06-前期信息收集

#2课时

信息收集简介

蚁景网安 edu.hetianlab.com

1. 什么是信息收集

信息收集是指通过各种方式获取所需要的信息，以便我们在后续的渗透过程更好的进行。比如目标站点IP、中间件、脚本语言、端口、邮箱等等。信息收集包含资产收集但不限于资产收集。

2. 信息收集的意义

- 信息收集是渗透测试成功的保障
- 更多的暴露面
- 更大的可能性

3. 信息收集分类

- 主动信息收集

通过直接访问网站在网站上进行操作、对网站进行扫描等，这种是有网络流量经过目标服务器的信息收集方式。

- 被动信息收集

基于公开的渠道，比如搜索引擎等，在不与目标系统直接交互的情况下获取信息，并且尽量避免留下痕迹。

4. 收集哪些信息

- 域名信息（whois、备案信息、子域名）
- 服务器信息（端口、服务、真实IP）
- 网站信息（网站架构、操作系统、中间件、数据库、编程语言、指纹信息、WAF、敏感目录、敏感文件、源码泄露、旁站、C段）
- 管理员信息（姓名、职务、生日、联系电话、邮件地址）

域名信息收集

域名介绍

域名（Domain Name），简称域名、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，用于在数据传输时标识计算机的电子方位（有时也指地理位置）。

DNS（域名系统，Domain Name System）是互联网的一项服务。它作为将域名和IP地址相互映射的一个分布式数据库，能够使人更方便地访问互联网。

域名分类

顶级域名	二级域名	三级域名
.com	baidu.com	www.baidu.com

政府域名	商业域名	教育域名
.gov	.com	.edu

二级域名是指顶级域名之下的域名，在国际顶级域名下，它是指域名注册人的网上名称，例如 ibm，yahoo，microsoft等；在国家顶级域名下，它是表示注册企业类别的符号，例如com，top，edu，gov，net等

Whois

1. whois简介

Whois 是用来查询**域名的IP**以及**所有者**等信息的传输协议。就是一个用来查询域名是否被注册，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)

Whois 简单来说，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、域名注册日期和过期日期等）。通过域名Whois服务器查询，可以查询域名归属者联系方式，以及注册和到期时间

WHOIS协议是什么？ (biancheng.net)

whois查询的用处：通过whois查询可以获得域名注册者邮箱地址等信息。一般情况下对于中小型网站域名注册者就是网站管理员。利用搜索引擎对whois查询到的信息进行搜索，获取更多域名注册者的个人信息。

2. whois查询

- web接口查询

> https://whois.aliyun.com/
> https://www.whois365.com/cn/
> http://whois.chinaz.com/

- whois命令行查询

```
[root@kvm ~]# whois hetianlab.com
   Domain Name: HETIANLAB.COM
   Registry Domain ID: 1919907269_DOMAIN_COM-VRSN
   Registrar WHOIS Server: grs-whois.hichina.com
   Registrar URL: http://www.net.cn
```

```
   Updated Date: 2022-02-18T01:48:09Z
   Creation Date: 2015-04-15T03:30:41Z
   Registry Expiry Date: 2023-04-15T03:30:41Z
   Registrar: Alibaba Cloud Computing (Beijing) Co., Ltd.
   Registrar IANA ID: 420
   Registrar Abuse Contact Email: DomainAbuse@service.aliyun.com
   Registrar Abuse Contact Phone: +86.95187
   Domain Status: ok https://icann.org/epp#ok
   Name Server: DNS10.HICHINA.COM
   Name Server: DNS9.HICHINA.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2022-02-17T07:50:06Z <<<
```

- python脚本查询

```python
import whois

domain = input("输入查询 Whois 的域名: ")
data = whois.whois(domain)
print("域名: %s" % data['domain_name'])
print("邮箱: %s" % data['emails'])
print("注册人: %s" % data['org'])
print("注册时间: %s" % data['creation_date'])
print("更新时间: %s" % data['updated_date'])
```

备案信息

备案号是网站是否合法注册经营的标志，可以用网页的备案号反查出该公司旗下的资产。

web接口查询
https://beian.miit.gov.cn/
http://www.beian.gov.cn/portal/registerSystemInfo
http://icp.chinaz.com/
https://icplishi.com/

Whois反查

whois反查，可以通过注册人、注册人邮箱、注册人手机电话反查whois信息

先通过whois获取注册人和邮箱，再通过注册人和邮箱反查域名。

缺点是很多公司都是DNS解析的运营商注册的,查到的是运营商代替个人和公司注册的网站信息。

https://whois.chinaz.com/reverse?ddlSearchMode=1
http://whois.4.cn/reverse
https://whois.aizhan.com/

子域名

子域名简介

子域名指二级域名,二级域名是顶级域名(一级域名)的下一级。

比如`mail.heetian.com`和`bbs.heetian.com`是`heetian.com`的子域，而`heetian.com`则是顶级域名`.com`的子域。

google hacking

```
site:hetianlab.com
```

domain_search_conf.py

```
# -*- coding: utf-8 -*-
# @time      : 2022/2/23 0023 13:04
# @Author    : mingy
# @File      : domain_search_conf.py
# @Software : PyCharm

bing_cookie = ""

baidu_cookie = ""

User_Agent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 " \
             "Safari/537.36 Edg/98.0.1108.56"

Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8, " \
         "application/signed-exchange;v=b3;q=0.9"

proxy = {
    'http': '127.0.0.1:7891',
    'https': '127.0.0.1:7891'
```

攻防实验室 edu.hetianlab.com

```
}
```

domain_search.py

```
# -*- coding: utf-8 -*-
# @time      : 2022/2/18 0018 16:52
# @Author    : mingy
# @File      : domain_search.py
# @Software : PyCharm

# ! /usr/bin/env python
# _*_  coding:utf-8 _*_

import requests
from urllib.parse import urlparse
import sys
import re
from domain_search_conf import *


def bing_search(site, page):
 Subdomain = []
 headers = {
 'User-Agent': User_Agent,
 'Cookie': bing_cookie
    }
 for p in range(int(page)):
 try:
 url = "https://cn.bing.com/search?q=site%3A{0}&qs=n&form=QBRE&sp=-1&pq=site%3A{0}" \
                  "&sc=2-11&sk=&cvid=C1A7FC61462345B1A71F431E60467C43&toHttps=1" \

"&redig=3FEC4F2BE86247E8AE3BB965A62CD454&pn=2&first={1}1&FROM=PERE".format(site, p)

 # url = "https://www.bing.com/search?q=site:{0}&qs=n&sp=-1&pq=site:{0}" \
 #       "&sc=0-18&sk=&cvid=8D775C5402784E6096D8B5C3A3BA386A&first={1}&FORM=PERE".format(site, p) html = requests.get(url, headers=headers, timeout=3).content.decode()
 except:
 pass
```

```
 job_bt = re.findall('<a target="_blank" target="_blank" href="(.*?)"', html)
 for h in job_bt:
 domain = urlparse(h).netloc
            Subdomain.append(domain)
 Subdomain = list(set(Subdomain)) # 去重
 return Subdomain


def baidu_search(site, page):
 Subdomain = []
 headers = {
 'Accept': Accept,
 'User-Agent': User_Agent,
 'Accept - Encoding': "gzip, deflate, br",
 'Cookie': baidu_cookie
    }
 for p in range(int(page)):
 try:
 url = "https://www.baidu.com/s?wd=site%3A{0}&pn={1}0&oq=site%3A{0}" \
                  "&tn=baiduhome_pg&ie=utf-8&rsv_idx=2&rsv_pq=d59fc7380000344c" \

"&rsv_t=38efmxGEvInEMk2hU6IhokqHGzr3WTIIPSDy2Kx%2FsmGphjpX6JSRFpfdGfHMYJkw3le%2B".format(site, p)
 html = requests.get(url, headers=headers, timeout=3).content.decode()
 except:
 pass
 job_bt = re.findall('style="text-decoration:none;position:relative;">(.*?)/', html)
 Subdomain.extend(job_bt)
 Subdomain = list(set(Subdomain)) # 去重
 return Subdomain


def google_search(site, page):
 Subdomain = []
 headers = {'User-Agent': User_Agent}
 proxies = proxy
    for p in range(int(page)):
 try:
 url = "https://www.google.com/search?q=site:{0}" \
```

```
                    "&newwindow=1&ei=lC4TYuqRB4ed0wT0q6-
oBg&start={1}0&sa=N" \
                    "&ved=2ahUKEwjqq-
6Lk5D2AhWHzpQKHfTVC2U4ChDy0wN6BAgBEDs&biw=1872&bih=929&dpr=
1".format(site, p)

 html = requests.get(url, headers=headers, proxies=proxies,
timeout=3).content.decode()
 except:
 pass
 job_bt = re.findall('<cite class="iUh30 qLRx3b tjvcx"
role="text">(.*?)<', html)
 for h in job_bt:
 domain = urlparse(h).netloc
            Subdomain.append(domain)
 Subdomain = list(set(Subdomain)) # 去重
 return Subdomain


if __name__ == '__main__':
 if len(sys.argv) == 3:
 site = sys.argv[1]
 page = sys.argv[2]
 else:
 print("usage: %s baidu.com 10" % sys.argv[0])
 sys.exit(-1)

 bing_subdomain = bing_search(site, page)
 print("bing 搜索引擎获取子域名: {}
个".format(len(bing_subdomain)))
 # print("bing 搜索引擎获取子域名: ")
 # print(bing_subdomain)
 baidu_subdomain = baidu_search(site, page)
 print("baidu 搜索引擎获取子域名: {}
个".format(len(baidu_subdomain)))
 # print("baidu 搜索引擎获取子域名: ")
 # print(baidu_subdomain)
 google_subdomain = google_search(site, page)
 print("google 搜索引擎获取子域名: {}
个".format(len(google_subdomain)))
 # print("google 搜索引擎获取子域名: ")
 # print(google_subdomain)
 domain = baidu_subdomain + bing_subdomain +
google_subdomain

```

```
106      Subdomain = list(set(domain))
107   print("---去重后总共获取子域名： {} 个---
      ".format(len(Subdomain)))
108   print("===子域名===")
109   filename = site + ".txt"
110   for i in Subdomain:
111   print(i)
112   with open(filename, "a+") as f:
113   f.write(i + "\n")
114   f.close()
115
116   print("子域名搜索完毕...")
```

```
PS D:\Code\Python> python3 .\domain_search.py hetianlab.com 10
bing 搜索引擎获取子域名: 2 个
baidu 搜索引擎获取子域名: 18 个
google 搜索引擎获取子域名: 2 个
---去重后总共获取子域名： 18 个---
===子域名===
cqupt.hetianlab.com
hetianlab.com
sdust.hetianlab.com
www.hetianlab.org
edu.hetianlab.com
www.hetianlab.cn
szitu.hetianlab.com
hebiace.hetianlab.com
hit.hetianlab.com
cz.hetianlab.com
hebust.hetianlab.com
tjtc.hetianlab.com
jsvist.hetianlab.com
hcit.hetianlab.com
www.hetianlab.com
nuc.hetianlab.com
xjpcedu.hetianlab.com
jsjzi.hetianlab.com
子域名搜索完毕...
PS D:\Code\Python>
```

蚁景网安 edu.hetianlab.com

第三方Web接口

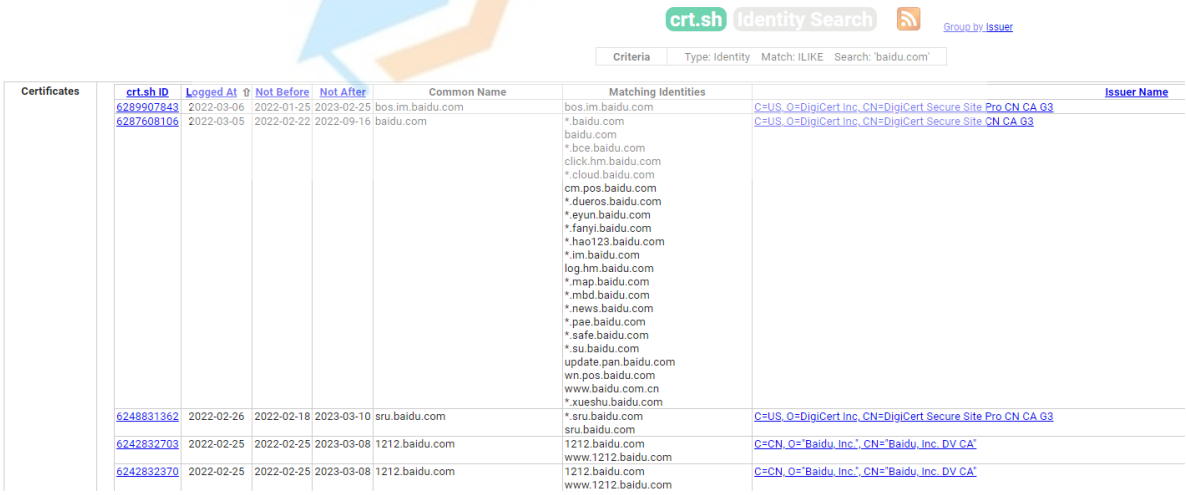
> https://dnsdumpster.com/
> https://www.dnsgrep.cn/
> https://developers.virustotal.com/reference/domains-relationships
> http://tool.chinaz.com/subdomain
> https://phpinfo.me/domain/
> https://www.nmmapper.com/sys/tools/subdomainfinder/

网络空间安全搜索引擎

```
https://fofa.info/
domain="hetianlab.com"

https://www.zoomeye.org/
site:"hetianlab.com"

https://hunter.qianxin.com/
domain="hetianlab.com"

https://www.shodan.io/
hostname:baidu.com
```

SSL证书查询

> https://crt.sh/



> https://developers.facebook.com/tools/ct/search/

蚁景网安 edu.hetianlab.com

hetianlab.com 搜索

域	主题	签发者	有效期	证书
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Jan 16, 2022 - Apr 17, 2022	显示详情
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Jan 16, 2022 - Apr 17, 2022	显示详情 (CT Precertificate)
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Nov 28, 2021 - Feb 27, 2022	显示详情 (CT Precertificate)
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Nov 28, 2021 - Feb 27, 2022	显示详情
hetianlab.com	CN=hetianlab.com	C=US, O=Let's Encrypt, CN=R3	Nov 17, 2021 - Feb 15, 2022	显示详情
hetianlab.com	CN=hetianlab.com	C=US, O=Let's Encrypt, CN=R3	Nov 17, 2021 - Feb 15, 2022	显示详情 (CT Precertificate)
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Sep 15, 2021 - Dec 15, 2021	显示详情
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Sep 15, 2021 - Dec 15, 2021	显示详情 (CT Precertificate)
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Aug 25, 2021 - Nov 24, 2021	显示详情 (CT Precertificate)
hetianlab.com	CN=hetianlab.com	C=AT, O=ZeroSSL, CN=ZeroSSL RSA Domain Secure Site CA	Aug 25, 2021 - Nov 24, 2021	显示详情
hetianlab.com	CN=hetianlab.com	C=US, O=Let's Encrypt, CN=R3	Jul 23, 2021 - Oct 21, 2021	显示详情

JS文件发现子域名

> https://github.com/Threezh1/JSFinder

```
root@kali:~/JSFinder# python3 JSFinder.py -u http://www.mi.com
url:http://www.mi.com
Find 64 URL:
http://s02.pre.mi.com/assets/
http://time.hd.mi.com/gettimestamp
http://hd.mi.com/x/03021d/img/loading.gif
http://order.mi.com
http://api.order.mi.com
http://cn.orderapi.mi.com
http://www.mi.com
http://cart.mi.com
http://item.mi.com
http://list.mi.com
http://search.mi.com
http://my.mi.com
http://tp.hd.mi.com/
```

子域名收集工具

1. 子域名挖掘机
2. OneForAll

> https://github.com/shmilylty/OneForAll

3. Subdomainsbrute

高并发的DNS暴力枚举工具

> https://github.com/lijiejie/subDomainsBrute

4. Sublist3r

https://github.com/aboul3la/Sublist3r

5. ESD

https://github.com/FeeiCN/ESD

6. dnsbrute

https://github.com/Q2h1Cg/dnsbrute

7. Anubis

https://github.com/jonluca/Anubis

8. subdomain3

https://github.com/yanxiu0614/subdomain3

9. teemo

https://github.com/bit4woo/teemo

10. Sudomy

https://github.com/screetsec/Sudomy

11. ARL

https://github.com/TophantTechnology/ARL

12. SubFinder + KSubdomain + HttpX

- **SubFinder**：用来查询域名的子域名信息的工具，可以使用很多国外安全网站的api接口进行自动化搜索子域名信息。
  - https://github.com/projectdiscovery/subfinder
- **HttpX**：一款运行速度极快的多功能HTTP安全工具，它可以使用retryablehttp库来运行多种网络探针，并使用了多线程机制来维持运行的稳定性和结果的准确性。
  - https://github.com/projectdiscovery/httpx
- **ksubdomain**是一款基于无状态子域名爆破工具，支持在Windows/Linux/Mac上使用，它会很快的进行DNS爆破，在Mac和Windows上理论最大发包速度在30w/s,linux上为160w/s的速度。

edu.hetianlab.com

- https://github.com/knownsec/ksubdomain

```
#subfinder基本使用
./subfinder -d baidu.com -o output.txt


#ksubdomain基本使用
./ksubdomain -d baidu.com

#管道操作
./subfinder -d baidu.com -silent|./ksubdomain -verify -silent|./httpx -title -content-length -status-code
#可以用管道结合在一起配合工作。达到收集域名，验证域名，http验证存活目的。
```

IP、端口信息收集

IP信息收集

- IP反查域名

http://stool.chinaz.com/same
https://tools.ipip.net/ipdomain.php
https://www.dnsgrep.cn/
https://site.ip138.com/

如果渗透目标为虚拟主机，那么通过IP反查到的域名信息很有价值，因为一台物理服务器上面可能运行多个虚拟主机。这些虚拟主机有不同的域名，但通常共用一个IP地址。如果你知道有哪些网站共用这台服务器，就有可能通过此台服务器上其他网站的漏洞获取服务器控制权，进而迂回获取渗透目标的权限，这种技术也称为“旁注”。

- 域名查询IP

http://ip.tool.chinaz.com/
https://ipchaxun.com/
https://site.ip138.com/

知道一个站点的域名需要得到它的IP以便之后获取端口信息或扫描等后续工作。

- C段存活主机探测

查找与目标服务器IP处于同一个C段的服务器IP

```
nmap -sP www.XXX.com/24
nmap -sP 192.168.1.*
```

https://github.com/se55i0n/Cwebscanner

蚁景网安 edu.hetianlab.com

```
python Cwebscan.py www.hetianlab.com
```

- CDN简介

CDN即内容分发网络。CDN是构建在网络之上的内容分发网络，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判断

1. 多地ping

http://ping.chinaz.com/
http://www.webkaka.com/Ping.aspx

用各种多地 ping 的服务，查看对应 IP 地址是否唯一

2. 国外访问

https://asm.ca.com/en/ping.php

因为有些网站设置CDN可能没有把国外的访问包含进去，所以可以这么绕过

- CDN绕过

1. 查询子域名的IP

https://ip.tool.chinaz.com/ipbatch

CDN 流量收费高，所以很多站长可能只会对主站或者流量大的子站点做了 CDN，而很多小站子站点又跟主站在同一台服务器或者同一个C段内，此时就可以通过查询子域名对应的 IP 来辅助查找网站的真实IP

2. MX记录邮件服务

MX记录是一种常见的查找IP的方式。如果网站在与web相同的服务器和IP上托管自己的邮件服务器，那么原始服务器IP将在MX记录中。

3. 查询历史DNS记录

https://dnsdb.io/zh-cn/
https://securitytrails.com/
https://viewdns.info/iphistory/
https://www.ip138.com/

查看 IP 与 域名绑定的历史记录，可能会存在使用 CDN 前的记录;

域名注册完成后首先需要做域名解析，域名解析就是把域名指向网站所在服务器的IP，让人们通过注册的域名可以访问到网站。

蚁景网安 edu.hetianlab.com

IP地址是网络上标识服务器的数字地址，为了方便记忆，使用域名来代替IP地址。

域名解析就是域名到IP地址的转换过程，域名的解析工作由DNS服务器完成。

DNS服务器会把域名解析到一个IP地址，然后在此IP地址的主机上将一个子目录与域名绑定。

域名解析时会添加解析记录，这些记录有：A记录、AAAA记录、CNAME记录、MX记录、NS记录、TXT记录。

DNS记录类型

> https://developer.aliyun.com/article/331012

- A记录

用来指定主机名（或域名）对应的IP地址记录

通俗来说A记录就是服务器的IP，域名绑定A记录就是告诉DNS，当你输入域名的时候给你引导向设置在DNS的A记录所对应的服务器。

- NS记录

域名服务器记录，用来指定该域名由哪个DNS服务器来进行解析。

- MX记录

邮件交换记录，它指向一个邮件服务器，用于电子邮件系统发邮件时根据收信人的地址后缀来定位邮件服务器。

- CNAME记录

别名记录，允许您将多个名字映射到同一台计算机

- TXT记录

一般指某个主机名或域名的说明

- 泛域名与泛解析

泛域名是指在一个域名根下，以 `*.Domain.com` 的形式表示这个域名根所有未建立的子域名。 泛解析是把 `*.Domain.com` 的A记录解析到某个IP 地址上，通过访问任意的前缀 `.domain.com` 都能访问到你解析的站点上。

- 域名绑定

域名绑定是指将域名指向服务器IP的操作

端口信息收集

- 端口简介

在Internet上，各主机间通过TCP/IP协议发送和接受数据包，各个数据包根据其目的主机的IP地址来进行互联网络中的路由选择，从而顺利的将数据包顺利的传送给目标主机

但当目的主机运行多个程序时，目的主机该把接受到的数据传给多个程序进程中的哪一个呢？端口机制的引入就是为了解决这个问题。端口在网络技术中，端口有两层意思：一个是物理端口，即物理存在的端口，如：集线器、路由器、交换机、ADSL Modem等用于连接其他设备的端口；另一个就是逻辑端口，用于区分服务的端口，一般用于TCP/IP中的端口，其范围是0~65535,，0为保留端口，一共允许有65535个端口比如用于网页浏览服务的端口是80端口，用于FTP服务的是21端口。 这里我们所指的不是物理意义上的端口，而是特指TCP/IP协议中的端口，是逻辑意义上的端口

- 协议端口

根据提供服务类型的不同，端口可分为以下两种：

TCP端口：TCP是一种面向连接的可靠的传输层通信协议
UDP端口：UDP是一种无连接的不可靠的传输层协议

TCP协议和UDP协议是独立的，因此各自的端口号也互相独立。

TCP：给目标主机发送信息之后，通过返回的应答确认信息是否到达
UDP：给目标主机放信息之后，不会去确认信息是否到达

而由于物理端口和逻辑端口数量较多，为了对端口进行区分，将每个端口进行了编号，即就是端口号。那么看到这里我们会好奇，有那么多的端口，他们到底是怎么分类的?

- 端口类型

周知端口：众所周知的端口号，范围：`0-1023`，如`80`端口是`www`服务
动态端口：一般不固定分配某种服务，范围：`49152-65535`
注册端口：范围：`1024-49151`，用于分配给用户进程或程序

- 渗透端口

https://www.cnblogs.com/bmjoker/p/8833316.html

常见端口介绍

- FTP-21

FTP：文件传输协议，使用TCP端口20、21，20用于传输数据，21用于传输控制信息
（1）ftp基础爆破：owasp的Bruter,hydra以及msf中的ftp爆破模块。
（2) ftp匿名访问：用户名：anonymous 密码：为空或者任意邮箱
（3）vsftpd后门 ： vsftpd 2到2.3.4版本存在后门漏洞，通过该漏洞获取root权限。
（4）嗅探：ftp使用明文传输，使用Cain进行渗透。（但是嗅探需要在局域网并需要欺骗或监听网关）
（5）ftp远程代码溢出。
（6）ftp跳转攻击。

漏洞复现-vsftpd-v2.3.4：
https://www.freebuf.com/column/143480.html

ProFTPD 1.3.3c远程命令执行：
https://blog.csdn.net/weixin_42214273/article/details/82892282

FTP跳转攻击：
https://blog.csdn.net/mgxcool/article/details/48249473

- SSH-22

SSH：(secure shell)是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。
（1）弱口令，可使用工具hydra，msf中的ssh爆破模块。
（2）SSH后门 （https://www.secpulse.com/archives/69093.html）
（3）openssh 用户枚举 CVE-2018-15473。（https://www.anquanke.com/post/id/157607）

- WWW-80

为超文本传输协议（HTTP）开放的端口，主要用于万维网传输信息的协议
（1）中间件漏洞，如IIS、apache、nginx等
（2）80端口一般通过web应用程序的常见漏洞进行攻击

- NetBIOS SessionService–139/445

139用于提供windows文件和打印机共享及UNIX中的Samba服务。
445用于提供windows文件和打印机共享。

（1）对于开放139/445端口，尝试利用MS17010溢出漏洞进行攻击；
（2）对于只开放445端口，尝试利用MS06040、MS08067溢出漏洞攻击；
（3）利用IPC$连接进行渗透

- MySQL-3306

3306是MYSQL数据库默认的监听端口
（1）mysql弱口令破解
（2）弱口令登录mysql，上传构造的恶意UDF自定义函数代码，通过调用注册的恶意函数执行系统命令
（3）SQL注入获取数据库敏感信息，load_file()函数读取系统文件，导出恶意代码到指定路径

- RDP-3389

3389是windows远程桌面服务默认监听的端口
（1）RDP暴力破解攻击
（2）MS12_020死亡蓝屏攻击
（3）RDP远程桌面漏洞（CVE-2019-0708）
（4）MSF开启RDP、注册表开启RDP

- Redis-6379

开源的可基于内存的可持久化的日志型数据库。
（1）爆破弱口令
（2）redis未授权访问结合ssh key提权
（3）主从复制rce

http://hetianlab.com/expc.do?ec=ECID9f92-ff93-4a94-a821-f0b968ef4985

- Weblogic-7001

WebLogic是美国Oracle公司出品的一个application server，确切的说是一个基于JAVAEE架构的中间件，WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器

（1）弱口令、爆破，弱密码一般为weblogic/Oracle@123 or weblogic
（2）管理后台部署 war包后门
（3）weblogic SSRF
（4）反序列化漏洞

https://fuping.site/2017/06/05/Weblogic-Vulnerability-Verification/

Weblogic_ssrf实例：
http://hetianlab.com/expc.do?ec=ECID9d6c0ca797abec20170210143122000001

CNVD-C-2019-48814 WebLogic反序列化远程命令执行：
http://hetianlab.com/expc.do?ec=ECID3f28-5c9a-4f95-999d-68fa2fa7b7aa

- Tomcat-8080

Tomcat 服务器是一个开源的轻量级Web应用服务器,在中小型系统和并发量小的场合下被普遍使用,是开发和调试Servlet、JSP 程序的首选

（1）Tomcat远程代码执行漏洞（CVE-2019-0232）
（2）Tomcat任意文件上传（CVE-2017-12615）
（3）tomcat 管理页面弱口令getshell

CVE-2019-0232 Tomcat远程代码执行漏洞：
http://hetianlab.com/expc.do?ec=ECIDefcf-3af2-438f-848f-8dc0f9e6b821

端口扫描

NMAP

- NMAP简介

Network Mapper，是一款开放源代码的网络探测和安全审核的工具

nmap参考指南（中文版）
https://nmap.org/man/zh/

- 功能介绍

1. 检测网络存活主机（主机发现）
2. 检测主机开放端口（端口发现或枚举）
3. 检测相应端口软件（服务发现）版本
4. 检测操作系统，硬件地址，以及软件版本
5. 检测脆弱性的漏洞（nmap的脚本）

- 端口状态

```
Open                端口开启，数据有到达主机，有程序在端口上监控
Closed              端口关闭，数据有到达主机，没有程序在端口上监控
Filtered            数据没有到达主机，返回的结果为空，数据被防火墙或IDS过滤
UnFiltered          数据有到达主机，但是不能识别端口的当前状态
Open|Filtered       端口没有返回值，主要发生在UDP、IP、FIN、NULL和Xmas扫描中
Closed|Filtered     只发生在IP ID idle扫描
```

- 基础用法

```
nmap -A -T4 192.168.1.1

A：全面扫描\综合扫描
T4：扫描速度，共有6级，T0-T5

不加端口说明扫描默认端口，1-1024 + nmap-service

```

```
单一主机扫描：namp 192.168.1.2
子网扫描：namp 192.168.1.1/24
多主机扫描：nmap 192.168.1.1 192.168.1.10
主机范围扫描：namp 192.168.1.1-100
IP地址列表扫描：nmap -iL target.txt

扫描除指定IP外的所有子网主机：
nmap 192.168.1.1/24 --exclude 192.168.1.1

扫描除文件中IP外的子网主机：
nmap 192.168.1.1/24 --excludefile xxx.txt

扫描特定主机上的80,21,23端口：
nmap -p 80,21,23 192.168.1.1

```

- 扫描全部端口

```
nmap -sS -v -T4 -Pn -p 0-65535 -oN FullTCP -iL liveHosts.txt

• -sS：SYN扫描,又称为半开放扫描，它不打开一个完全的TCP连接，执行得很快，效率高（一个完整的tcp连接需要3次握手，而-sS选项不需要3次握手）

优点：Nmap发送SYN包到远程主机，但是它不会产生任何会话，目标主机几乎不会把连接记入系统日志。（防止对方判断为扫描攻击），扫描速度快，效率高，在工作中使用频率最高
缺点：它需要root/administrator权限执行

• -Pn：扫描之前不需要用ping命令，有些防火墙禁止ping命令。可以使用此选项进行扫描
• -iL：导入需要扫描的列表

```

- 扫描常用端口及服务信息

```
nmap -sS -T4 -Pn -oG TopTCP -iL LiveHosts.txt

系统扫描
nmap -O -T4 -Pn -oG OSDetect -iL LiveHosts.txt

版本检测
nmap -sV -T4 -Pn -oG ServiceDetect -iL LiveHosts.txt

```

- NMAP漏洞扫描

```
nmap.exe -p445 -v --script smb-ghost 192.168.1.0/24
```

网站信息收集

操作系统

1. `ping`判断：windows的TTL值一般为128，Linux则为64。
   TTL大于100的一般为windows，几十的一般为linux。
2. `nmap -O`参数
3. windows大小写不敏感，linux则区分大小写

- 网站服务、容器类型

1. F12查看响应头Server字段
2. whatweb
   https://www.whatweb.net/
3. wappalyzer插件

apache，nginx，tomcat，IIS
通过容器类型、版本可考虑对应容器存在的漏洞（解析漏洞）

脚本类型

1. php
2. jsp
3. asp/aspx
4. python

数据库类型

1. mysql
2. sqlserver
3. access
4. oracle

知道是什么语言才可以针对性的进行文件扫描、文件上传

CMS识别

CMS：内容管理系统，用于网站内容文章管理
https://github.com/lengjibo/dedecmscan

常见CMS：dedecms(织梦)、Discuz、phpcms等。

蚁景网安 edu.hetianlab.com

在线识别工具
http://whatweb.bugscaner.com/look/

Onlinetools
https://github.com/iceyhexman/onlinetools
https://pentest.gdpcisa.org/

敏感文件、目录

敏感文件、敏感目录挖掘一般都是靠工具、脚本来找，比如御剑、BBscan，当然大佬手工也能找得到。

```
github
git
svn
.DS_Store
.hg
.bzr
cvs
WEB-INF
备份文件
```

前面七种为版本管理工具所泄露的常规方式，上面的两种方式为操作不当，安全意识薄弱所造成的泄露。

Github泄露

开发人员将代码上传到网站，在上传的时候，没有删除重要的一些信息。如邮箱信息，SVN信息，内部账号和密码，数据库连接信息，服务器配置信息等。尤其是邮箱信息和内部账号和密码。这类信息可以通过在github上搜索公司的一些特定信息，查看是否有程序员将这些信息上传到了github上。

如公司的域名如下：niniub.com；则可以在github上用这个信息去进行搜索，看看是否有包含该类关键字的文件。这类安全漏洞只能靠人员的安全意识进行防护，没有其它方法进行。

.git泄露

成因及危害：当前大量开发人员使用git进行版本控制，对网站进行自动部署。如果配置不当，可能会将.git文件部署到线上环境，这就引起了git泄露漏洞。在网站安全维护方面，git和svn信息泄露，是非常常见也是非常致命的漏洞。会导致整个网站的源码泄露。

渗透测试人员、攻击者，可以进一步审计代码，挖掘：文件上传，SQL注入等web安全漏洞。

防护方法：在部署的时候，对.git文件夹进行删除；也可以在nginx配置中，对.git目录的访问进行屏蔽。

```
".git" intitle:"index of"

https://github.com/lijiejie/GitHack
GitHack是一个.git泄露利用脚本，通过泄露的.git文件夹下的文件，重建还原工程源代码。
```

.svn泄露

跟git一样，都是用来版本迭代的一个功能。具体一点就是使用svn checkout功能来更新代码。

如果没有将.svn版本控制的目录进行删除，恶意用户就可以使用这个目录下的文件，来恢复源码。从而可以获取如数据库密码，源码漏洞等信息。

防护：在部署的时候，将该文件进行删除

```
".svn" intitle:"index of“

https://github.com/admintony/svnExploit
```

网站备份文件

网站备份文件泄露指管理员误将网站备份文件或是敏感信息文件存放在某个网站目录下。
https://github.com/7kbstorm/7kbscan-WebPathBrute

目录探测

外部黑客可通过暴力破解文件名等方法下载该备份文件，导致网站敏感信息泄露。

dirsearch：
https://github.com/maurosoria/dirsearch

dirmap：
https://github.com/H4ckForJob/dirmap

御剑后台扫描工具

蚁景网安 edu.hetianlab.com

网站WAF识别

- WAF定义

`WAF`，即：`Web Application FireWall`（Web应用防火墙）。可以通俗的理解为：用于保护网站，防黑客、防网络攻击的安全防护系统；是最有效、最直接的Web安全防护产品。

- WAF功能

1. 防止常见的各类网络攻击，如：SQL注入、XSS跨站、CSRF、网页后门等；
2. 防止各类自动化攻击，如：暴力破解、撞库、批量注册、自动发贴等；
3. 阻止其它常见威胁，如：爬虫、0 DAY攻击、代码分析、嗅探、数据篡改、越权访问、敏感信息泄漏、应用层DDOS、远程恶意包含、盗链、越权、扫描等。

- WAF识别

wafw00f

https://github.com/EnableSecurity/wafw00f

```
nmap -p80,443 --script http-waf-detect ip
nmap -p80,443 --script http-waf-fingerprint ip
```

看图识waf,常见WAF拦截页面总结：
https://mp.weixin.qq.com/s/PWkqNsygi-c_S7tW1y_Hxw

教育网安 edu.hetianlab.com