

在之前的一系列文章中，我们主要讲了内网渗透的一些知识，而在现实中，要进行内网渗透，一个很重要的前提便是：你得能进入内网啊！所以，从这篇文章开始，我们将开启Web渗透的学习（内网渗透系列还会继续长期更新哦）。

渗透测试，是渗透测试工程师完全模拟黑客可能使用的攻击技术和漏洞发现技术，对目标网络、主机、应用的安全作深入的探测，帮助企业挖掘出正常业务流程中的安全缺陷和漏洞，助力企业先于黑客发现安全风险，防患于未然。

Web应用的渗透测试流程主要分为3个阶段，分别是：信息收集→漏洞发现→漏洞利用。本文我们将对信息收集这一环节做一个基本的讲解。

信息收集介绍

进行web渗透测试之前，最重要的一步那就是就是信息收集了，俗话说“渗透的本质也就是信息收集”，信息收集的深度，直接关系到渗透测试的成败。打好信息收集这一基础可以让测试者选择合适和准确的渗透测试攻击方式，缩短渗透测试的时间。一般来说收集的信息越多越好，通常包括以下几个部分：

域名信息收集

子域名信息收集

站点信息收集

敏感信息收集

服务器信息收集

端口信息收集

真实IP地址识别

社会工程学

下面我们对这几种信息收集分别做相应的讲解。

域名信息收集

域名（英语：Domain Name），又称网域，是由一串用点分隔的名字组成的Internet上某一台计算机或计算机组的名称，用于在数据传输时对计算机的定位标识（有时也指地理位置）。由于IP地址具有不方便记忆并且不能显示地址组织的名称和性质等缺点，人们设计出了域名，并通过网域名称系统来将域名和IP地址相互映射，使人更方便地访问互联网，而不用去记住能够被机器直接读取的IP地址数串。

顶级域名/一级域名：

顶级域（或顶级域名，也称为一级域名），是互联网DNS等级之中的最高级的域，它保存于DNS根域的名字空间中。顶级域名是域名的最后一个部分，即是域名最后一点之后的字母，例如在<http://www.example.com>这个域名中，顶级域是`**.com**`。

二级域名：

除了顶级域名，还有二级域名，就是最靠近顶级域名左侧的字段。例如在<http://www.example.com>这个域名中，example就是二级域名。

子域名：

子域名（或子域；英语：Subdomain）是在域名系统等级中，属于更高一层域的域。比如，mail.example.com和calendar.example.com是example.com的两个子域，而example.com则是顶级域.com的子域。凡顶级域名前加前缀的都是该顶级域名的子域名，而子域名根据技术的多少分为二级子域名，三级子域名以及多级子域名。

一般来说，在做渗透测试之前，渗透测试人员能够了解到的信息有限，一般也就只知道一个域名，这就需要渗透测试人员首先要针对一个仅有的域名进行信息搜集，获取域名的注册信息，包括该域名的DNS服务器信息、子域信息和注册人的联系信息等信息。可以用以下几种方法来收集域名信息。

Whois 查询

whois 是用来查询域名的IP以及所有者等信息的传输协议。简单说，whois就是一个用来查询域名是否已经被注册，以及注册域名的详细信息的数据库（如域名所有人、域名注册商），不同域名后缀的Whois信息需要到不同的Whois数据库查询。通过whois来实现对域名信息的查询，可以得到注册人的姓名和邮箱信息通常对测试个人站点非常有用，因为我们可以通过搜索引擎和社交网络挖掘出域名所有人的很多信息。

（1）在线查询

如今网上出现了一些网页接口简化的线上查询工具，可以一次向不同的数据库查询。网页接口的查询工具仍然依赖whois协议向服务器发送查询请求，命令列接口的工具仍然被系统管理员广泛使用。whois通常使用TCP协议43端口。每个域名/IP的whois信息由对应的管理机构保存。

常见的网站包括：

Whois站长之家查询：<http://whois.chinaz.com/>

阿里云中国万网查询：<https://whois.aliyun.com/>

Whois Lookup 查找目标网站所有者的信息：<http://whois.domaintools.com/>

Netcraft Site Report 显示目标网站上使用的技术：http://toolbar.netcraft.com/site_report?url=

Robtex DNS 查询显示关于目标网站的全面的DNS信息：<https://www.robtex.com/>

全球Whois查询：<https://www.whois365.com/cn/>

站长工具爱站查询：<https://whois.aizhan.com/>

（2）使用kali中的whois工具查询

在Kali Linux下自带的Whois查询工具，通过命令Whois查询域名信息，只需输入要查询的域名即可

备案信息查询

网站备案信息是根据国家法律法规规定，由网站所有者向国家有关部门申请的备案，是国家信息产业部对网站的一种管理途径，是为了防止在网上从事非法网站经营活动，当然主要是针对国内网站。

在备案查询中我们主要关注的是：单位信息例如名称、备案编号、网站负责人、法人、电子邮箱、联系电话等。

常用的备案信息查询网站有以下几个：

ICP/IP地址/域名信息备案管理系统：<http://beian.miit.gov.cn/publish/query/indexFirst.action>

ICP备案查询网：<http://www.beianbeian.com/>

备案吧吧：<https://www.beian88.com/>

天眼查：<https://www.tianyancha.com/>

The screenshot displays the official ICP/IP Address/Domain Name Information Filing Management System interface. The header includes the system name and navigation links. A search bar at the top contains 'baidu.com'. The results are divided into two sections: 'ICP Filing Entity Information' and 'ICP Filing Website Information'. The first section shows details for Baidu's ICP license, including the license number (京ICP证030173号), approval time (2020-08-05 09:50:20), and the entity name (北京百度网讯科技有限公司). The second section shows website-specific details, including the website name (百度), website ICP license number (京ICP证030173号-1), website address (www.baidu.com), and website domain (baidu.com). A 'Return Query Result' button is located at the bottom.

ICP/IP地址/域名信息备案管理系统					
首页	ICP备案查询	短信核验	违法违规域名查询	通知公告	政策文件
baidu.com					
Q 搜索					
全国咨询电话 010-66411166					
ICP备案主体信息					
备案/许可证号:	京ICP证030173号	审核通过时间:	2020-08-05 09:50:20		
主办单位名称:	北京百度网讯科技有限公司	主办单位性质:	企业		
ICP备案网站信息					
网站名称:	百度	网站备案/许可证号:	京ICP证030173号-1		
网站首页地址:	www.baidu.com	网站域名:	baidu.com		
网站前置审批项:					
返回查询结果					

子域名信息收集

子域名（或子域；英语：Subdomain）是在域名系统等级中，属于更高一层域的域。比如，mail.example.com和calendar.example.com是example.com的两个子域，而example.com则是顶级域.com的子域。凡顶级域名前加前缀的都是该顶级域名的子域名，而子域名根据技术的多少分为二级子域名，三级子域名以及多级子域名。

为什么要收集子域名

子域名枚举可以在测试范围内发现更多的域或子域，这将增大漏洞发现的几率。

有些隐藏的、被忽略的子域上运行的应用程序可能帮助我们发现重大漏洞。

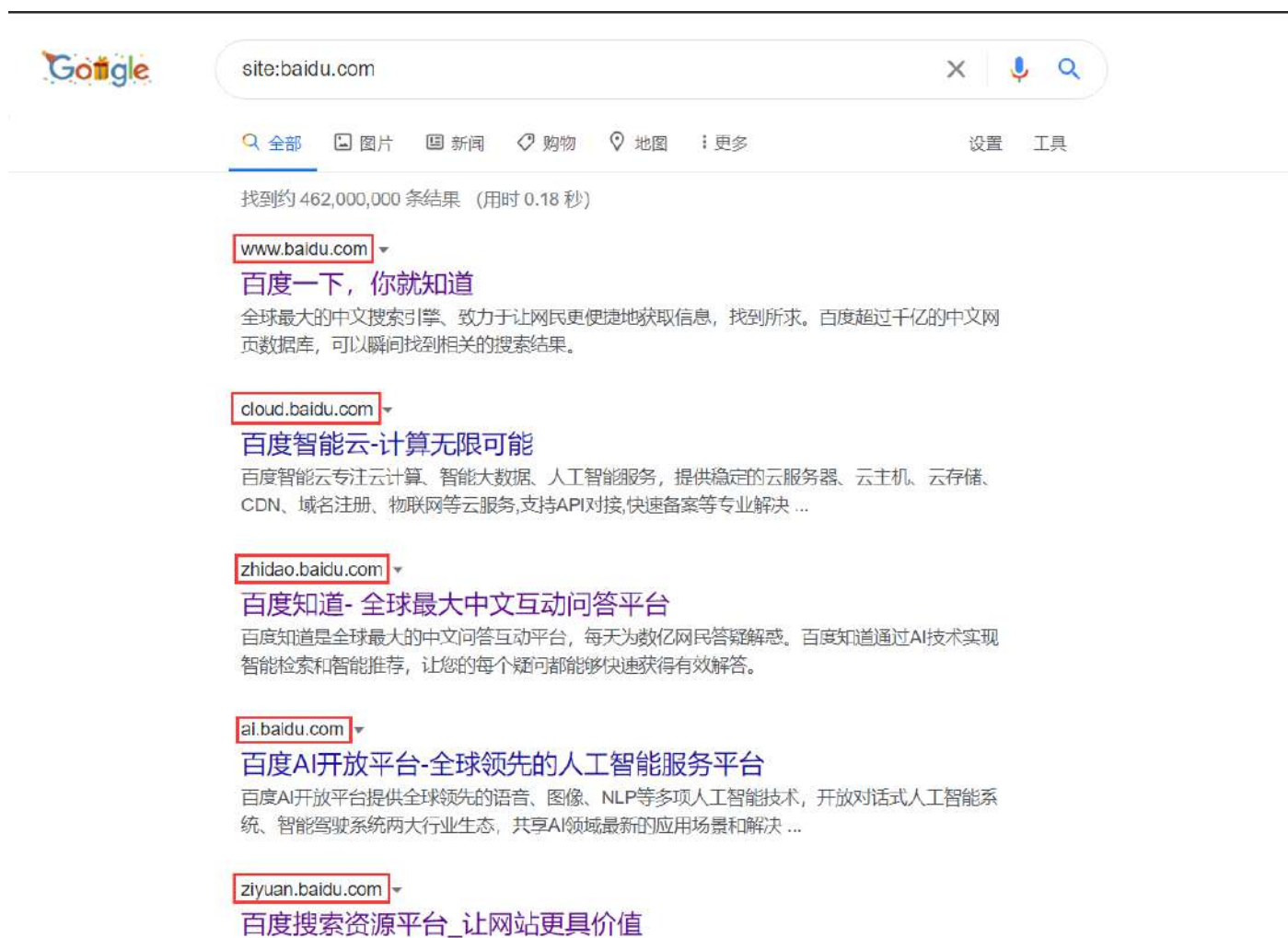
在同一个组织的不同域或应用程序中往往存在相同的漏洞

假设我们的目标网络规模比较大，直接从主域入手显然是很不理智的，因为对于这种规模的目标，一般其主域都是重点防护区域，所以不如先进入目标的某个子域，然后再想办法迂回接近真正的目标，这无疑是个比较好的选择。

收集子域名的方法有以下几种：

利用搜索引擎查询

我们可以利用Google语法搜索子域名，我们以百度的域名为例，使用“site:baidu.com”语法，如下图所示。



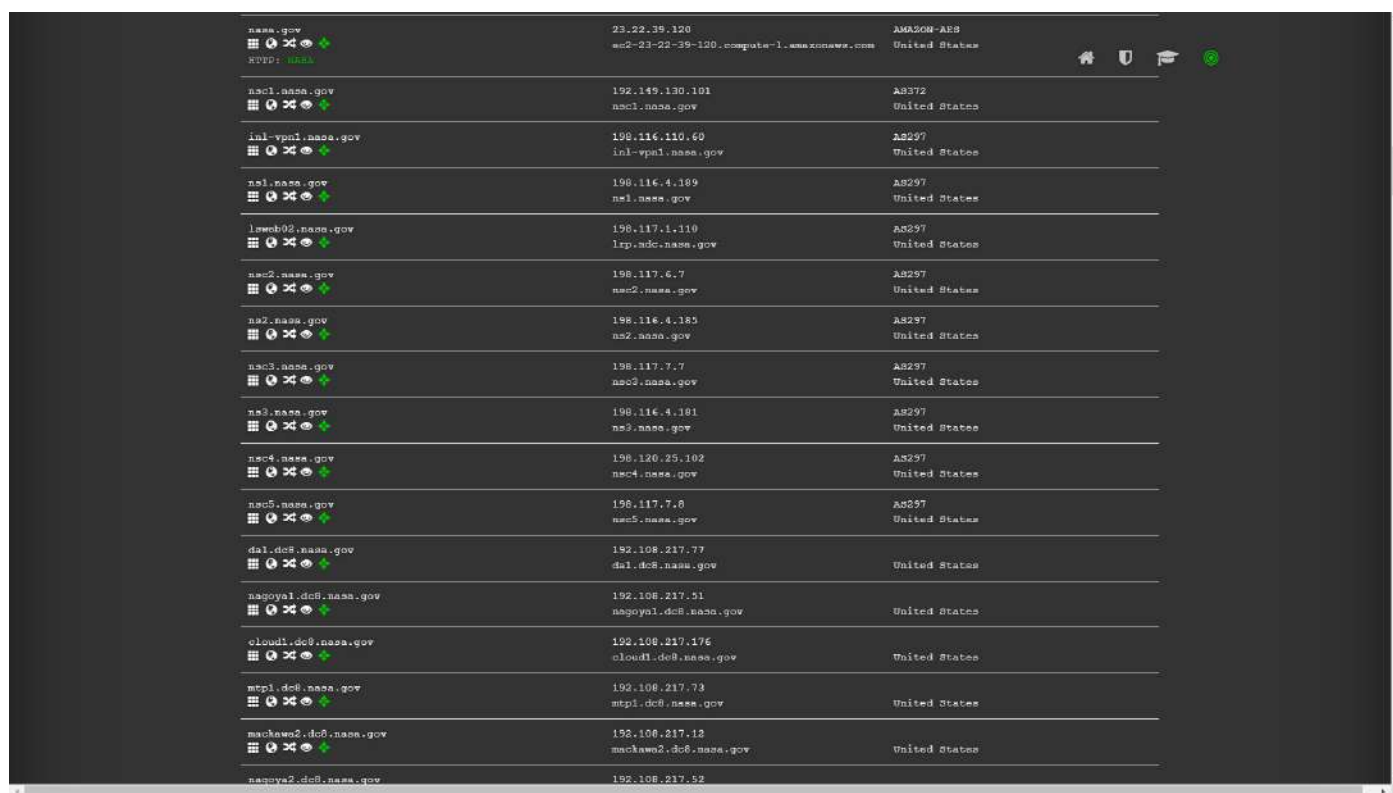
利用在线工具查询

网上有很多子域名的查询站点，可通过它们检索某个给定域名的子域名。如：

- DNSdumpster: <https://dnsdumpster.com/>
- whois反查: <http://whois.chinaz.com/>
- virustotal: www.virustotal.com
- 子域名爆破: <https://phpinfo.me/domain/>

- IP反查绑定域名：<http://dns.aizhan.com/>
- <https://hackertarget.com/find-dns-host-records/>
- <https://site.ip138.com>

我们用DNSdumpster查询nasa的子域名：



Domain	IP Address	Location
nasa.gov	23.22.39.120 ec2-23-22-39-120.compute-1.amazonaws.com	AMAZON-AR5 United States
nsc1.nasa.gov	192.149.130.101 nsc1.nasa.gov	A8372 United States
in1-vpn1.nasa.gov	198.116.110.60 in1-vpn1.nasa.gov	A8297 United States
ns1.nasa.gov	198.116.4.189 ns1.nasa.gov	A8297 United States
lweb02.nasa.gov	198.117.1.110 lwp.nsc.nasa.gov	A8297 United States
nsc2.nasa.gov	198.117.6.7 nsc2.nasa.gov	A8297 United States
ns2.nasa.gov	198.116.4.185 ns2.nasa.gov	A8297 United States
nsc3.nasa.gov	198.117.7.7 nsc3.nasa.gov	A8297 United States
ns3.nasa.gov	198.116.4.181 ns3.nasa.gov	A8297 United States
nsc4.nasa.gov	198.120.25.102 nsc4.nasa.gov	A8297 United States
nsc5.nasa.gov	198.117.7.8 nsc5.nasa.gov	A8297 United States
dal.dcs.nasa.gov	192.108.217.77 dal.dcs.nasa.gov	United States
nagoya1.dcs.nasa.gov	192.108.217.51 nagoya1.dcs.nasa.gov	United States
cloud1.dcs.nasa.gov	192.108.217.176 cloud1.dcs.nasa.gov	United States
mtp1.dcs.nasa.gov	192.108.217.73 mtp1.dcs.nasa.gov	United States
machawo2.dcs.nasa.gov	192.108.217.12 machawo2.dcs.nasa.gov	United States
nagoya2.dcs.nasa.gov	192.108.217.52	

通过证书透明度公开日志枚举子域名

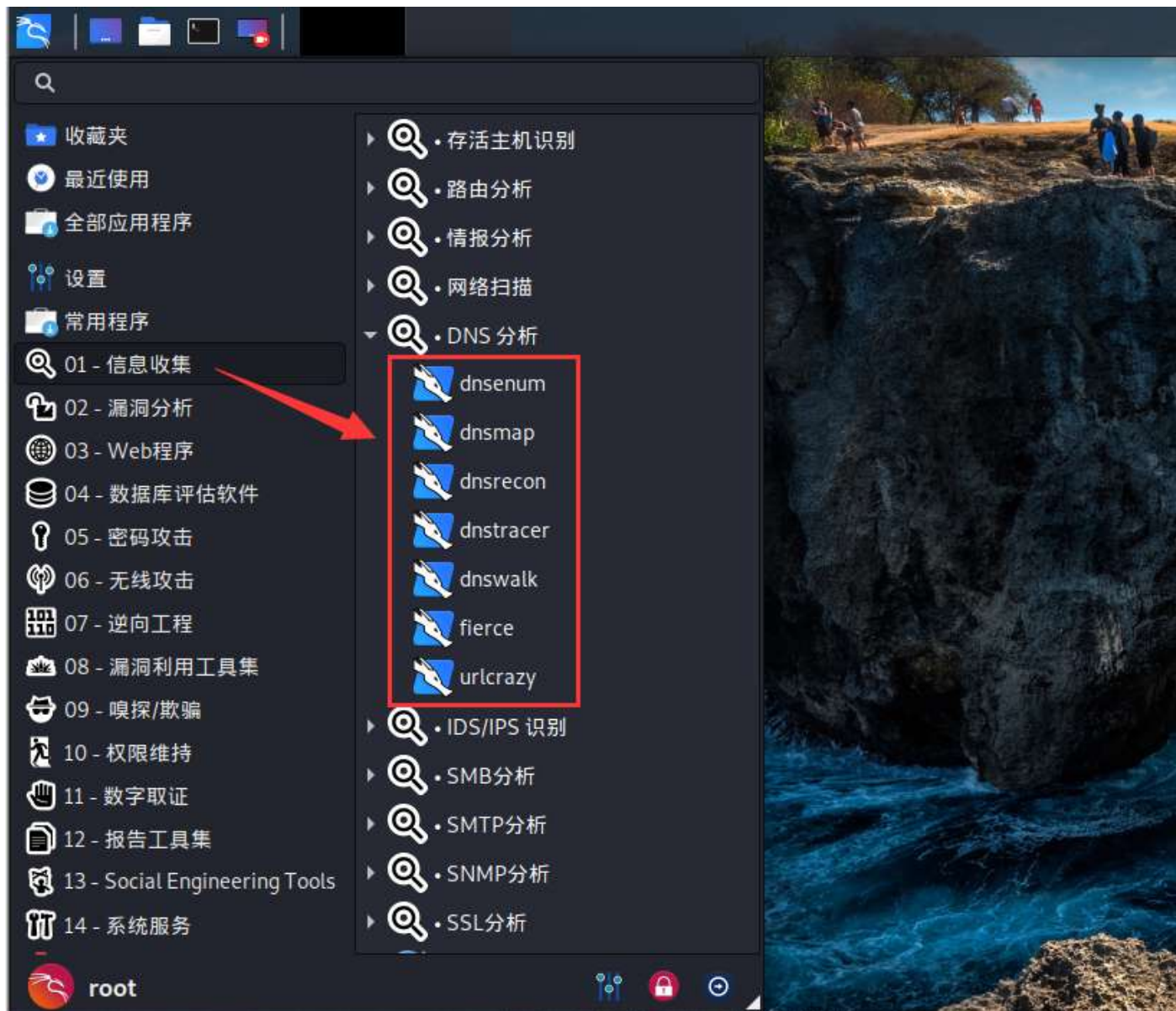
证书透明度是证书授权机构的一个项目，证书授权机构会将每个SSL/TLS证书发布到公共日志中。一个SSL/TLS证书通常包含域名、子域名和邮件地址，这些也经常成为攻击者非常希望获得的有用信息。

查找某个域名所属证书的最简单的方法就是使用搜索引擎来搜索一些公开的CT日志，例如以下网站：

- crt.sh: <https://crt.sh>
- censys: <https://censys.io>

利用工具枚举子域名

kali上的工具



在kali中的信息收集模块的DNS分析中，有很多工具可以进行域名信息收集，如上图。

- **Dnsenum：域名信息收集**
- **Dnsmap：收集信息和枚举DNS信息**
- **Dnsrecon：用于DNS侦察**
- **Fierce：子域名查询**
- **whois查询**

我们可以利用Fierce工具，进行子域名枚举。该工具首先测试是否有域传送漏洞，若存在则应该直接通过域传送搜集子域信息，没有域传送漏洞则采用爆破的方式。

使用方法：

```
fierce -dns <域名>  
fierce -dns <域名> -threads 100 // threads 是线程数，可以自己指定
```

Windows上的工具

Windows上的子域名查询工具主要有：

- Layer子域名挖掘机
- **subDomainsbrute**
- K8
- Sublist3r
- Maltego
-

subDomainsbrute工具可以用于二级域名收集，下载地址：<https://github.com/lijiejie/subDomainsBrute>

Python3环境下运行需要安装aiodns库。使用该工具的命令执行如下：

```
python3 subDomainsBrute.py xxxx.com
```

```
root@kali:~/subDomainsBrute# python3 subDomainsBrute.py nasa.gov  
SubDomainsBrute v1.3 https://github.com/lijiejie/subDomainsBrute  
[+] Validate DNS servers  
[+] Server 114.114.115.115 < OK > Found 4  
[+] 4 DNS Servers found  
[+] Run wildcard test  
[+] Start 6 scan process  
[+] Please wait while scanning ...  
  
All Done. 6 found, 30699 scanned in 126.6 seconds.  
Output file is nasa.gov.txt  
root@kali:~/subDomainsBrute#
```

收集完后，会将收集结果写入一个域名对应的文件中：

```
/root/subDomainsBrute/nasa.gov.txt - Mousepad  
文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H)  
警告：您正在使用 root 账户，操作不当可能会损害您的系统。  
box.nasa.gov 99.84.206.126, 99.84.206.43, 99.84.206.54, 99.84.206.99  
at.nasa.gov 128.158.4.233, 129.166.10.144, 129.166.32.126, 129.166.32.127, 129.166.8.86  
bet.nasa.gov 192.107.192.151  
aero.nasa.gov 198.116.65.32  
ams.nasa.gov 192.68.197.14  
www.ams.nasa.gov 192.68.197.14
```

除了subDomainsbrute工具，Layer子域名挖掘机也是十分强大的，用它收集子域名将详细的显示域名、解析IP、CDN列表、Web服务器和网站状态等信息。

该工具请自行上网搜索下载。

站点信息收集

接下来我们进行web网站站点信息收集，主要收集如下信息：

- CMS指纹识别
- 历史漏洞
- 脚本语言
- 敏感目录/文件
- Waf识别
-

CMS指纹识别

CMS（内容管理系统）又称为整站系统或文章系统，用于网站内容管理。用户只需要下载对应的CMS软件包，就能部署搭建，并直接利用CMS。但是各种CMS都具有其独特的结构命名规则和特定的文件内容，因此可以利用这些内容来获取CMS站点的具体软件CMS与版本。

在渗透测试中，对进行指纹识别是相当有必要的，识别出相应的CMS，才能查找与其相关的漏洞，然后才能进行相应的渗透操作。

常见的CMS有Dedecms(织梦)、Discuz、PHPWEB、PHPWind、PHPCMS、ECShop、Dvbbs、SiteWeaver、ASPCMS、帝国、Z-Blog、WordPress等。

（1）在线识别

如今，网上一些在线的网站查询CMS指纹识别，如下所示：

- BugScanner: <http://whatweb.bugscanner.com/look/>
- 潮汐指纹: <http://finger.tidesec.net/>
- 云悉: <http://www.yunsee.cn/info.html>
- WhatWeb: <https://whatweb.net/>
- 云悉指纹: <http://www.yunsee.cn/finger.html>

如下，我们用WhatWeb: <https://whatweb.net/>在线识别一下我的博客：

（2）利用工具

常见的CMS指纹识别工具有WhatWeb、WebRobo、椰树、御剑Web指纹识别。大禹CMS识别程序等，可以快速识别一些主流CMS。

如下，我们利用kali上的WhatWeb工具识别目标站点的cms：


```
root@kali:~# whatweb www.kingcms.com
/usr/lib/ruby/vendor_ruby/target.rb:188: warning: URI.escape is obsolete
/usr/lib/ruby/vendor_ruby/target.rb:188: warning: URI.escape is obsolete
http://www.kingcms.com [301 Moved Permanently] Country[HONG KONG][HK], HTTPServer[nginx], IP[39.108.101.13], RedirectLocation[https://www.kingcms.com/], Title[301 Moved Permanently], nginx
https://www.kingcms.com/ [200 OK] Country[HONG KONG][HK], HTTPServer[nginx], IP[39.108.101.13], JQuery, PasswordField[userpass], Script[text/javascript], Title[KingCMS内容管理系统 - CMS界的轻骑士], nginx
root@kali:~#
```

如上图，WhatWeb将目标站点的服务器、中间节、cms等都识别了出来。

当我们得知了一个站点的cms类型后，我们可以在网上查找与其相关的漏洞并进行相应的测试。

(3) 手工识别

- \1. 根据HTTP响应头判断，重点关注X-Powered-By、cookie等字段
- \2. 根据HTML 特征，重点关注 body、title、meta等标签的内容和属性。
- \3. 根据特殊的class判断。HTML 中存在特定 class 属性的某些 div 标签，如
-

敏感目录/文件收集

也就是对目标网站做个目录扫描。在web渗透中，探测Web目录结构和隐藏的敏感文件是一个十分重要的环节，从中可以获取网站的后台管理页面、文件上传界面、robots.txt，甚至可能扫描出备份文件从而得到网站的源代码。

常见的网站目录的扫描工具主要有：

- 御剑后台扫描工具
- dirbuster扫描工具
- dirsearch扫描工具
- dirb
- wwwscan
- Spinder.py
- Sensitivefilescan
- Weakfilescan
-

(1) dirsearch目录扫描

下载地址：<https://github.com/maurosoria/dirsearch>

该工具使用很简单，简单使用如下：

```
python3 dirsearch.py -u <URL> -e <EXTENSION>
```

- -u: url (必须)
- -e: 扫描网站需要指定网站的脚本类型，* 为全部类型的脚本 (必须)
- -w: 字典 (可选)
- -t: 线程 (可选)

```
C:\Users\LiuSir\Desktop\Web_Weapon\dirsearch
λ python3 dirsearch.py -u http://47.251.72/ -e *

dirsearch v0.3.9

Extensions: | HTTP method: get | Suffixes: * | HTTP method: get | Threads: 10 | Wordlist size: 6564 | Request count: 6564

Error Log: C:\Users\LiuSir\Desktop\Web_Weapon\dirsearch\logs\errors-20-09-27_20-13-57.log

Target: http://47.251.72/

Output File: C:\Users\LiuSir\Desktop\Web_Weapon\dirsearch\reports\47.251.72\20-09-27_20-13-57

[20:13:57] Starting:
[20:13:58] 403 - 5488 - /.X30/
[20:13:58] 403 - 5488 - /.abusefghjklmnp/
[20:13:58] 403 - 5488 - /.adain/
[20:13:58] 403 - 5488 - /.acnh
[20:13:58] 403 - 5488 - /.acnh
[20:13:58] 403 - 5488 - /.action
[20:13:58] 403 - 5488 - /.actionScript?properties
[20:13:58] 403 - 5488 - /.agignore
[20:13:58] 403 - 5488 - /.agilekeychain.zip
[20:13:58] 403 - 5488 - /.analysis.options
[20:13:58] 403 - 5488 - /.aconf
[20:13:58] 403 - 5488 - /.angular-cli.json
[20:13:58] 403 - 5488 - /.all-contributors
[20:13:58] 403 - 5488 - /.agilekeychain
[20:13:58] 403 - 5488 - /.ansible/
[20:13:58] 403 - 5488 - /.aliases
[20:13:58] 403 - 5488 - /.appert-ignore.xml
[20:13:58] 403 - 5488 - /.approve.yml
[20:13:58] 403 - 5488 - /.atfu_history
[20:13:58] 403 - 5488 - /.architect
[20:13:58] 403 - 5488 - /.autotest
[20:13:58] 403 - 5488 - /.avod
```

(2) DirBuster目录扫描

DirBuster是Owasp(开放Web软体安全项目- Open Web Application Security Project)开发的一款专门用于探测Web服务器的目录和隐藏文件。(需要java环境)

使用如下：

OWASP DirBuster 0.12 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)
http://www.xxx.com/

Work Meth... ☐ Use GET requests only ☒ Auto Switch (HEAD and GET)

Number Of Threads ☐ Go Faster

Select scanning type: ☒ List based brute force ☐ Pure Brute Force

File with list of dirs/files
D:\DirBuster\directory-list-2.3-small.txt

Char set a-zA-Z0-9%20- Min length 1 Max Length 8

Select starting options: ☐ Standard start point ☒ URL Fuzz

☒ Brute Force Dirs ☒ Be Recursive Dir to start with /

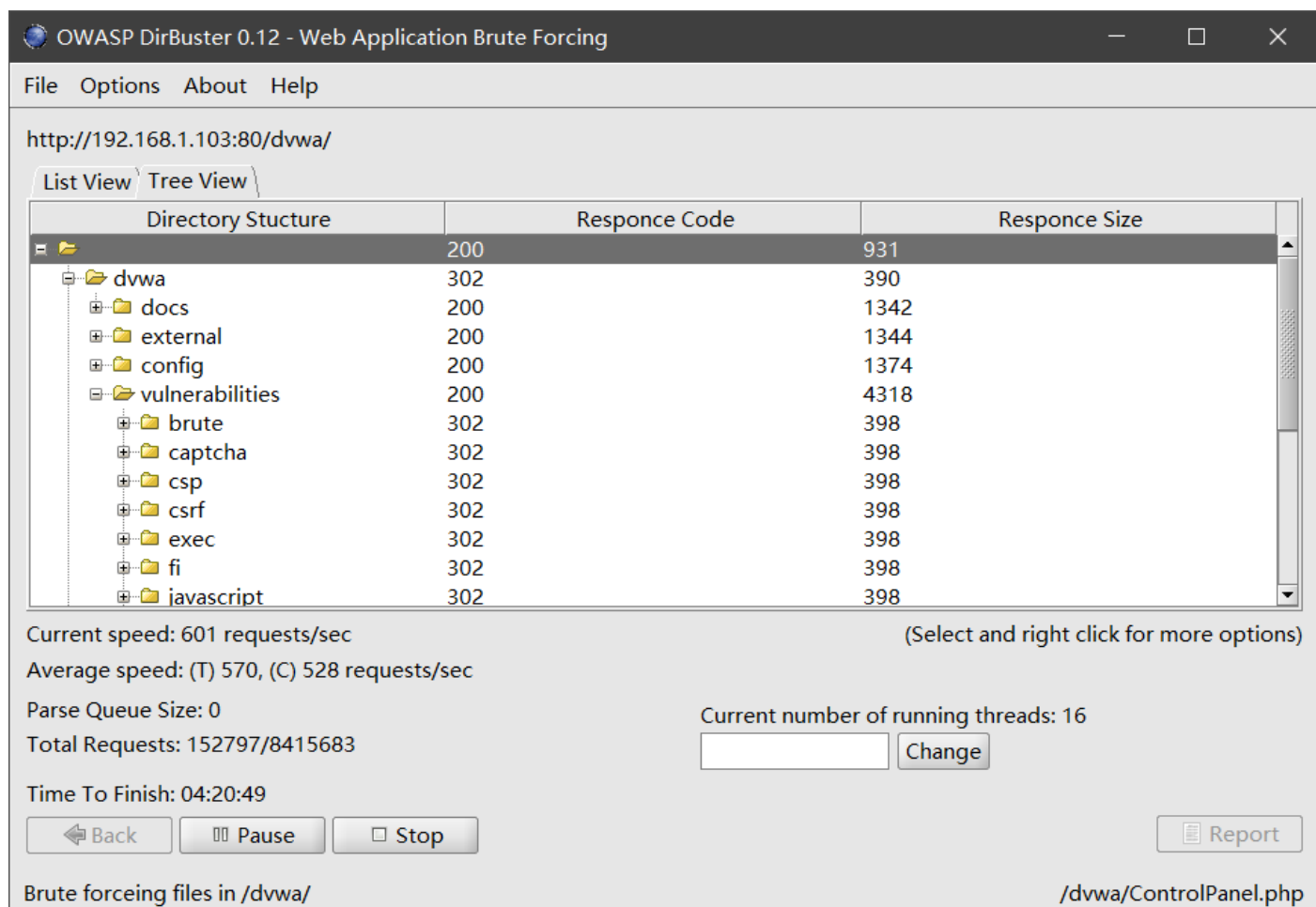
☒ Brute Force Files ☐ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp
/{dir}

Please complete the test details

1. 首先在Target URL输入框中输入要扫描的网址并将扫描过程中的请求方法设置为“Auto Switch(HEAD and GET)”。
2. 自行设置线程（太大了容易造成系统死机哦）
3. 选择扫描类型，如果使用个人字典扫描，则选择“List based bruteforce”选项。
4. 单击“Browse”加载字典。
5. 单击“URL Fuzz”，选择URL模糊测试（不选择该选项则使用标准模式）
6. 在URL to fuzz里输入“/{dir}”。这里的{dir}是一个变量，用来代表字典中的每一行，运行时{dir}会被字典中的目录替换掉。
7. 点击“start”开始扫描

使用DirBuster扫描完成之后，查看扫描结果，这里的显示方式可以选择树状显示，也可以直接列出所有存在的页面：



Waf识别

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。

wafw00f是一个Web应用防火墙（WAF）指纹识别的工具。

下载地址：<https://github.com/EnableSecurity/wafw00f>

wafw00f的工作原理：

1. 发送正常的HTTP请求，然后分析响应，这可以识别出很多WAF。
2. 如果不成功，它会发送一些（可能是恶意的）HTTP请求，使用简单的逻辑推断是哪一个WAF。
3. 如果这也不成功，它会分析之前返回的响应，使用其它简单的算法猜测是否有某个WAF或者安全解决方案响应了我们的攻击。

kali上内置了该工具，wafw00f支持非常多的WAF识别。要查看它能够检测到哪些WAF，请使用-l选项，简单使用如下：

```
wafw00f https://www.xxx.com/
```


intext: 寻找正文中含有关键字的网页
intitle: 寻找标题中含有关键字的网页
allintitle: 用法和intitle类似, 只不过可以指定多个词
inurl: 搜索url中含有关键词的网页
allinurl: 用法和inurl类似, 只不过可以指定多个词
site: 指定访问的站点
filetype: 指定访问的文件类型
link: 指定链接的网页
related: 搜索相似类型的网页
info: 返回站点的指定信息, 例如: info:www.baidu.com 将返回百度的一些信息
phonebook: 电话簿查询美国街道地址和电话号码信息
Index of: 利用 Index of 语法可以发现允许目录浏览的web网站, 就像在本地的普通目录一样

查找网站后台

- intext:后台登录: 将只返回正文中包含“后台登录”的网页
- intitle:后台登录: 将只返回标题中包含“后台登录”的网页



查找指定网站后台

- site:xx.com intext:管理
- site:xx.com inurl:login
- site:xx.com intitle:后台

查看指定网站的文件上传漏洞

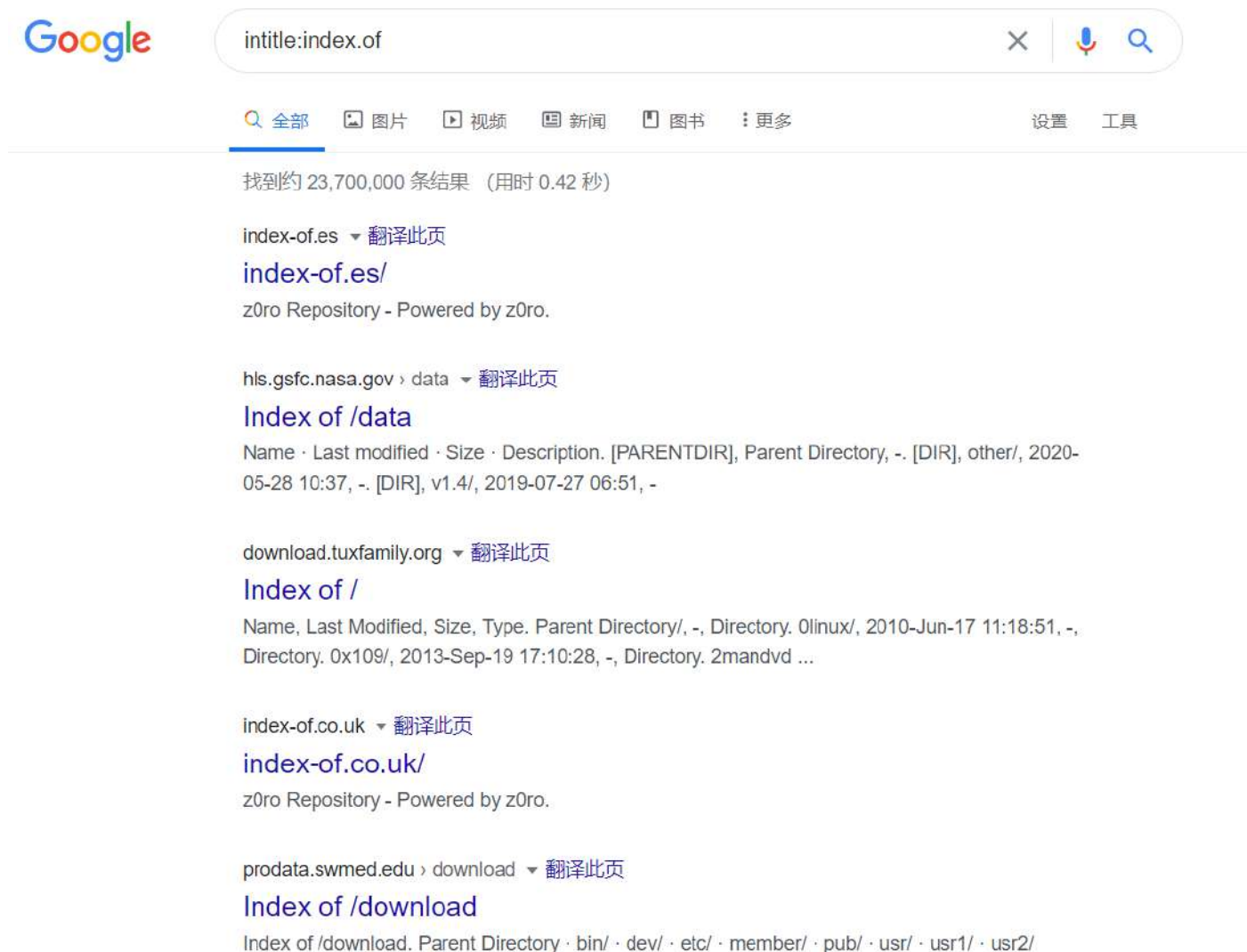
- site:xx.com inurl:file
- site:xx.com inurl:load

利用Index of可以发现允许目录浏览的web网站，就像在本地的普通目录一样

```
index of /admin
index of /passwd
index of /password
index of /mail
"index of /" +passwd
"index of /" +password.txt
"index of /config"
```

用index of目录列表列出存在于一个web服务器上的文件和目录。

intitle:index.of 这里的休止符代表的是单个字母的通配符



随便进去一个看看：

Index of /pub

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 CTAN/	2020-03-31 22:01	-	
 FreeBSD/	2020-07-12 18:28	-	
 GNU/	2020-07-12 18:11	-	
 Linux/	2020-03-30 16:31	-	
 NetBSD/	2020-07-12 21:19	-	
 OpenBSD/	2020-07-12 20:25	-	
 X11/	2020-07-12 06:11	-	
 iris/	2015-02-22 08:10	-	
 lang/	2008-07-07 15:26	-	
 net/	2018-06-14 14:33	-	
 office/	2019-08-14 17:10	-	
 pc/	2013-07-05 15:32	-	
 sagemath/	2018-03-26 19:20	-	
 tex-archive/	2020-03-31 22:01	-	

备份文件泄露

- intitle:index.of index.php.bak
- inurl:index.php.bak
- intitle:index.of [www.zip](#)

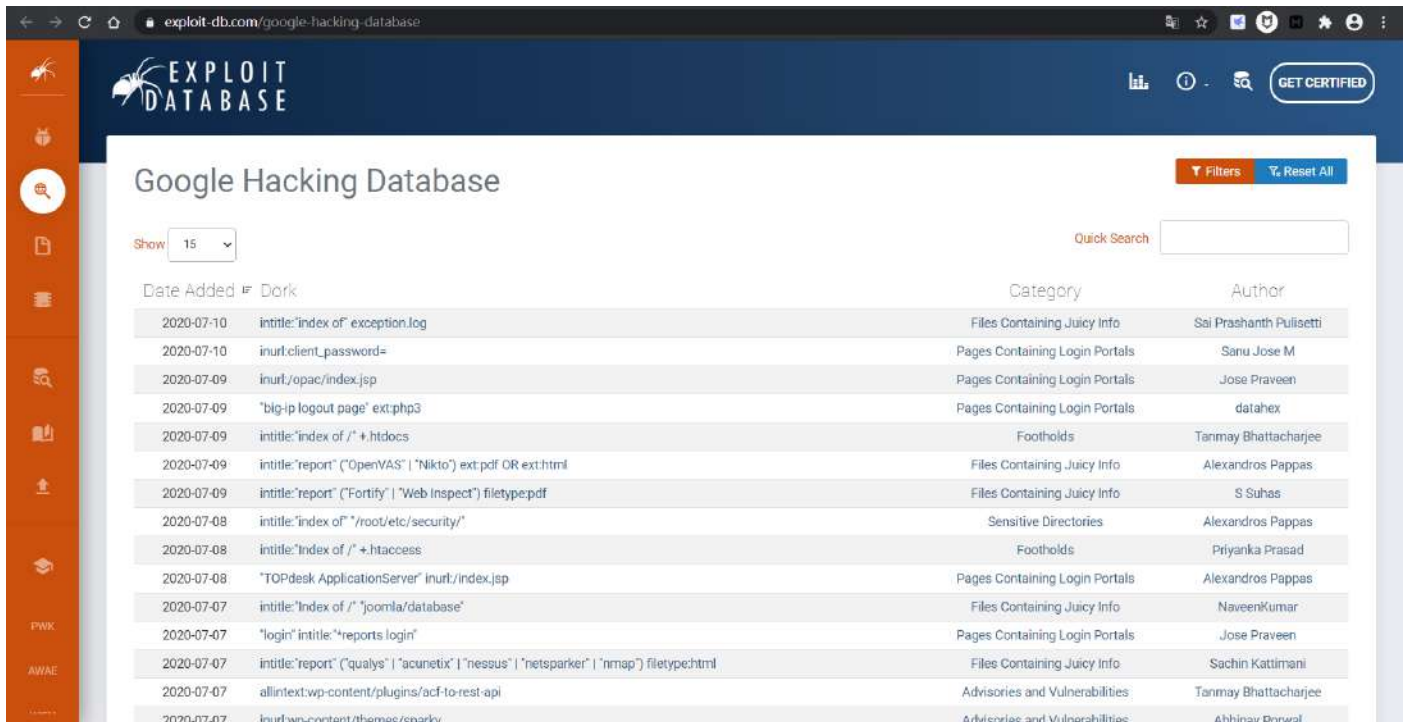
查找sql注入

- inurl:?id=1
- inurl: php?id=

GHDB 谷歌黑客数据库

链接: <https://www.exploit-db.com/google-hacking-database/>

黑客们能通过简单的搜索框在网络中出入于无形，在这背后还有一个强大的后盾，那就是Google Hacking Database (GHDB)



Date Added	Query	Category	Author
2020-07-10	intitle:"index of" exception.log	Files Containing Juicy Info	Sai Prashanth Puliseti
2020-07-10	inurl:client_password=	Pages Containing Login Portals	Sanu Jose M
2020-07-09	inurl:/opac/index.jsp	Pages Containing Login Portals	Jose Praveen
2020-07-09	"big-ip logout page" ext:php3	Pages Containing Login Portals	datahex
2020-07-09	intitle:"index of/" +htdocs	Footholds	Tanmay Bhattacharjee
2020-07-09	intitle:"report" ("OpenVAS" "Nikto") ext:pdf OR ext:html	Files Containing Juicy Info	Alexandros Pappas
2020-07-09	intitle:"report" ("Fortify" "Web Inspect") filetype:pdf	Files Containing Juicy Info	S Suhas
2020-07-08	intitle:"index of" "/root/etc/security/"	Sensitive Directories	Alexandros Pappas
2020-07-08	intitle:"Index of/" +htaccess	Footholds	Priyanka Prasad
2020-07-08	"TOPdesk ApplicationServer" inurl:/index.jsp	Pages Containing Login Portals	Alexandros Pappas
2020-07-07	intitle:"Index of/" "joomla/database"	Files Containing Juicy Info	NaveenKumar
2020-07-07	"login" intitle:"*reports login"	Pages Containing Login Portals	Jose Praveen
2020-07-07	intitle:"report" ("qualys" "acunetix" "nessus" "nessus" "netsparker" "nmap") filetype:html	Files Containing Juicy Info	Sachin Kattimani
2020-07-07	allintext:wp-content/plugins/acf-to-rest-api	Advisories and Vulnerabilities	Tanmay Bhattacharjee
2020-07-07	inurl:wp-content/themes/sparky	Advisories and Vulnerabilities	Abhinav Porwal

这是全世界的黑客朋友们自发维护的一个汇集着各种已经被优化的查询语句的数据库，每天都在不断地更新各种好用有效的Google查询语句。

Github信息泄露

GitHub作为开源代码平台，给程序员提供了很多便利，但如果使用不当，比如将包含了账号密码、密钥等配置文件的代码上传了，导致攻击者能发现并进一步利用这些泄露的信息，就是一个典型的GitHub敏感信息泄露漏洞，再如开发人员在开发时，常常会先把源码提交到github，最后再从远程托管网站把源码pull到服务器的web目录下，如果忘记把.git文件删除，就造成此漏洞。利用.git文件恢复网站的源码，而源码里可能会有数据库的信息，详情参见：https://blog.csdn.net/qq_45521281/article/details/105767428

很多网站及系统都会使用pop3和smtp发送来邮件，不少开发者由于安全意识不足会把相关的配置文件信息也放到Github上，所以如果这时候我们动用一下Google搜索语法，就能把这些敏感信息给找出来了。

```
site:Github.com smtp
site:Github.com smtp @qq.com
site:Github.com smtp @126.com
site:Github.com smtp @163.com
site:Github.com smtp @sina.com.cn
... ..
```

数据库信息泄露：

```
site:Github.com sa password
site:Github.com root password
```

服务器信息收集

我们还需要对目标服务器的信息进行收集，主要包括一下部分：

- Web服务器指纹识别
- 真实IP地址识别
- 编程语言
- Web中间件
- 端口信息收集
- 后端存储技术识别
-

Web服务器指纹识别

Web服务器指纹识别是了解正在运行的web服务器类型和版本，目前市场上存在几种不同的web服务器提供商和软件版本，了解被测试的web服务器的类型，能让测试者更好去测试已知漏洞和大概的利用方法，将会在渗透测试过程中有很大的帮助，甚至会改变测试的路线。

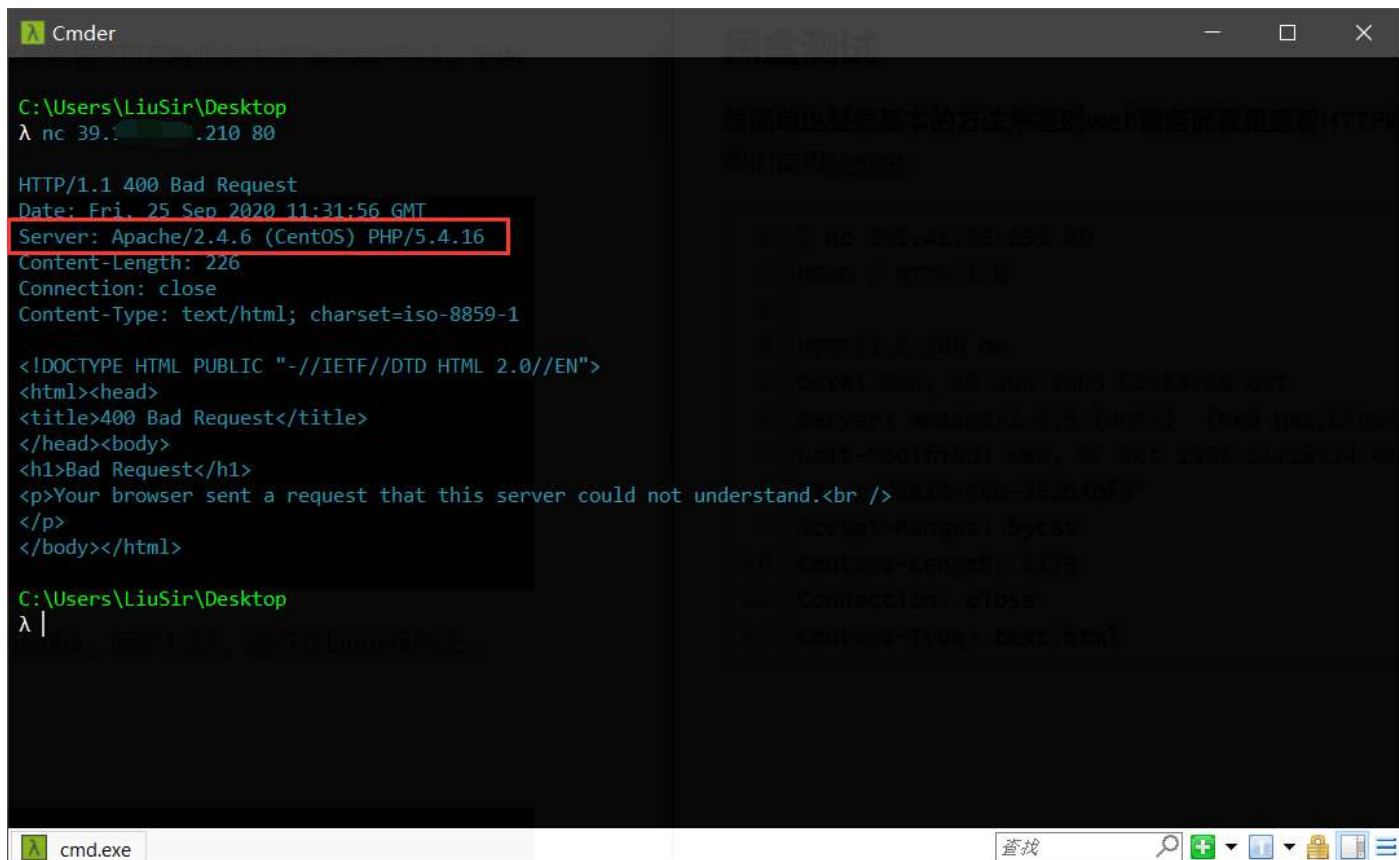
Web服务器指纹识别主要识别一下信息：

- 1、Web服务器名称，版本
- 2、Web服务器后端是否有应用服务器
- 3、数据库(DBMS)是否部署在同一主机(host)，数据库类型
- 4、Web应用使用的编程语言
- 5、Web应用框架
-

(1) 手工检测

- 1. HTTP头分析

即查看HTTP响应头中的Server、X-Powered-By、Cookie 等字段，这也是最基本的方法。



如上图，从Server字段，我们可以发现服务器可能是Apache，版本2.4.6，运行在CentOS Linux系统上。

根据X-Powered-By字段我们可以判断识别出web框架，并且不同的web框架有其特有的cookie，根据这个我们也能判断识别出web应用框架。

• 2. 协议行为

即从HTTP头字段顺序分析，观察HTTP响应头的组织顺序，因为每个服务器都有一个内部的HTTP头排序方法。

• 3. 浏览并观察网站

我们可以观察网站某些位置的HTML源码(特殊的class名称)及其注释(comment)部分，可能暴露有价值信息。观察网站页面后缀可以判断Web应用使用的编程语言和框架。

• 4. 刻意构造错误

错误页面可以给你提供关于服务器的大量信息。可以通过构造含有随机字符串的URL，并访问它来尝试得到404页面。

(2) 利用工具识别

whatweb是一款用于辅助的自动化Web应用指纹分析工具

常规扫描：

```
whatweb 域名/ip地址
```

```
root@kali:~# whatweb http://www.asfaa.org/  
/usr/lib/ruby/vendor_ruby/target.rb:188: warning: URI.escape is obsolete  
http://www.asfaa.org/ [200 OK] Apache[2.4.39], Country[UNITED STATES][US], Google-API[ajax/libs/jquery/1.4.2/jquery.min.js], Google-  
Analytics[UA-40438775-1], HTML5, HTTPServer[Apache/2.4.39], IP[70.32.68.120], JQuery[1.4.2], PHP[7.3.11], Script[text/javascript], T  
itle[ASFAA Home], X-Powered-By[PHP/7.3.11]  
root@kali:~#
```

批量扫描：

指定要扫描的文件

```
whatweb -i 含有需要扫描的域名的文件的路径
```

详细回显扫描：

```
whatweb -v 域名
```

Whatweb是一个基于Ruby语言的开源网站指纹识别软件，如上图，正如它的名字一样，whatweb能够识别各种关于网站的详细信息包括：CMS类型、博客平台、中间件、web框架模块、网站服务器、脚本类型、JavaScript库、IP、cookie等等。

另外，我们可以使用Nmap OS指纹初步判断操作系统。对于后端DBMS的识别，如果主机对外开放DBMS的话，可以通过端口特征判断，尤其是在开放默认端口比如3306、1443、27017等。

真实IP地址识别

在渗透测试中，一般只会给你一个域名，那么我们就需要根据这个域名来确定目标服务器的真实IP，我们可以通过像www.ip138.com这样的IP查询网直接获取目标的一些IP及域名信息，但这里的前提是目标服务器没有使用CDN。

什么是CDN？

CDN的全称是Content Delivery Network，即内容分发网络。CDN是构建在现有网络基础之上的智能虚拟网络，依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率。CDN的关键技术主要有内容存储和分发技术。

CDN将用户经常访问的静态数据资源直接缓存到节点服务器上，当用户再次请求时，会直接分发到在离用户近的节点服务器上响应给用户，当用户有实际数据交互时才会从远程Web服务器上响应，这样可以大大提高网站的响应速度及用户体验。CDN网络的诞生大大地改善了互联网的服务质量，因此传统的大型网络运营商纷纷开始建设自己的CDN网络。

因此，如果目标服务器使用了CDN服务，那么我们直接查询到的IP并不是真正的目标服务器的IP，而是一台离你最近的目标节点的CDN服务器，这就导致了我们没法直接得到目标服务器的真实IP。

如何判断目标服务器使用了CDN？

我们可以ping这个网站域名，比如我们ping百度：

```
C:\Users\LiuSir>ping www.baidu.com

正在 Ping www.a.shifen.com [61.135.185.32] 具有 32 字节的数据:
来自 61.135.185.32 的回复: 字节=32 时间=37ms TTL=53
来自 61.135.185.32 的回复: 字节=32 时间=40ms TTL=53
来自 61.135.185.32 的回复: 字节=32 时间=44ms TTL=53
来自 61.135.185.32 的回复: 字节=32 时间=46ms TTL=53

61.135.185.32 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 37ms, 最长 = 46ms, 平均 = 41ms

C:\Users\LiuSir>
```

如上图，我们可以看到百度使用了CDN。

我们也可以设置代理或者通过在线ping网站来在不同地区进行ping测试，然后对比每个地区ping出的IP结果，查看这些IP是否一致，一致，则极有可能不存在CDN。根据CDN的工作原理，如果网站使用了CDN，那么从全国各地访问网站的IP地址是各个CDN节点的IP地址，那么如果ping出来的IP大多不太一样或者规律性很强，可以尝试查询这些IP的归属地，判断是否存在CDN。有以下网站可以进行ping测试：

- <http://ping.chinaz.com/>
- <https://www.wepcc.com>
- <https://www.17ce.com>

以 <https://www.wepcc.com> 为例，如图所示，对 <https://www.baidu.com> 进行ping命令测试，根据IP地址和归属地不同，可以判断 <https://www.baidu.com> 使用了CDN。

网络工具Ping测试网站测试IP查询新版测试

登录

全球Ping测试

https://www.baidu.com

查询

全部电信联通移动多线港澳台海外

节点名称	解析IP	IP归属地	响应时间	TTL	赞助商
浙江-绍兴 (电信)	180.101.12	中国江苏南京 电信	15.8 ms	51	快快云
福建-泉州 (电信)	14.19.39	中国广东广州 电信	14.0 ms	53	快快网络
山东-济南 (联通)	61.13.32	中国北京 联通	13.4 ms	52	快快网络
北京 (移动)	39.18	中国北京 移动	3.53 ms	52	移动云
四川-成都 (多线)	14.7.38	中国广东广州 电信	33.1 ms	51	腾讯云
浙江-宁波 (多线)	18.12	中国江苏南京 电信	16.4 ms	49	快快网络
贵州-贵阳 (多线)	16.1.109	中国广东广州 联通	23.6 ms	46	快快云
香港 (港澳台)	10.3.39	中国香港 baidu.com	1.91 ms	56	阿里云国际版
日本 (海外)	11.151	日本 baidu.com	3.25 ms	54	阿里云
法国 (海外)	10.3.39	中国香港 baidu.com	240 ms	49	online
泰国-曼谷 (海外)	10.6.39	中国香港 baidu.com	66.1 ms	47	快快网络
美国-达拉斯 (海外)	10.9.123	美国加利福尼亚州圣克拉拉 baidu.com	18.0 ms	55	快快网络

Copyright © 2018 网络工具. All rights reserved.

联系我们

如何绕过CDN找到目标真实IP？

1. 利用子域名。一般来说很多站长可能只会对主站或者流量较大的分站使用CDN，但是一些流量比较小的分站可能没有挂CDN，这些分站和主站虽然不是同一个IP但是都在同一个C段下面的情况，所以我们可以通过ping二级域名获取分站IP，从而能判断出目标的真实IP段。
2. 查询主域。以前用CDN的时候有个习惯，只让WWW域名使用cdn，秃域名不使用，为的是在维护网站时更方便，不用等cdn缓存。所以试着把目标网站的www去掉，ping一下看ip是不是变了，如下图，这个方法还是挺有效的：

```
命令提示符
C:\Users\LiuSir>ping www.baidu.com

正在 Ping www.a.shifen.com [61.135.185.32] 具有 32 字节的数据:
来自 61.135.185.32 的回复: 字节=32 时间=37ms TTL=53
来自 61.135.185.32 的回复: 字节=32 时间=40ms TTL=53
来自 61.135.185.32 的回复: 字节=32 时间=44ms TTL=53
来自 61.135.185.32 的回复: 字节=32 时间=46ms TTL=53

61.135.185.32 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 37ms, 最长 = 46ms, 平均 = 41ms

C:\Users\LiuSir>ping baidu.com

正在 Ping baidu.com [220.181.38.148] 具有 32 字节的数据:
来自 220.181.38.148 的回复: 字节=32 时间=42ms TTL=48
来自 220.181.38.148 的回复: 字节=32 时间=53ms TTL=48
来自 220.181.38.148 的回复: 字节=32 时间=42ms TTL=48
来自 220.181.38.148 的回复: 字节=32 时间=40ms TTL=48

220.181.38.148 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 40ms, 最长 = 53ms, 平均 = 44ms

C:\Users\LiuSir>
```

3.扫描网站敏感文件，如phpinfo.php等，从而找到目标的真实IP。

4.从国外访问。国内很多CDN厂商因为各种原因只做了国内的线路，而针对国外的线路可能几乎没有，此时我们使用国外的主机直接访问可能就能获取到真实P。我们可以通过国外在线代理网站访问，可能会得到真实的IP地址，外国在线代理网站：

- <https://asm.ca.com/en/ping.php>

Ping a server or web site using our network of over 60 monitoring stations worldwide

(e.g. www.yahoo.com)

Start

Ping to: www.freebuf.com

Checkpoint	Result	min. rtt	avg. rtt	max. rtt	IP
India - Bangalore (inblr01)	Packets lost (100%)				123 129.169
Bulgaria - Sofia (bgsof03)	Packets lost (100%)				123 29.169
Australia - Brisbane (aubne03)	Packets lost (100%)				123 29.169
United States - Council Bluffs (uscb01)	Packets lost (100%)				123 29.169
India - Chennai (inche01)	Packets lost (100%)				123 29.169
United Kingdom - Cardiff (gbcar01)	Packets lost (100%)				123 29.169
United States - Cheyenne (usche01)	Packets lost (100%)				123 29.169
United States - Charleston (uschs02)	Packets lost (100%)				123 29.169
United States - Charleston (uschs01)	Packets lost (100%)				123 29.169
Canada - Toronto (cator03)	Packets lost (100%)				123 29.169
Czech Republic - Prague (czprg02)	Packets lost (100%)				123 29.169
Germany - Berlin (deber01)	Packets lost (100%)				123 29.169
Germany - Frankfurt (defra05)	Packets lost (100%)				123 29.169
Ireland - Dublin (iedub03)	Packets lost (100%)				123 29.169
Austria - Vienna (atvie02)	Packets lost (100%)				123 29.169
Netherlands - Eemshaven (nleem01)	Packets lost (100%)				123 29.169
France - Paris (frpar05)	Packets lost (100%)				123 29.169
United Kingdom - London (gblon03)	Packets lost (100%)				123 29.169
United Kingdom - Edinburgh (gbedi01)	Packets lost (100%)				123 29.169
Greece - Athens (grath02)	Packets lost (100%)				123 129.169
China - Hong Kong (nhkg03)	Packets lost (100%)				123 29.169
Finland - Hamina, Finland (fham01)	Packets lost (100%)				123 29.169
Hungary - Budapest (hubud02)	Packets lost (100%)				123 29.169
Italy - Milan (itmil01)	Packets lost (100%)				123 129.169
Indonesia - Jakarta (idjkt02)	Packets lost (100%)				123 29.169
India - Mumbai (inbom03)	Packets lost (100%)				123 29.169

?

Help

如上图，从国外代理访问目标网站的IP都是一样的。

5.通过邮件服务器。一般的邮件系统都在内部，没有经过CDN的解析，通过目标网站用户注册或者RSS订阅功能，查看邮件，寻找邮件头中的邮件服务器域名IP，ping这个邮件服务器的域名，由于这个邮件服务器的有可能跟目标Web在一个段上，我们直接一个一个扫，看返回的HTML源代码是否跟web的对的上，就可以获得目标的真实IP(必须是目标自己内部的邮件服务器，第三方或者公共邮件服务器是没有用的)。

6. 查看域名历史解析记录。也许目标很久之前没有使用CDN，所以可能会存在使用CDN前的记录。所以可以通过<https://www.netcraft.com>、<https://viewdns.info/>等网站来观察域名的IP历史记录。

7.Nslookup查询。查询域名的NS记录、MX记录、TXT记录等很有可能指向的是真实ip或同C段服务器。

8.利用网络空间搜索引擎。这里主要是利用网站返回的内容寻找真实原始IP，如果原始服务器IP也返回了网站的内容，那么可以在网上搜索大量的相关数据。最常见的网络空间搜索引擎有如下：

- Shodan: <https://www.shodan.io/>
- 钟馗之眼: <https://www.zoomeye.org/>
- FOFA: <https://fofa.so/>

9. 让目标主动连接我们。

1、发邮件给我们。比如订阅、注册的时候会有注册连接发送到我们的邮件，然后查看邮件全文源代码或邮件标头，寻找邮件头中的邮件服务器域名IP就可以了。

2、利用网站漏洞。比如有代码执行漏洞、SSRF、存储型的XSS都可以让服务器主动访问我们预设的web服务器，那么就能在日志里面看见目标网站服务器的真实IP。

.....

验证获得的真实IP地址

通过上面的方法获取了很多的IP地址（上面的方法4），此时我们需要确定哪一个才是真正的IP地址，如果是Web，最简单的验证方法是直接尝试用IP访问，看看响应的页面是不是和访问域名返回的一样即可。：

端口信息收集

在渗透测试的过程中，收集端口信息是一个十分重要的过程，通过扫描目标服务器开放的端口可以从该端口判断服务器上运行的服务。因为针对不同的端口具有不同的攻击方法，收集端口信息可以对症下药，便于我们渗透目标服务器。我们可以通过一下方法收集目标服务器的端口信息：

- 1. 使用nmap工具收集

```
nmap -A -v -T4 -O -sV 目标地址
```

• 2. 使用masscan探测端口开放信息

Masscan号称是最快的互联网端口扫描器，最快可以在六分钟内扫遍互联网。masscan的扫描结果类似于nmap(一个很著名的端口扫描器)，在内部，它更像scanrand, unicornscan, and ZMap，采用了异步传输的方式。它和这些扫描器最主要的区别是，它比这些扫描器更快。而且，masscan更加灵活，它允许自定义任意的地址范和端口范围。

```
root@kali:~# masscan 39.1.1.210 -p0-65535 --rate=10000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-09-28 01:06:56 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65536 ports/host]
Discovered open port 3790/tcp on 39.1.1.210
Discovered open port 22/tcp on 39.1.1.210
```

由于使用工具通常会在目标网站留下痕迹，接下来提供一种在线网站探测方法。

- 在线网站：<http://tool.chinaz.com/port/>
- ThreatScan在线网站（网站基础信息收集）：<https://scan.top15.cn/>
- Shodan：<https://www.shodan.io/>

常见的端口及其攻击方向可以参考：https://blog.csdn.net/weixin_42320142/article/details/102679143

社会工程学

相信看过《我是谁：没有绝对的安全系统》的朋友对社会工程学可能有深刻的印象了。全片的过程中，一直在处处渗透着社会工程学的原理，利用人们的胆小怕事，来获取自己利益。而后来，男主将社会工程学运用到了极致，成功为自己赢得了一个新的身份。

社会工程学（Social Engineering）是一种通过人际交流的方式获得信息的非技术渗透手段。其实，现在的黑客攻击也不止是仅仅通过网络来进行远程的渗透与入侵，还会通过社会工程学在线下场景中针对人性弱点进行相应的攻击。而且不幸的是，这种手段对于黑客来说非常有效，成功率也非常之高。事实上，社会工程学已是企业安全最大的威胁之一。狭义与广义社会工程学最明显的区别就是是否会与受害者产生交互行为。广义是有针对性的去对某一单一或多一目标进行攻击的行为。社会工程学在渗透测试中起着不小的作用，利用社会工程学，攻击者可以从一名员工的口中挖掘出本应该是秘密的信息。

凯文·米特尼克在《反欺骗的艺术》中曾提到，人为因素才是安全的软肋。很多企业、公司在信息安全上投入大量的资金，最终导致数据泄露的原因，往往却是发生在人本身。你们可能永远都想象不到，对于黑客们来说，通过一个用户名、一串数字、一串英文代码，社会工程师就可以通过这么几条的线索，通过社工攻击手段，加以筛选、整理，就能把你的所有个人情况信息、家庭状况、兴趣爱好、婚姻状况、你在网上留下的一切痕迹等个人信息全部掌握得一清二楚。虽然这个可能是最不起眼，而且还是最麻烦的方法。一种无需依托任何黑客软件，更注重研究人性弱点的黑客手法正在兴起，这就是社会工程学黑客技术。

社会工程学攻击包括四个阶段：

- 研究：信息收集（WEB、媒体、垃圾桶、物理），确定并研究目标
- 钩子：与目标建立第一次交谈（HOOK、下套）
- 下手：与目标建立信任并获取信息
- 退场：不引起目标怀疑的离开攻击现场

社会工程学收集的常见信息包括：姓名、性别、出生日期、身份证号、身份证家庭住址、身份证所在公安局、快递收货地址、大致活动范围、qq、手机号、邮箱、银行卡号（银行开户行）、支付宝、贴吧、百度、微博、猎聘、58、同城、网盘、微信、常用ID、学历（小/初/高/大学/履历）、目标性格详细分析、常用密码、照片EXIF信息。

常见可获取信息系统包括：中航信系统、春秋航空系统、12306系统、三大运营商网站、全国人口基本信息资源库、全国机动车/驾驶人信息资源库、各大快递系统（越权）、全国出入境人员资源库、全国在逃人员信息资源库、企业相关系统、全国安全重点单位信息资源库等。

举个例子：

假设我们要对一家公司进行渗透测试，正在收集目标的真实IP阶段，此时就可以利用收集到的这家公司的某位销售人员的电子邮箱。首先，给这位销售人员发送邮件，假装对某个产品很感兴趣，显然销售人员会回复邮件。这样攻击者就可以通过分析邮件头来收集这家公司的真实IP地址及内部电子邮件服务器的相关信息。通过进一步地应用社会工程学，假设现在已经收集了目标人物的邮箱、QQ、电话号码、姓名，以及域名服务商，也通过爆破或者撞库的方法获取邮箱的密码，这时就可以冒充目标人物要求客服人员协助重置域管理密码，甚至技术人员会帮着重置密码，从而使攻击者拿下域管理控制台，然后做域劫持。

详情请参考：

<https://www.zhihu.com/question/26113526>

<https://blog.csdn.net/Eastmount/article/details/100585715>

推荐书籍：

《黑客心理学》、《欺骗的艺术》

Ending.....

“知己知彼，百战不殆”，进行web渗透测试之前，最重要的一步那就是就是信息收集了，俗话说“渗透的本质也就是信息收集”，信息收集的深度，直接关系到渗透测试的成败。打好信息收集这一基础可以让测试者选择合适和准确的渗透测试攻击方式，缩短渗透测试的时间。

参考

https://blog.csdn.net/qq_41880069/article/details/83037081

https://blog.csdn.net/qq_32434307/article/details/107353811

<https://blog.csdn.net/Eastmount/article/details/102816621>

<https://www.freebuf.com/sectool/104256.html>

https://blog.csdn.net/weixin_41970600/article/details/104766547

https://blog.csdn.net/qq_36119192/article/details/84029809

<https://www.freebuf.com/news/137497.html>

<https://www.fujieace.com/penetration-test/cdn-find-ip.html>

https://blog.csdn.net/qq_36119192/article/details/89151336

<https://blog.csdn.net/zyhj2010/article/details/45064903>

<https://blog.csdn.net/Eastmount/article/details/100585715>

<https://blog.csdn.net/Eastmount/article/details/100585715>

<https://www.zhihu.com/question/26113526>