盖临近寄存器或内存
* 影响：远程代码执行、DoS
* 利用模糊测试方法发现缓冲区溢出漏洞

**CesarFTP 0.99 服务漏洞**

* ftp\_fuzz.py # MKD/RMD
* MS12-020远程桌面协议DoS漏洞

**攻防演示过程：针对PHP版本低漏洞发起攻击**

开启PHP网站服务：



利用代码向网站发起攻击：

# 在windows系统cmd命令行进行攻击测试即可

php-multipartform-dos-poc.py -t http://192.168.110.12/DVWA/login.php

**应用层DoS攻击方式\_Slowhttptest**

* 低带宽应用层慢速DoS攻击（相对于CC等快速攻击而言的慢速)
* 最早由Python编写，跨平台支持（Linux、win、Cygwin、OSX)
* 尤其擅长攻击apache、tomcat (几乎百发百中)

客户端 100请求 -- 服务端

1. **攻击方法实现：Slowloris、Slow HTTP POST攻击**

耗尽应用的并发连接池，类似于Http层的Syn flood。

HTTP协议默认在服务器全部接收请求之后才开始处理，若客户端发送速度缓慢或不完整，服务器时钟为其保留连接资源池占用，此类大量并发将导致DoS

Slowloris：完整的http请求结尾是\r\n\r\n，攻击发\r\n......

Slow POST: HTTP头content-length声明长度，但body部分缓慢发送

1. **攻击方法实现：Slow Read attack攻击**

* 与slowloris and slow POST目的相同，都是耗尽应用的并发连接池
* 不同之处在于请求正常发送，但慢速读取响应数据
* 攻击者调整TCP window窗口大小，使服务器慢速返回数据

1. **攻击方法实现：Apache Range Header attackk攻击**

* 客户端传输大文件时，体积超过HTTP Body大小限制时进行分段
* 耗尽服务器CPU、内存资源

攻击测试：

torshammer.py –t 192.168.32.84 -p 8080 -r 1000

## 拒绝服务攻击防护思路

### 拒绝服务攻击防护方法

* **利用开源软件实现防护：**

1. TCP连接有效性检查

校验TCP连接时间，指定时间内服务器最少发送的报文数，校验失败的屏蔽时间

TCP连接约束 禁止代理访问

1. 服务器资源占用限制
2. TCP连接会话限制
3. HTTP代理限制



也可以使用阿里的抗ddos服务。