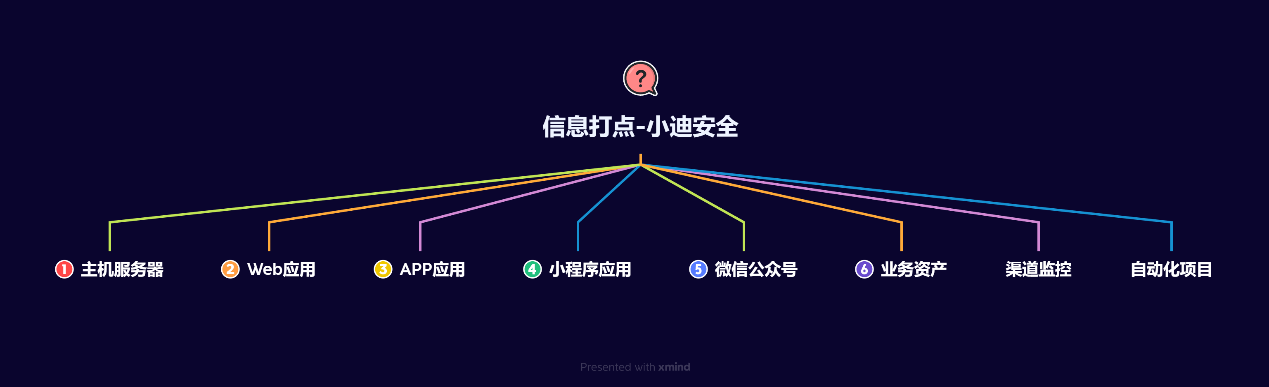
信息收集-Web应用&各语言框架&安全组件&联动系统&数据特征&人工分析&识别项目



#知识点：  
1、信息收集-服务器系统-操作系统&IP资产  
2、信息收集-服务器系统-端口扫描&服务定性  
---------------------------------------------------  
1、信息收集-Web应用-机构产权&域名相关性  
2、信息收集-Web应用-DNS&证书&枚举子域名  
---------------------------------------------------  
1、信息收集-Web应用-架构分析&指纹识别  
2、信息收集-Web应用-架构分析&WAF&蜜罐  
3、信息收集-Web应用-架构分析&框架组件识别  
---------------------------------------------------  
1、信息收集-Web应用-源码获取-已知指纹&未知指纹  
2、信息收集-Web应用-源码获取-泄漏问题&发现指纹  
---------------------------------------------------  
1、信息收集-Web应用-JS提取分析-人工&插件\*项目  
2、信息收集-Web应用-JS提取分析-URL&配置&逻辑  
---------------------------------------------------  
1、信息收集-Web应用-CDN加速-如何识别  
2、信息收集-Web应用-CDN加速-绕过方法  
---------------------------------------------------  
1、信息收集-Web应用-开发框架-识别安全  
2、信息收集-Web应用-安全组件-特征分析  
  
#章节点：(待补充)  
#具体点：(待补充)

演示案例：

* Web应用-架构分析-WAF&蜜罐识别
* Web应用-架构分析-框架组件指纹识别
* Web应用-源码获取-已知指纹&未知指纹
* Web应用-源码获取-泄漏问题&发现指纹
* Web应用-JS前端分析-人工&提取项目
* Web应用-JS前端分析-插件&流量匹配
* Web应用-框架组件-开发语言&识别&安全
* Web应用-CDN加速-识别&绕过&安全影响

#操作系统  
1、Web大小写  
2、端口服务特征  
3、TTL值判断返回  
  
#IP资产  
归属地查询  
归属云厂商  
IP反查机构  
IP反查域名  
IP-C段查询  
这里可以转换成Web上，机构上，从而获取更多信息  
  
#端口资产  
1、网络资产引擎：直接使用网络测绘引擎(Fofa、Hunter、Quake、00信安等搜索IP；此类网络资产测绘引擎都是每隔一段时间会对全网的网络资产去做一个轮询，那可能每个引擎的轮询周期、扫描精准度都不同，故建议可使用不同的搜索引擎以获取更多资产。  
地址导航：https://dh.aabyss.cn/  
参考：https://mp.weixin.qq.com/s/FRgPQKJDj2xRCduwPfZrTw  
  
2、在线端口扫描：百度或google直接搜索在线端口扫描就会有一些网站，同理很多功能都可以直接搜索在线xxx；例如在线正则提取解析、在线编码转换等。  
  
3、本地离线工具：推荐使用Nmap、Masscan、Fscan、KScan，其中Nmap最为准确，但最慢；Masscan最快，误报相对更高。  
端口扫描：https://xz.aliyun.com/t/15753  
演示：Yakit Nmap TscanPlus FScan Tanggo等  
考虑：1、防火墙 2、内网环境  
可能出现案例：数据库端口开放，但进行端口扫描，发现数据库端口没有开放（排除防火墙）\*注意：扫描中选择扫描协议是绕过安全组防火墙设置的一种手法，具体成功需看出网入网配置  
  
#应用服务  
见上图端口协议对应服务应用  
  
#角色定性判定  
1、网站服务器  
2、数据库服务器  
3、邮件系统服务器  
4、文件存储服务器  
5、网络通信服务器  
6、安全系统服务器

主动信息收集：  
通过直接经过目标服务器网络流量的信息收集方式。  
  
被动信息收集：  
不与目标系统直接交互的情况下获取信息收集方式。  
  
域名：  
1、备案信息  
通过域名查备案信息，备案信息获取更多域名  
2、企业产权  
通过企业产权查询Web,APP,小程序等版权资产  
3、域名相关性  
Whois信息：例如域名所有人、域名注册商、邮箱等。  
通过域名注册接口获取后缀  
查询域名注册邮箱  
通过域名查询备案号  
通过备案号查询域名  
反查注册邮箱  
反查注册人  
通过注册人查询到的域名在查询邮箱  
通过上一步邮箱去查询域名  
查询以上获取出的域名的子域名  
  
子域名：  
在后续测试中，还要注意对子域名进行筛选整理，太多的垃圾子域名和没用的子域名，主要看你的收集的子域名方法决定。  
1、DNS数据  
以DNS解析历史记录查询域名资产  
2、证书查询  
以SSL证书解析查询域名资产  
3、网络空间  
多网络空间综合型获取的记录  
4、威胁情报  
各类接口的集成的记录  
5、枚举解析  
结果主要以字典决定  
https://github.com/knownsec/ksubdomain  
https://github.com/shmilylty/OneForAll  
6、JS提取子域名  
后续会讲到

#Web架构  
开源CMS：  
Discuz、WordPress、PageAdmin、蝉知等  
  
前端技术：  
HTML5、Jquery、Bootstrap、Vue、NodeJS等  
  
开发语言：  
PHP、JAVA、Ruby、Python、C#、JS、Go等  
  
框架组件：  
SpringMVC、Thinkphp、Yii、Tornado、Vue等  
  
Web服务器：  
Apache、Nginx、IIS、Lighttpd等  
  
应用服务器：  
Tomcat、Jboss、Weblogic、Websphere等  
  
数据库类型：  
端口扫描，组合判断，应用功能  
Mysql、SqlServer、Oracle、Redis、MongoDB等  
  
操作系统信息：  
Linux、Windows、Mac等  
  
应用服务信息：  
FTP、SSH、RDP、SMB、SMTP、LDAP、Rsync等  
  
CDN信息：  
帝联、Cloudflare、网宿、七牛云、阿里云等  
  
WAF信息：  
创宇盾、宝塔、ModSecurity、玄武盾、OpenRASP等。  
  
蜜罐信息：  
HFish、TeaPot、T-Pot、Glastopf等  
  
其他组件信息：  
FastJson、Shiro、Log4j、Solr、XStream等  
  
#指纹识别：  
1、平台：  
https://www.yunsee.cn/ http://finger.tidesec.net/  
https://fp.shuziguanxing.com/  
  
2、项目：  
https://github.com/AliasIO/wappalyzer  
https://github.com/HackAllSec/hfinger  
https://github.com/Tuhinshubhra/CMSeeK  
https://github.com/lemonlove7/EHole\_magic  
https://github.com/emo-crab/observer\_ward   
3、使用网络空间的ico识别  
借助目标的ic图标进行识别  
  
#WAF识别：  
拦截页面，identywaf项目内置  
https://github.com/stamparm/identYwaf  
https://github.com/EnableSecurity/wafw00f  
  
#蜜罐识别：  
1、项目识别：  
https://github.com/graynjo/Heimdallr  
https://github.com/360quake/quake\_rs  
quake.exe init apikey值  
quake.exe honeypot 目标  
2、人工识别：  
\*端口多而有规律性（针对多服务蜜罐）  
\*Web访问协议就下载（针对多服务蜜罐）  
\*设备指纹分析（见上图，针对蜜罐特征）  
  
#框架组件识别：  
后续会讲到更多识别方法及项目插件  
例子：  
FastJson：https://forum.butian.net/share/1679  
Shiro：https://mp.weixin.qq.com/s/j1nDnb0Ub5bk2-Tq5tt2pg

参考：  
https://www.secpulse.com/archives/124398.html  
https://mp.weixin.qq.com/s/QgLDdaefXlZtvlSiFQShZw  
https://mp.weixin.qq.com/s/zhjxW2mUEgq6dFAp3kBcgQ  
  
#源码泄漏原因：  
1、从源码本身的特性入口  
2、从管理员不好的习惯入口  
3、从管理员习惯的开发入口  
4、从管理员不好的配置入口  
5、从管理员不好的意识入口  
6、从管理员资源信息搜集入口  
源码泄漏大概集合：  
Webpack打包泄漏  
composer.json  
GitHub源码泄漏  
git源码泄露  
svn源码泄露  
网站备份压缩文件  
DS\_Store文件泄露  
hg源码泄漏  
SWP 文件泄露  
CVS泄露  
Bzr泄露  
WEB-INF/web.xml 泄露  
  
思路点：  
1、使用指纹识别平台获取到目标是什么名字的源码程序  
通过搜索引擎获取到源码  
  
2、使用指纹识别平台不能获取目标信息  
后端：借助svn git composer.json ds\_store 备份扫描等配置泄漏安全  
前端：webpack打包 谷歌插件获取源码  
如果没有这些问题 还想获取源码怎么办 见下面方法  
  
3、借助资源平台去搜索（github gitee oschina等源码）  
-特定文件  
-特征联系方式  
-github监控（ 除了源码，漏洞更新，工具更新等都有用处）  
目标使用的源码是开源还是闭源，是否公开  
  
#演示案例：  
1、已知指纹识别获取源码途径  
  
2、GIT泄漏-未知指纹识别获取源码途径  
3、SVN泄漏-未知指纹识别获取源码途径  
4、DS\_Store泄漏-未知指纹识别获取源码途径  
GIT：https://github.com/lijiejie/GitHack  
SVN：https://github.com/callmefeifei/SvnHack  
DS\_Store：https://github.com/lijiejie/ds\_store\_exp  
通用：https://github.com/0xHJK/dumpall  
  
5、WebPack打包-未知指纹识别获取源码途径  
https://github.com/NothingCw/SourceDetector-dist  
  
6、composer.json-未知指纹识别获取源码途径  
7、源码文件备份泄漏-未知指纹识别获取源码途径  
目录文件扫描工具获取存在，直接访问  
 8、Github资源搜索-未知指纹识别获取源码途径  
\*拓展：配合后期监控保证第一时间通知  
资源搜索语法：  
in:name test #仓库标题搜索含有关键字   
in:descripton test #仓库描述搜索含有关键字   
in:readme test #Readme文件搜素含有关键字   
stars:>3000 test #stars数量大于3000的搜索关键字   
stars:1000..3000 test #stars数量大于1000小于3000的搜索关键字 forks:>1000 test #forks数量大于1000的搜索关键字   
forks:1000..3000 test #forks数量大于1000小于3000的搜索关键字 size:>=5000 test #指定仓库大于5000k(5M)的搜索关键字 pushed:>2019-02-12 test #发布时间大于2019-02-12的搜索关键字 created:>2019-02-12 test #创建时间大于2019-02-12的搜索关键字 user:test #用户名搜素   
license:apache-2.0 test #明确仓库的 LICENSE 搜索关键字 language:java test #在java语言的代码中搜索关键字   
user:test in:name test #组合搜索,用户名test的标题含有test的  
关键字配合谷歌搜索：  
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   
site:Github.com smtp password   
site:Github.com String password smtp  
  
9、另类目标源码获取  
https://www.huzhan.com/  
https://www.goofish.com/  
从目标转到寻找源码系统上，从源码系统上找应用目标  
  
#源码泄漏实战文章  
渗透测试实战---某资金盘信息收集(.idea目录泄露)  
https://mp.weixin.qq.com/s/7cSrDZci\_drE6wTGvQOOHw  
SRC挖掘-js.map泄露到接管云上域控  
https://mp.weixin.qq.com/s/QZsNjbTiaqC1qwzARVAiGQ

在日常渗透测试中，从JavaScript(JS)文件中提取信息是一项关键的步骤，以往有些案例就是通过JS文件中发现的敏感信息从而拿下重要的系统。在JS开发的WEB应用和PHP，JAVA,NET等区别在于即没有源代码，也可以通过浏览器的查看源代码获取部分源代码逻辑。从而获取URL，获取JS敏感信息，获取代码传参等，所以相当于JS开发的WEB应用大部分属于白盒测试（默认有大量源码参考），一般会在JS中寻找更多的URL地址，在JS代码逻辑（加密算法，APIkey配置，验证逻辑等）进行后期安全测试。  
  
0、JS信息收集  
 接口URL提取  
 后端地址提取  
 敏感信息提取  
 配置信息提取  
 其他信息提取（环境，注释，用途等）  
  
1、JS安全问题  
 源码泄漏，代码审计，JS逆向分析  
 未授权访问=JS里面分析更多的URL访问确定接口路径  
 敏感Key泄漏=JS文件中配置接口信息（云应用，短信，邮件，数据库等）  
 API接口安全=（代码中加密提交参数传递，更多的URL路径）  
  
2、流行的Js框架有那些？  
 Vue NodeJS jQuery Angular等  
  
3、如何判定JS开发应用？  
 插件wappalyzer  
 源程序代码简短  
 引入多个js文件  
 一般有/static/js/app.js等顺序的js文件  
 一般cookie中有connect.sid  
   
4、如何获取更多的JS文件？  
 手工-浏览器搜索  
 半自动-Burpsuite插件  
 工具化-各类提取&FUZZ项目  
   
5、如何快速获取价值信息？  
 src=  
 path=  
 method:"get"  
 http.get("  
 method:"post"  
 http.post("  
 $.ajax  
 http://service.httppost  
 http://service.httpget  
  
案例：  
0、人工JS中提取URL信息  
1、从JS中提取到URL信息  
2、从JS中提取到敏感信息  
3、从JS中提取到配置信息  
4、从JS中Fuzz提取到更多  
5、WebPack打包器信息获取  
  
#项目插件工具：  
https://github.com/ffuf/ffuf  
用Go编写的快速 Web 模糊测试程序。  
  
https://github.com/gh0stkey/HaE  
HaE是一款网络安全（数据安全）领域下的框架式项目，采用了乐高积木式模块化设计理念，巧妙地融入了人工智能大模型辅助技术，实现对HTTP消息（包含WebSocket）精细化的标记和提取。  
  
https://github.com/Threezh1/JSFinder  
一款用作快速在网站的js文件中提取URL，子域名的工具  
  
https://github.com/pingc0y/URLFinder  
URLFinder是一款快速、全面、易用的页面信息提取工具  
用于分析页面中的js与url,查找隐藏在其中的敏感信息或未授权api接口  
  
https://github.com/rtcatc/Packer-Fuzzer  
针对Webpack等前端打包工具所构造的网站进行快速、高效安全检测的扫描工具  
  
https://github.com/shuanx/BurpAPIFinder  
攻防演练过程中，我们通常会用浏览器访问一些资产，但很多未授权/敏感信息/越权隐匿在已访问接口过html、JS文件等，通过该BurpAPIFinder插件我们可以：  
1、发现通过某接口可以进行未授权/越权获取到所有的账号密码、私钥、凭证  
2、发现通过某接口可以枚举用户信息、密码修改、用户创建接口  
3、发现登陆后台网址  
4、发现在html、JS中泄漏账号密码或者云主机的Access Key和SecretKey  
5、自动提取js、html中路径进行访问，也支持自定义父路径访问 ...  
  
https://github.com/GerbenJavado/LinkFinder  
功能类似于JSFinder，但JSFinder好久没更新了。  
  
https://github.com/ttstormxx/jjjjjjjjjjjjjs  
针对webpack站点，爬取网站JS文件，分析获取接口列表，自动结合指纹识别和fuzz获取正确api根，可指定api根地址（针对前后端分离项目，可指定后端接口地址），根据有效api根组合爬取到的接口进行自动化请求，发现未授权/敏感信息泄露，回显api响应，定位敏感信息、敏感文件、敏感接口。支持批量模式。支持需认证接口自动尝试bypass。  
  
https://github.com/momosecurity/FindSomething  
该工具是用于快速在网页的html源码或js代码中提取一些有趣的信息的浏览器插件，包括请求的资源、接口的url，请求的ip和域名，泄漏的证件号、手机号、邮箱等信息。  
  
https://github.com/ScriptKid-Beta/Unexpected\_information  
BurpSuite插件用来标记请求包中的一些敏感信息、JS接口和一些特殊字段，  
防止我们疏忽了一些数据包，使用它可能会有意外的收获信息。

#前置知识：  
1.传统访问：用户访问域名–>解析服务器IP–>访问目标主机  
2.普通CDN：用户访问域名–>CDN节点–>真实服务器IP–>访问目标主机  
3.带WAF的CDN：用户访问域名–>CDN节点（WAF）–>真实IP–>访问目标主机  
安全影响：主要对于信息收集时端口扫描和IP探针的相关真实信息存在有误等  
  
国内服务商：  
阿里云 百度云 七牛云   
又拍云 腾讯云 Ucloud  
360 网宿科技 ChinaCache  
  
国外服务商  
CloudFlare StackPath Fastly  
Akamai CloudFront Edgecast  
CDNetworks Google Cloud CDN  
CacheFly Keycdn Udomain CDN77  
  
#CDN配置：  
配置1：加速域名-需要启用加速的域名（子域名没配置）  
配置2：加速区域-需要启用加速的地区（国外地区没加速）  
配置3：加速类型-需要启用加速的资源  
配置4：其他SSL证书，DNS解析记录等（历史记录查询）  
  
#参考知识：  
超级Ping：http://www.17ce.com/  
超级Ping：https://ping.chinaz.com/  
接口查询：https://get-site-ip.com/  
接口查询：https://fofa.info/extensions/source  
国外请求：https://tools.ipip.net/cdn.php  
国外请求：https://boce.aliyun.com/detect/  
IP社区库：https://www.cz88.net/geo-public  
CDN厂商查询：https://cdn.chinaz.com/  
CDN厂商查询：https://tools.ipip.net/cdn.php  
全网扫描：https://github.com/Tai7sy/fuckcdn  
全网扫描：https://github.com/boy-hack/w8fuckcdn  
全网扫描：https://github.com/Pluto-123/Bypass\_cdn  
  
参考  
https://mp.weixin.qq.com/s/M\_tdPxhtOm9XPoVq721FPg  
https://mp.weixin.qq.com/s/zxEH-HMqKukmq7qXfrdnQQ  
常见方法：  
子域名，邮件系统，国外访问，证书查询，APP抓包，网络空间  
通过漏洞或泄露获取，扫全网，以量打量，第三方接口查询等  
1. 查询历史DNS记录：通过查询历史DNS记录，可以找到使用CDN前的IP地址。常用网站有DNSDB、微步在线、Netcraft、ViewDNS等。  
2. 查询子域名：一些网站可能只对主站或流量大的子站点使用CDN。通过查询子域名对应的IP地址，可以辅助找到网站的真实IP。常用工具包括微步在线、Dnsdb查询法、Google搜索等。  
3. 网络空间引擎搜索：利用网络空间搜索引擎（如Shodan、Fofa、ZoomEye）搜索特定关键词或特征，可以找到使用CDN的网站的真实IP地址。  
4. SSL证书：扫描互联网获取SSL证书，进而找到服务器的真实IP。Censys是一个用于搜索联网设备信息并扫描整个互联网的强大工具。  
5. HTTP标头：通过比较HTTP标头来查找原始服务器。例如，使用SecurityTrails平台搜索特定HTTP标头。  
6. 网站源代码：浏览网站源代码，寻找独特的代码片段或隐藏的IP地址信息。  
7. 国外主机解析域名：由于一些CDN服务在国外地区可能无法提供完整的保护，使用国外的主机直接访问目标网站可能会获取到真实IP地址。  
8. 遗留文件：如phpinfo页面泄露，可能会显示服务器的外网IP地址。9. 漏洞探针：利用网站上的漏洞（如SSRF漏洞）让VPS获取对方反向连接的IP地址。  
10. 网站邮件订阅：查看RSS邮件订阅的邮件源码，通常包含服务器的真实IP地址。  
11. 全网扫描：使用Zmap、Masscan等工具对整个互联网进行扫描，针对扫描结果进行关键字查找，获取网站真实IP。  
12. 解码F5 LTM负载均衡器：当服务器使用F5 LTM做负载均衡时，通过对set-cookie关键字的解码，可以获取到真实IP地址。  
13. 利用MX记录：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。  
14. 利用favicon.ico：查看网站的favicon.ico文件，通过其唯一的hash值进行识别，有时可以找到与网站相关的真实IP地址。  
15. 配置错误：检查CDN的配置细节，有时小小的配置错误就可能导致CDN防护被绕过。例如，CDN只配置了www域名，而直接访问非www域名可能获取到真实IP。  
16. 协议差异：如果站点同时支持http和https访问，而CDN只配置了https协议，那么访问http可能获取到真实IP。  
17. 利用网站功能：通过网站提供的某些功能（如邮箱注册、找回密码等），让网站主动暴露真实IP地址。  
18. 搜索引擎查询：利用搜索引擎的特定查询语法（如site:和inurl:），结合目标网站的信息，可能找到与网站相关的真实IP地址。  
19. 子域名挖掘机：输入域名即可基于字典挖掘子域名，通过查找子域名的IP地址来辅助找到网站的真实IP。20. IP库查询：使用IP库（如纯真数据库）来查询IP段，结合网站特征进行爆破，可能找到真实IP地址。  
  
#前置后置-CDN服务-识别加速&绑定访问  
超级Ping：http://17ce.com/  
超级Ping：https://ping.chinaz.com/  
拨测工具：https://boce.aliyun.com/detect/  
各地ping（出现多个IP即启用CDN服务）  
后置：绑定HOST访问解析（参考基础课CDN安全影响）  
  
#某应用-CDN绕过-主动漏洞&遗留文件&综合查询  
配置加速选项中只加速主域名，导致其他子域名未加速（解析IP可能同IP也可能C段）  
1、国外接口访问  
2、子域名解析IP  
3、综合查询（历史记录，证书记录等）  
综合接口查询：https://get-site-ip.com/  
综合接口查询：https://fofa.info/extensions/source  
使用网络空间&第三方功能集合查询判断  
  
#某应用-CDN绕过-主动漏洞&遗留文件&邮件系统  
1、漏洞如：SSRF RCE等  
利用漏洞让对方真实服务器主动出网连接，判断来源IP即真实IP  
2、遗留文件：phpinfo类似功能  
通过访问类似PHPINFO类似代码函数获取本地IP造成的地址泄漏  
3、邮箱系统：  
判断条件：发信人是当前域名邮件用户名  
-让他主动给你发：  
部署架设的邮件服务器如果向外部用户发送邮件的话，  
那么邮件头部的源码中会包含此邮件服务器的真实IP地址。  
常见的邮件触发点有：  
1、RSS订阅  
2、邮箱注册、激活处  
3、邮箱找回密码处  
4、产品更新的邮件推送  
5、某业务执行后发送的邮件通知  
6、员工邮箱、邮件管理平台等入口处的忘记密码  
-你给未知邮箱发：（需要自己的邮件服务器不能第三方）  
通过发送邮件给一个不存在的邮箱地址，因为该用户邮箱不存在，所以发送将失败，并且还会收到一个包含发送该电子邮件给你的服务器的真实IP通知。  
  
#某应用-CDN绕过-全网扫描  
案例1：结合关键字固定IP段找真实IP  
1、判断加速厂商  
2、IP库筛地址段  
3、配置范围扫描  
厂商查询：  
https://cdn.chinaz.com/  
https://tools.ipip.net/cdn.php  
工具项目：  
https://www.cz88.net/geo-public  
https://github.com/Tai7sy/fuckcdn  
  
案例2：ico筛选资产+特征端口找真实IP  
https://mp.weixin.qq.com/s/fAbKQsTE0jzUnMWqOkDjMA

ICO图标：  
1、某个应用系统的标示  
2、某个公司/机构/个人团队的标示  
  
#Python-开发框架-Django&Flask  
Django  
1、识别插件  
2、固定的数据包格式：Set-Cookie:expires=  
  
Flask  
1、识别插件  
2、固定的数据包格式：Etag: "flask X-Powered-By: Flask  
  
Tornado  
1、识别插件  
2、固定的数据包格式：Server: TornadoServer  
  
#JavaScript-开发框架-Vue&Node.js  
Vue  
1、识别插件  
2、JS语法和加载文件  
  
Node.js  
1、识别插件  
2、固定的数据包格式：ETag: W/"  
  
#PHP-开发框架-ThinkPHP&Laravel&Yii  
ThinkPHP：  
0、识别插件  
1、X-Powered-By: ThinkPHP  
2、CMS识别到源码体系TP开发  
  
Laravel：  
1、识别插件  
2、Set-Cookie中特征的格式 XSRF-TOKEN= laravel\_session=   
  
Yii:  
1、识别插件  
2、Set-Cookie中特征的格式 YII\_CSRF\_TOKEN=   
  
#Java-框架组件-Fastjson&Shiro&Solr&Spring  
52类110个主流Java组件和框架介绍：  
https://blog.csdn.net/agonie201218/article/details/125300729  
  
框架：  
Struts2  
一般使用struts2框架后缀带do或action，可以尝试进行利用  
  
SpringBoot  
1、默认web应用标签小绿叶图标  
2、通过springboot框架默认页面  
3、ICO源码体系采用SpringBoot开发  
  
组件：  
FastJson/Jackson  
在提交JSON数据包中修改测试：  
-FastJson组件会把01解析成1  
-Jackson组件在解析01时会抛出异常  
https://forum.butian.net/share/1679  
  
Shiro  
请求包的cookie中存在rememberMe字段。  
返回包中存在set-Cookie：remeberMe=deleteMe。  
请求包中存在rememberMe=x时，响应包中存在rememberMe=deleteMe。  
有时候服务器不会主动返回remeberMe=deleteMe，直接发包即可，将Cookie内容改为remember Me=1，若相应包有rememberMe=deleteMe，则基本可以确定网站apache shiro搭建的。  
  
Solr识别   
一般开放8983端口,访问页面也可以探针到  
  
#工具使用：  
https://github.com/projectdiscovery/nuclei

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)