Shodan自动化利用

phorse / 2018-02-26 23:00:52 / 浏览数 20671

本文由D0g3安全小组，@Sky3，@Zev3n，@Ph0rse，合作编写

Shodan是什么？

Shodan，是一个暗黑系的谷歌，作为一个针对网络设备的搜索引擎，它可以在极短的时间内在全球设备中搜索到你想找的设备信息。对于渗透工作者来说，就是一个辅助我

安全工作者的日常工作少不了跟进最新漏洞和使用实战靶机进行漏洞测试，漏洞信息我们大多可以通过Exploit-DB和HackNews来获取，那搜索可利用的实战靶机就可以通过

配合其组合参数，可以做到以下功能：

1. 批量搜索现有漏洞主机
2. 统计感染某木马的主机数量
3. 批量扫描登录入口，并使用弱口令字典进行爆破
4. 批量抓取shell
5. 使用自己的0day打遍天下

而本篇文章起到的作用，希望能够将这款工具的用途、使用方法、深入利用技巧以及实战应用，通俗易懂地介绍给读者们。使读者可以使用Shodan搜索引擎进行简单Web页

Shodan的工作原理

Shodan每隔一段时间就会对全球大约6亿主机进行端口扫描，通过对返回Banner信息的处理，识别特定主机，并进行分类储存。为了避免因政治、技术等原因导致的扫描信
当我们发出一条搜索请求，其背后的逻辑是这样的

那Banner是什么呢？
在探测端口时数据包里存在Banner信息

```
HTTP/1.1 200 OK
Server: nginx/1.2.19
Date: Sat, 13 Oct 2017 16:09:24 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 5322
Connection: keep-alive
```

这是一个普通的HTTP Banner，可以从上获知服务端使用的是1.2.19版本的nginx

西门子S7工控协议的Banner：

```
Copyright: Original Siemens Equipment
PLC name: S7_Turbine
Module type: CPU 313C
Unknown (129): Boot Loader           A
Module: 6ES7 313-5BG04-0AB0  v.0.3
Basic Firmware: v.3.3.8
Module name: CPU 313C
Serial number of module: S Q-D9U083642013
Plant identification:
Basic Hardware: 6ES7 313-5BG04-0AB0  v.0.3
```

其中能获取更多的信息

Shodan的Banner处理机制比较复杂，我们这里只需要知道探测端口是来往的数据包中包含Banner，并包含了主机的部分信息即可

Shodan的基本语法

简单语法

Shodan的参数有很多，这里只介绍简单的几种

hostname："主机或域名"
如 `hostname:"google''`

port："端口或服务"
如 `port:"21"`

ip : "ip地址"
如 ip : "168.205.71.64"

net："IP地址或子网"

如 `net:"210.45.240.0/24"`

vuln :指定漏洞的cve

如 vuln:CVE-2015-8869

但是这个命令最好搭配起来使用，如 country:CN vuln:CVE-2014-0160

os :"操作系统"

 如 os:"centOS"

isp："ISP供应商"

如 `isp:"China Telecom"`

product："操作系统/软件/平台"

如 `product:"Apache httpd"`

version："软件版本"

如 `version:"3.1.6"`

geo："经纬度"

如 `geo:"39.8779,116.4550"`

country`："国家"

如 country:"China"
country:"UN"

city："城市"
如 `city:"Hefei"`

org："组织或公司"
如 `org:"google"`

before/after："日/月/年"
如 before:"25/09/2017"
after:"25/09/2017"

asn : "自治系统号码"
如 asn:"AS2233"

查询指令实战组合实例

下面是一些组合实例：

搜索成都开放8080端口的主机

Apache city:"Chengdu" port:"8080"

IP为72.34.62.0——255的Linux主机

os:"linux" net:"72.34.62.0/24"

搜索网络摄像头

netcam country:"US" (网络摄像头)

点击Maps按钮还可以看到直观的图示

Shodan参数还有很多：

使用时可以查看中文文档

Shodan的命令行环境配置

Shodan除了在Web上直接操作，还可以在命令行进行操作，以及编写Python脚本去进行批量化利用

操作所需环境的搭建步骤为：

准备好Linux下Python3的环境

#安装shodan板块
pip install Shodan
#下载Shodan项目
git clone https://github.com/achillean/shodan-python.git

#执行安装
cd shodan-python
python setup.py install

如果安装过程中出现问题，可以移动到shodan-python目录下的bin目录，使用python命令操作

```
(test_py3) C:\Users\Prude\shodan-python\bin>python shodan
python shodan
```

账户权限不同能够调用API的次数也不同（付费账号每年黑色星期五打折），因此在使用前，需要使用自己的API Key进行初始化
在Web网站的个人主页可以看到自己的API Key

```
shodan init API_Key
```

host-查看指定主机的相关信息，如地理位置信息、开放端口、可能存在的漏洞等信息。

```
shodan host 220.181.111.188
```

search-直接将查询结果展示在命令行中，默认情况下只显示IP、端口号、主机名和HTTP数据。也可以通过使用 –fields
来自定义显示内容，例如，我们只显示IP、端口号、主机名：

```
shodan search --fields ip_str,port,org,hostnames apache
```

也可以使用download参数把查询结构下载下来

```
shodan download apche-data apche
```

parse可以将结构解析为Json数据，多用于Python脚本中
shodan parse --fields ip_str,port,org --separator , apache-data.json.gz

stats-汇总查询结果，方便统计
例如查看weblogic主机在全球的分布情况

```
shodan stats weblogic
```

在Python脚本中使用Shodan API时，可以通过查看官方文档进行学习

Python脚本自动化利用

上面提到了使用Shodan
API的开发文档，实战过程中为了方便起见，大可编写适合自己的脚本工具，进行批量化搜索和利用。本文的作者之一@Zev3n编写了一个自动化的Python工具：Shodan_S

该工具基于Shodan API的辅助查询工具，功能如下：

```
root@kali:~/Desktop/shodan# ./Shodan_So.py -h
usage: Shodan_So.py [-search Apache] [-f ips.txt] [-ip 217.140.75.46]
                    [-iprg 217.140.75.46/24] [--hostnameonly] [--history]
                    [--page 1] [--list_ip] [--list_ip_port]

Shodan_So - Search Assistant: Searching shodan via API. --By: Zev3n


optional arguments:
 -search Apache        when searching Shodan for a string.
 -f ips.txt            Using THe Ips List - File containing IPs to search
                       shodan for.
 -ip 217.140.75.46     Shodan Host Search against IP & return results from
```

```
                      Shodan about a specific IP.
 -iprg 217.140.75.46/24
                      Used to return results from Shodan about a specific
                      CIDR to IP range .
 --hostnameonly       Only provide results with a Shodan stored hostname.
 --history            Return all historical banners.
 --page 1             Page number of results to return (default 1 (first
                      page)).
 --list_ip            Singled out IP address from query results.
 --list_ip_port       Singled out IP address with port from query results.
```

其中，-search命令后面接你要查询的内容，语法和在官网查询是一样的。不过需要注意的是，查询的内容里如果包含了引号或者空格或者其他特殊符号，需要用引号在查询
-search "apache country:"US""

```
_______           __              ____
 / __/ /  ___  ___/ /__ _____     / __/__
_\ \/ _ \/ _ \/ _  / _ `/ _ \   _\ \/ _ \
/___/_//_/\___/\_,_/\_,_/_//_/__/___/\___/
                           /___/
[*] Searching Shodan...

Total number of results back: 7546877

...
```

可以看到，本工具的查询效果和网页查询几乎是一样的。

这里需要着重介绍的参数还有--list_ip
在查询命令后面加上这个参数之后，脚本会自动提取查询结果中的ip地址并打印。我们可以利用输出重定向将查询的ip列表保存到txt文件中，以供后续利用。例如：
./Shodan_So.py -search "apache country:"US"" --list_ip >> ips.txt
另外，加上--list_ip_port参数后是返回IP■■:■■形式的地址，以供后续不同场景的利用。
-ip参数则是指定你要查询的IP地址，会分端口返回一些简要的banner以供参考
比如这里去搜索百度首页的IP
./Shodan_So.py -ip 119.75.217.109

```
_______           __              ____
 / __/ /  ___  ___/ /__ _____     / __/__
_\ \/ _ \/ _ \/ _  / _ `/ _ \   _\ \/ _ \
/___/_//_/\___/\_,_/\_,_/_//_/__/___/\___/
                           /___/
[*] Searching Shodan for info about 119.75.217.109...

***** RESULT *****

IP: 119.75.217.109
Organization: Beijing Baidu Netcom Science and Technology Co.
Operating System: None


*** PORT 1***
Port: 80
Banner:
HTTP/1.1 200 OK
Date: Fri, 23 Feb 2018 06:45:10 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: Keep-Alive
Vary: Accept-Encoding

...


*** PORT 2***
Port: 443
Banner:
HTTP/1.1 200 OK
Bdpagetype: 1

...
```

```
X-Powered-By: HPHP
X-Ua-Compatible: IE=Edge,chrome=1
Transfer-Encoding: chunked



[+] Host 119.75.217.109 Found ports Record:2
80 443
```

`-iprg`参数接受的则是CIDR地址块，可实现指定子网范围内主机的查询
`-f`参数接受的是一个ip地址的文本文档，不同地址用换行隔开，本工具会依次读取文档内的ip地址并依次查询

另外的参数在这里就不再进行介绍了，大家可以自行探索

Shodan_So的实战使用

在这里我们查询中国存在心脏滴血漏洞风险的服务器
输入命令：`./Shodan_So.py -search "vuln:CVE-2014-0160 country:CN"`

可以看到满足查询条件的共有7940个，这里只显示了100个结果。这里返回了每个符合条件的记录的概要信息。虽然这些概要信息有时候能给我们一些帮助，但是批量利用的
那么，如何能够轻松的批量利用呢？这里我们只需要在查询的命令之后加上一个参数`--list_ip`，然后输出重定向为txt文档，就可以导出查询到的ip列表，配合漏洞的批量
`./Shodan_So.py -search "vuln:CVE-2014-0160 country:CN" --list_ip >> ip.txt`

打开漏洞利用工具，选择载入，将脚本导出的ip.txt导入到目标列表，点击开始攻击

稍等一段时间，待扫描结束之后可以看到，在100个主机中，算上各种端口，共有108处存在漏洞，且泄露的内存数据被保存到了CSHS文件夹内。

再配合一个python脚本来自动搜索泄露的内存中的敏感信息（比如cookie、管理员的帐号密码等），威力不可小觑

抓到内存中cookie信息的服务器列表

内存中的cookie信息

配合burpsuite抓包修改cookie，可实现绕过管理员帐号密码直接登录

Shodan环境操作的视频教程

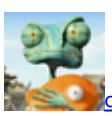
上一篇：在《WAF攻防之SQL注入篇》中几... 下一篇：某CMS V5.7 SP2 后台G...

1. 5 条回复

phorse 2018-02-26 23:06:09

最后的演示视频没显示出来~贴个链接哈：http://ovc15d5kr.bkt.clouddn.com/VID_20180222_113411.mp4

0 回复Ta



orich1 2018-02-26 23:29:33

;)

0 回复Ta


balisong 2018-02-28 10:20:29

nice.

0 回复Ta


curtain 2018-02-28 16:30:41

然鹅，并没有key。 :(

0 回复Ta


1558893025510967 2018-03-09 19:02:06

你好，我的为什么显示不能链接Shodan

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

目录