第八讲 综合防护技术

陈伟
Email: chenwei@njupt.edu.cn
Tel: 18951896489

课程内容

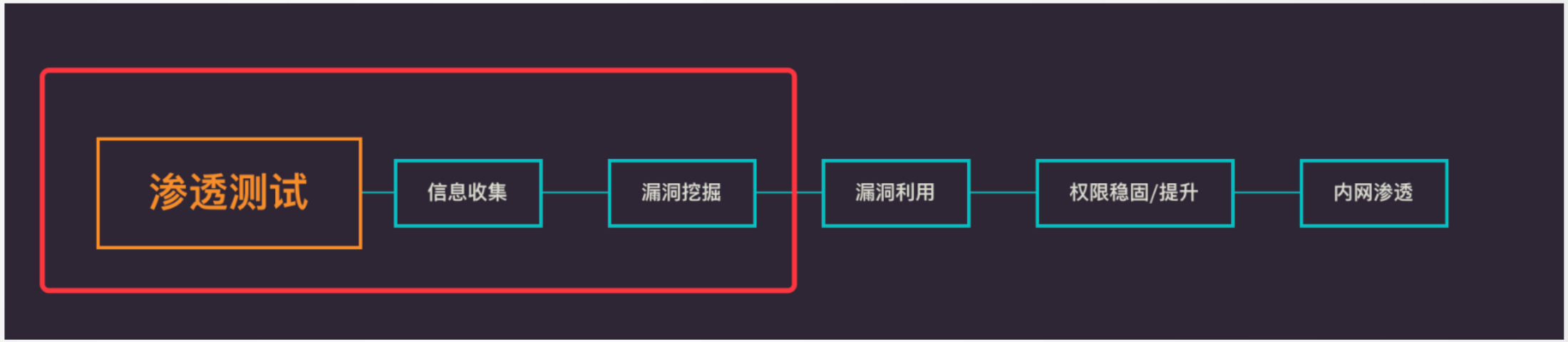
1. 渗透测试
2. 信息收集
3. 利用搜索引擎
4. 真实IP地址发现
5. 代码审计

什么是渗透测试

- 渗透测试是一种合法的、经过授权的，定位某企业网络及应用系统，并对其成功实施漏洞扫描或攻击，找出企业网络和系统的安全薄弱环节
- 渗透测试是一种利用模拟黑客攻击的方式，来评估计算机网络系统安全性能的方法。评估计算机网络系统安全性能的方法。

渗透测试流程

- 建议安全管理人员能从一名攻击者的角度进行分析
- 针对目标站点渗透攻击的第一步，攻击者会尽可能收集目标各方面的信息
- 作为安全管理人员，应考虑如何从用户视角下的所见范围进行探测，避免攻击者拿到敏感信息



信息收集的方式

- 可以分为两种：主动和被动
- 主动的信息收集方式
  - 通过直接访问、扫描网站，这种将流量流经网站的行为。
- 被动的信息收集方式
  - 利用第三方暴露在外，主要是互联网的信息进行收集。

域名信息

- 在攻击者的视角下，容易获得的第一个重要信息为目标域名
- 可以利用域名访问目标网站，获取IP地址
- 域名均由域名提供商对公众开放
- 容易被忽略的whois信息whois（读作“Who is”，非缩写）是用来查询 域名 的IP以及 所有者 等信息的 传输协议
- 在线相关工具：
  1. http://tool.chinaz.com/
  2. http://site.ip138.com/
  3. https://dns.aizhan.com/

Fierce工具

- Fierce是一款IP、域名互查的DNS工具，可进行域传送漏洞检测、字典爆破子域名、反查IP段、反查指定域名上下一段IP，属于一款半轻量级的多线程信息收集用具。Fierce可尝试建立HTTP连接以确定子域名是否存在，此功能为非轻量级功能，所以，定义为半轻量级。
- 域名枚举：在得到主域名信息之后，如果能通过主域名得到所有子域名信息，再通过子域名查询其对应的主机IP，这样我们能得到一个较为完整的信息。

- **使用fierse工具，可以进行域名列表查询：fierce -dns domainName**
  - fierce -dns ziroom.com
  - fierce依次获取指定域的DNS服务器、检查DNS区域传送漏洞
- **检查是否有泛域名解析、用字典爆破子域名**
  - 通过wordlist指定字典
  - 字典文件内为域名的前缀，如：admin.ziroom.com的前缀为admin
  - root@kali:/tmp# fierce -dns ziroom.com -wordlist subziroom.txt

反查指定域名附近的IP段

- 通过traverse来扫描指定域名的IP范围
- root@kali:/# fierce -dns ziroom.com -traverse
- 在线子域名查询
  - https://phpinfo.me/domain/

利用搜索引擎发现敏感信息

- 利用搜索引擎发现网站相关信息，可以减少与站点产生交互
- 建议使用多个搜索引擎，使用高级功能
- 不同搜索引擎的基本语法有些区别，大致相同
- 常用操作符
  - inurl
  - title
  - filetype
  - site
  - cache

综合利用搜索引擎

- 使用缓存匿名浏览
  - cache:xxx.com/xxx.html
- 获取特殊信息
  - site:xxx.cn filetype:sql
- 后台管理页面、目录列表、特殊页面
  - intitle: index of

专用搜索

- **GitHub**
  - 开发者利用Github进行代码开源和协同工作，里面可能存在敏感内容
- **网页存档，搜索历史页面和文件**
  - http://web.archive.org/
- **各大云平台**
  - http://www.daysou.com/
  - http://so.baiduyun.me
- **Web网站、特定设备的搜索攻击**
  - ZoomEye,FOFA.so, shodanhq.com

真实IP地址发现

- 一些大型网站，为了提高用户的访问速度，使用了CDN（内容分发网络）技术
  - CDN网络在全国各地做内容缓存分发
  - 用户从最近的缓存上获取数据
- 用户直接访问的域名对应的IP地址为CDN地址，非目标Web站点的真实地址
- 获取真实IP地址的目的在于明确真实目标
- 如果没有获得真实地址，那么渗透成功的可能性就非常小了

常用的方法

- **查找分站的IP**
  - CDN为付费服务，有些网站仅主站配置，分站并不配置。找到分站，再扫描该IP段的80端口
- **尝试服务器主动发起联系**
  - 例如，服务器发送验证码邮件，但目前也有很多利用第三方验证邮件发送平台，则无法验证。
- **Ping xxx.com**
  - www.baidu.com 和 baidu.com有区别
- **Phpinfo**
  - 有些网站存放了phpinfo(),可以尝试爆破

其他的方法

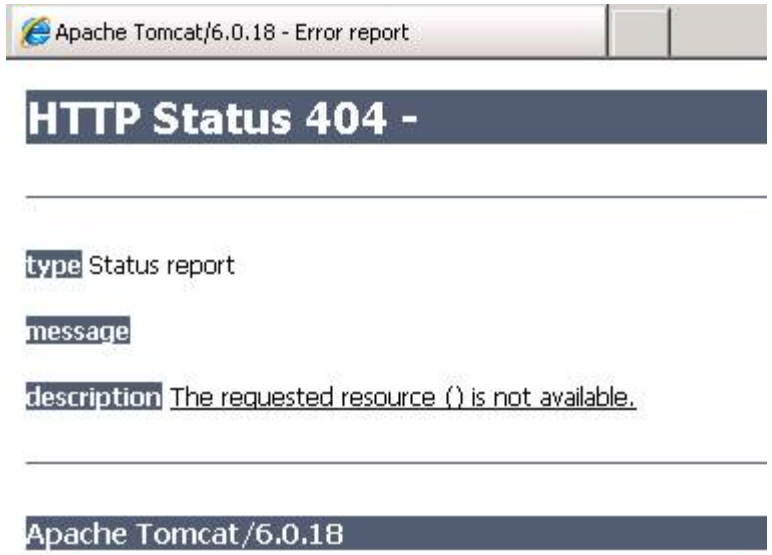
- XSS
  - 如果目标存在存储型XSS攻击漏洞，可以利用漏洞，让管理员打开特殊构造的页面，使用Java Applet接口进行IP查找
  - 条件较为苛刻，仅技术上探讨
- 全网扫描
  - 找到80端口主机，找到host为www.xxx.com的IP，再扫一次抓取特征，去除错误页面
  - 对带宽占用非常大
- CDN服务商
  - 想办法进入CDN服务商的后台

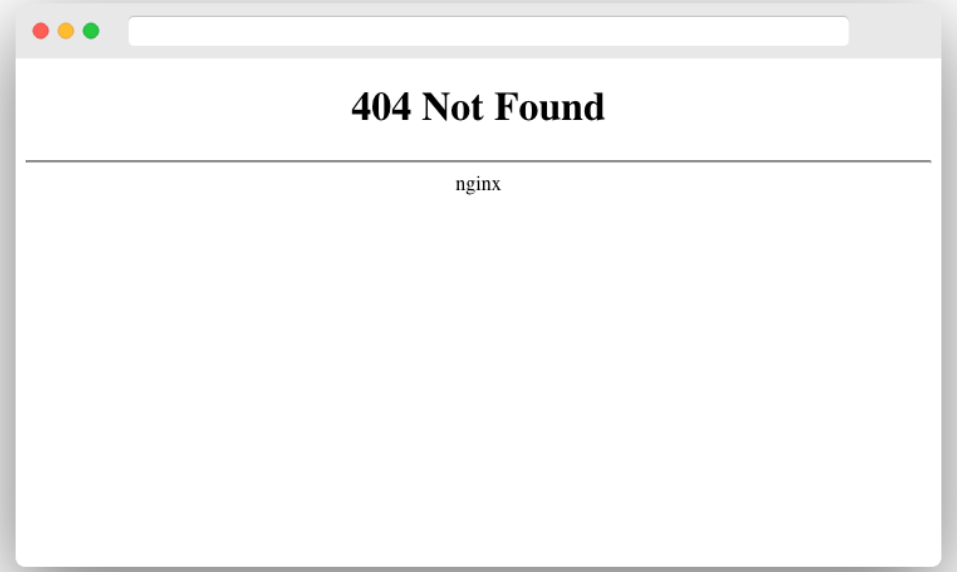
探测目标端口开放情况

- 指TCP/IP协议支持的0-65535端口
  - HTTP默认80
  - HTTPS默认443
  - 也有很多攻击，故意修改端口
- 常利用NMAP进行端口扫描
  - 获取远程主机的系统类型及开放端口
  - 列出指定端口的主机列表
  - 在局域网里寻找所有在线主机

目标版本特征发现

- 中间件版本对于攻击者来说非常重要
- 会利用非正常手段导致目标报错，诱骗目标返回错误页面
- 建议隐藏这些版本信息，修改默认的错误页面





常用的防护方案

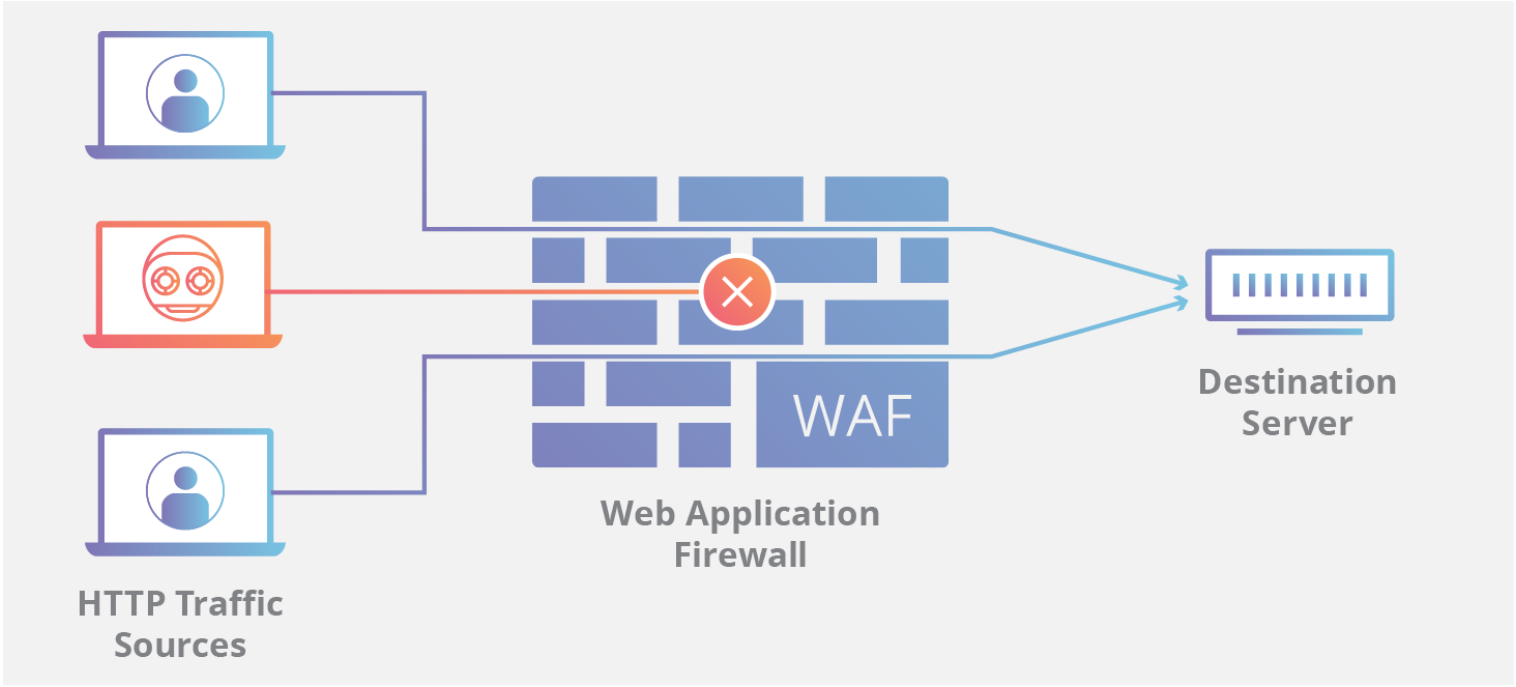
- **直观想法是尽可能消灭漏洞，但无法保证，在设计方案时，必须考虑“适度防护”**
  - 适度防护：建立防护手段，使攻击者的攻击代价（时间、经济成本等）大于攻击后的价值，这样可以使攻击者放弃攻击目标
- **整体防护思路：识别攻击者能连接过来的路径及攻击者可看到的信息，包括：**
  - 目标的端口号
  - 目标中间件及服务的版本
  - 是否有明显漏洞等信息

简单的防护方案

- 关闭或修改服务器开放的端口
- 隐藏Web服务器的banner
- 利用防护类工具
  - 购买软件、硬件、服务等
  - 中小用户，可以考虑在线云WAF进行防护
  - 一定要根据Web站点的特点进行修改
- 采用成熟的CMS系统
  - 成熟的CMS系统在安全性上比很多不知名的CMS系统要好
  - 付费也是值得的

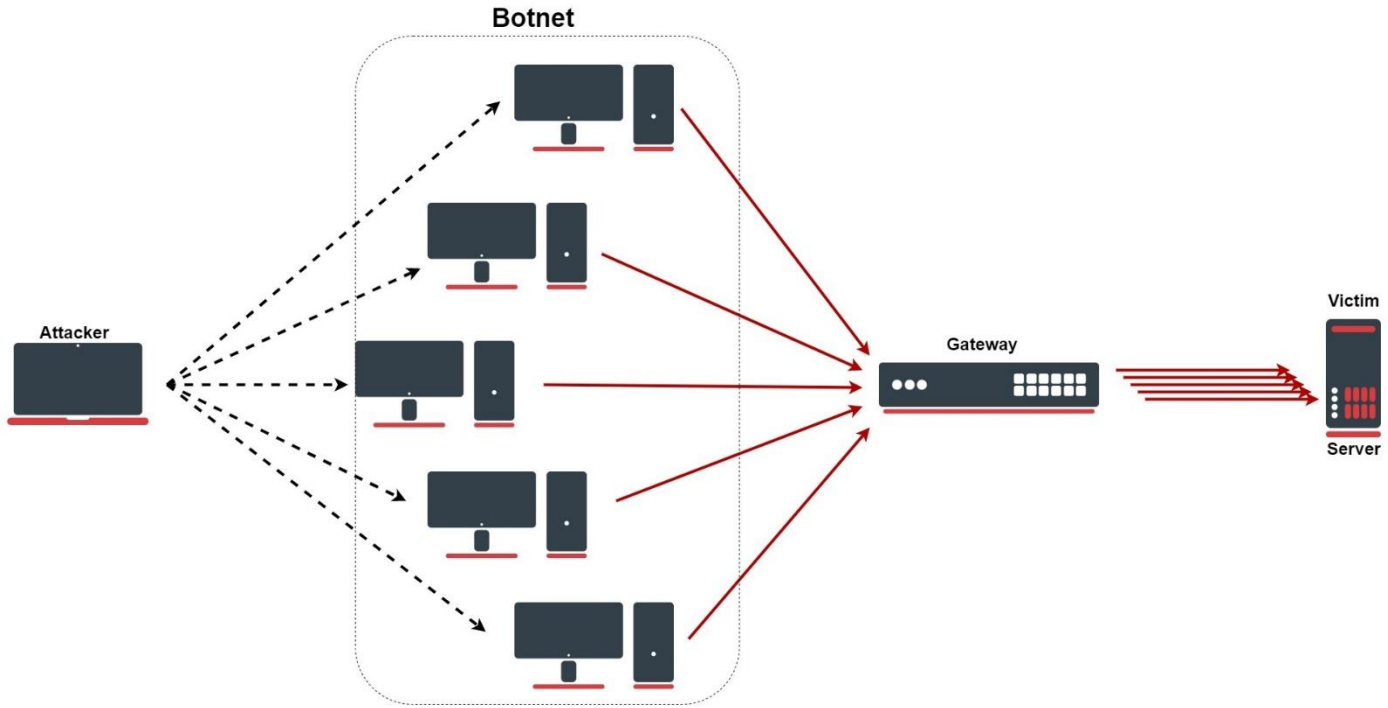
WAF

❖Web应用防护系统（也称为：网站应用级入侵防御系统。英文：Web Application Firewall，简称： WAF）。利用国际上公认的一种说法：Web应用防火墙是通过执行一系列针对HTTP/HTTPS的安全策略来专门为Web应用提供保护的一款产品



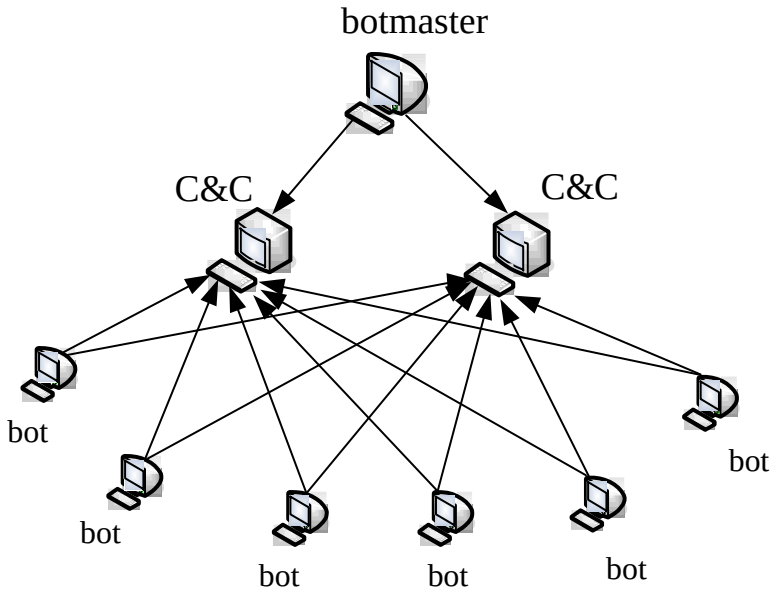
DDoS攻击及防范方法

- 分布式拒绝服务攻击（Distributed Denial of Service）依然是目前Web的主要威胁

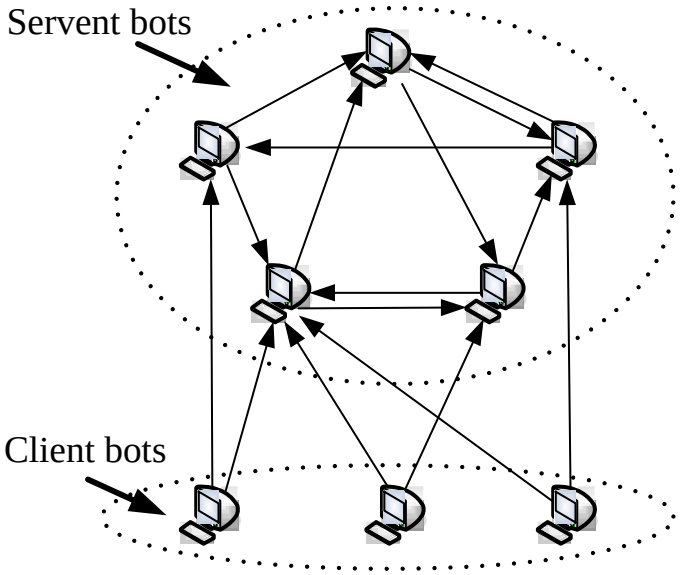


僵尸网络的分类

- 按控制与命令机制分类，僵尸网络可分为集中式僵尸网络和分布式僵尸网络两种，以下是这两种僵尸网络的结构图：



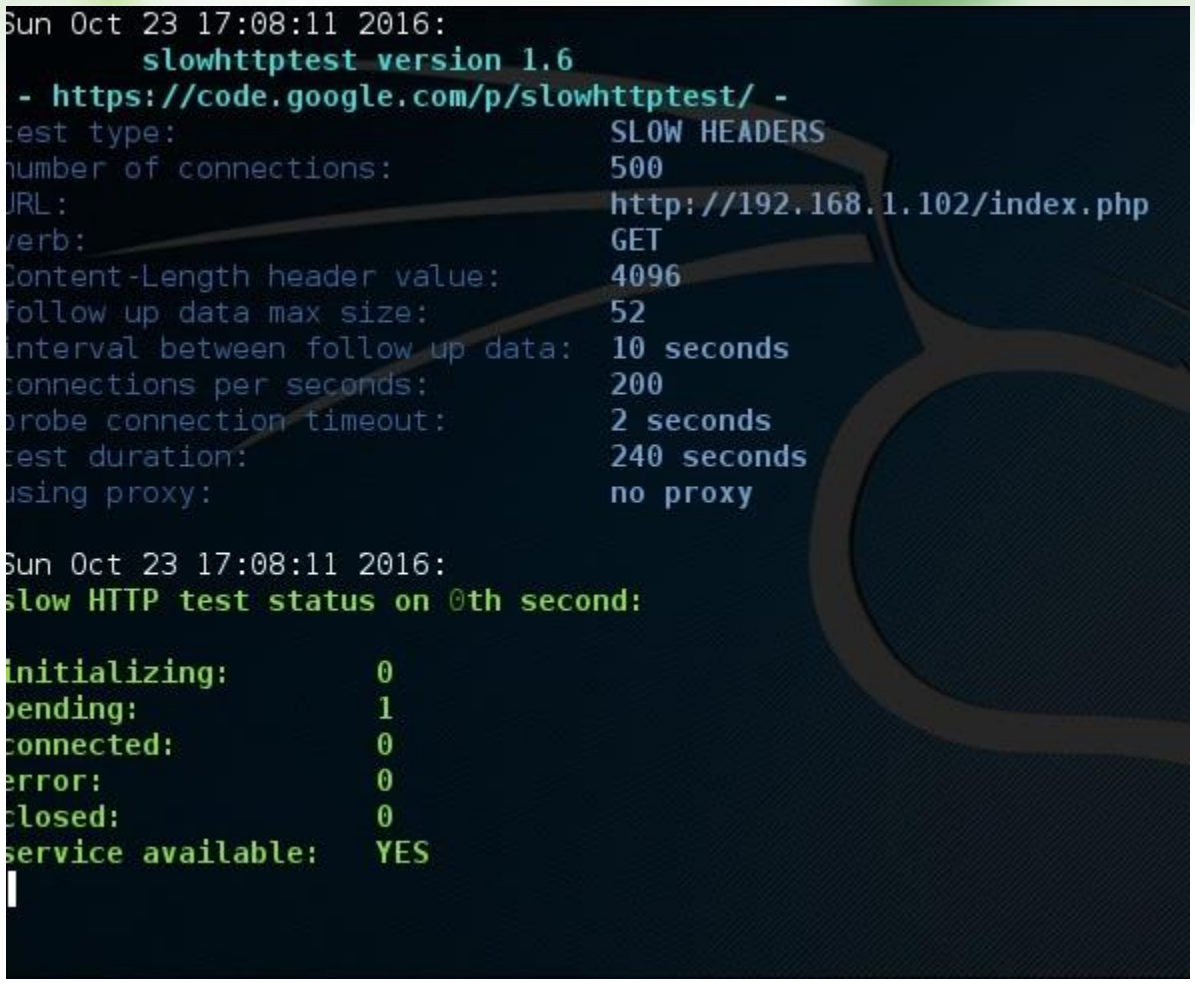
(1) 集中式僵尸网络



(2) 分布式僵尸网络

慢速连接攻击

- Slowloris
- Slow HTTP POST
- Slow Read Attack



攻击包的结构特点

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	154.49.2.126	119.162.193.187	TCP	74	52074→80 [SYN] Seq=0 Win=28400 Len=0 MSS=1420 SACK_PERM=1 TSval=4309376 TSecr=0 WS=128
2	0.006674	119.162.193.187	154.49.2.126	TCP	74	80→52074 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=96939986 TSecr=4309376 WS=512
3	0.006742	154.49.2.126	119.162.193.187	TCP	66	52074→80 [ACK] Seq=1 Ack=1 Win=28416 Len=0 TSval=4309377 TSecr=96939986
4	0.022204	154.49.2.126	119.162.193.187	TCP	305	[TCP segment of a reassembled PDU]
7	0.027801	119.162.193.187	154.49.2.126	TCP	66	80→52074 [ACK] Seq=1 Ack=240 Win=30208 Len=0 TSval=96939992 TSecr=4309381
8	0.335573	154.49.2.126	119.162.193.187	TCP	74	[TCP segment of a reassembled PDU]
6	0.341096	119.162.193.187	154.49.2.126	TCP	66	80→52074 [ACK] Seq=1 Ack=248 Win=30208 Len=0 TSval=96940070 TSecr=4309459
1	100.336657	154.49.2.126	119.162.193.187	TCP	74	[TCP segment of a reassembled PDU]
7	100.343301	119.162.193.187	154.49.2.126	TCP	66	80→52074 [ACK] Seq=1 Ack=256 Win=30208 Len=0 TSval=96965071 TSecr=4334460
1	200.338008	154.49.2.126	119.162.193.187	TCP	74	[TCP segment of a reassembled PDU]
0	200.344576	119.162.193.187	154.49.2.126	TCP	66	80→52074 [ACK] Seq=1 Ack=264 Win=30208 Len=0 TSval=96990071 TSecr=4359460
1	300.338787	154.49.2.126	119.162.193.187	TCP	74	[TCP segment of a reassembled PDU]
9	300.345158	119.162.193.187	154.49.2.126	TCP	66	80→52074 [ACK] Seq=1 Ack=272 Win=30208 Len=0 TSval=97015071 TSecr=4384460
0	387.549020	154.49.2.126	119.162.193.187	TCP	66	52074→80 [FIN, ACK] Seq=272 Ack=1 Win=28416 Len=0 TSval=4406263 TSecr=97015071
9	387.571788	119.162.193.187	154.49.2.126	TCP	66	80→52074 [FIN, ACK] Seq=1 Ack=273 Win=30208 Len=0 TSval=97036878 TSecr=4406263
0	387.571796	154.49.2.126	119.162.193.187	TCP	66	52074→80 [ACK] Seq=273 Ack=2 Win=28416 Len=0 TSval=4406269 TSecr=97036878

```
▸ .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
  Window size value: 222
```

```
000  42 01 0a f0 00 01 42 01  0a f0 00 14 08 00 45 00   B.....B. ......E.
010  01 23 19 68 40 00 40 06  4a 60 9a 31 02 7e 77 a2   .#.h@.@. J`.1.~w.
020  c1 bb cb 6a 00 50 1f c3  d8 e7 0d 8a d6 e2 80 18   ...j.P.. ........
030  00 de 9f 47 00 00 01 01  08 0a 00 41 c1 85 05 c7   ...G.... ...A....
040  2f d2 47 45 54 20 2f 3f  36 37 31 32 38 30 31 30   /.GET /? 67128010
050  38 39 33 31 35 36 20 48  54 54 50 2f 31 2e 31 0d   893156 H TTP/1.1.
060  0a 48 6f 73 74 3a 20 62  62 63 2e 63 6f 6d 0d 0a   .Host: b bc.com..
070  55 73 65 72 2d 41 67 65  6e 74 3a 20 4d 6f 7a 69   User-Age nt: Mozi
080  6c 6c 61 2f 34 2e 30 20  28 63 6f 6d 70 61 74 69   lla/4.0  (compati
090  62 6c 65 3b 20 4d 53 49  45 20 37 2e 30 3b 20 57   ble; MSI E 7.0; W
0a0  69 6e 64 6f 77 73 20 4e  54 20 35 2e 31 3b 20 54   indows N T 5.1; T
0b0  72 69 64 65 6e 74 2f 34  2e 30 3b 20 2e 4e 45 54   rident/4 .0; .NET
0c0  20 43 4c 52 20 31 2e 31  2e 34 33 32 32 3b 20 2e    CLR 1.1 .4322; .
0d0  4e 45 54 20 43 4c 52 20  32 2e 30 2e 35 30 33 6c   NET CLR  2.0.503l
0e0  33 3b 20 2e 4e 45 54 20  43 4c 52 20 33 2e 30 2e   3; .NET  CLR 3.0.
0f0  34 35 30 36 2e 32 31 35  32 3b 20 2e 4e 45 54 20   4506.215 2; .NET
100  43 4c 52 20 33 2e 35 2e  33 30 37 32 39 3b 20 4d   CLR 3.5. 30729; M
110  53 4f 66 66 69 63 65 20  31 32 29 0d 0a 43 6f 6e   SOffice  12)..Con
120  74 65 6e 74 2d 4c 65 6e  67 74 68 3a 20 34 32 0d   tent-Len gth: 42.
130  0a                                                 .
```

代码审计

- ❖渗透测试从“黑盒”角度进行分析和测试，代码审计从“白盒”角度进行检测
  - 对源码进行检查
  - Web系统由于功能复杂，代码量大，人工审计工作量很大
- ❖只需关注具有用户交互功能的业务点及系统的业务流程
  - 整体业务流程分析及绘制
  - 重点业务流程及相同项目归类
  - 参数含义确定
  - 根据业务流程展开代码审计

代码审计的要求

- 具有良好的代码阅读能力
  - 看懂代码的逻辑
  - 函数没见过没有关系，可以网上搜索
- 有漏洞分析经验
  - 熟悉XSS、SQL注入、文件上传等攻击
- 在业务流程方面具有一定的经验

代码审计环境准备

- 不同环境会影响漏洞的利用
  - 建议Linux和Windows系统下的PHP环境都搭建一套
  - 需要多个PHP版本
- phpStudy集成环境适合于版本切换
  - Apache+Nginx+LightTPD+PHP+MySQL+phpMyAdmin+Zend

代码审计工具

- 代码编辑器
  - Notepad++
  - Sublime
  - UltraEdit
  - Zend Studio
- 代码审计工具
  - Seay源代码审计系统
  - Fortify SCA
  - RIPS
- 漏洞验证辅助
  - Burp Suite
  - 浏览器扩展
  - 编码转换及加解密工具
  - 正则调试工具
  - SQL执行监控工具

通用代码审计思路

- **敏感函数回溯参数过程**
  - 目前使用得最多的一种方式
  - 优点：只需要搜索相应敏感关键字，快速、高效
  - 缺点：没有通读代码，覆盖不全面，定位不容易
- **非函数使用不当的漏洞，也有特征**
  - 例如SQL注入，结合Select、From、Where等关键字
- **通读全文代码**
  - 也有技巧，首先看代码结构、模块目录、核心文件
  - 函数集文件：包含functions、common等
  - 配置文件：包含config等
  - 安全过滤文件：filter、safe、check
  - Index文件

根据功能点定向审计

- **文件上传功能**
  - 文章编辑、资料编辑、头像上传、附件上传
- **文件管理功能**
  - 如果程序将文件名或路径直接在参数中传递
- **登录认证功能**
  - 认证方式大多是基于Cookie和Session，如果Cookie信息没有加盐加密，可能导致任意用户登录漏洞
- **找回密码功能**
  - 重置管理员密码，最常见的是验证码爆破
  - APP应用，验证码很多是4位，且没有限制错误次数和时间