

信息收集与漏洞扫描 实 验 报 告



实验名称 信息收集与漏洞扫描

班 级 信安 20-2

姓 名 李天昊

学 号 20101110201

指导教师 徐 刚

2020 年 12 月 18 日

实验一 信息收集与漏洞扫描

一、实验目的

通过本实验初步了解黑客入侵和攻击的方法以及一般的应对测量,掌握常见工具的基本应用,包括如下几个方面:

1. 了解网络主机信息收集的方法和工具;
2. 了解安全扫描技术;
3. 了解网络搜索引擎对系统安全的威胁。

二、实验环境

1. 攻击方 (笔记本电脑: Thinkpad T480s)

DEVICE NAME : Thinkpad T480s

PROCESSOR : Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz

SYSTEM TYPE : 64-bit operating system,x64-based processor

SYSTEM EDITION : Windows 10 Professional

VERSION : 20H2

OS BUILD : 19042.685

2. 被攻击方 (阿里云 ECS 实例: 1vCPU 2.0GB)

DEVICE NAME : iZbp16c0oppuzs5g1l8p2kZ

PROCESSOR : Intel(R) Xeon(R) CPU E5-2682 V4 @ 2.50GHz * 1

SYSTEM INFORMATION :

- CentOS Linux release 7.8.2003 (Core)
- Linux version 3.10.0-1127.19.1.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)) #1 SMP Tue Aug 25 17:23:54 UTC 2020
- Linux izbp16c0oppuzs5g1l8p2kz 3.10.0-1127.19.1.el7.x86_64 #1 SMP Tue Aug 25 17:23:54 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux

HOST LOCATION : Eastern China 1 – HANG ZHOU CITY

PUBLIC IP : 47.97.4.98

DOMAIN NAME : is202.top (*information security 20 - 2 class website*)

三、实验要求

通过本实验初步了解黑客入侵和攻击的方法以及一般的应对策略，掌握常见工具的基本应用，包括如下几个方面：

1. 了解网络主机信息收集的方法和工具；
2. 了解安全扫描技术；
3. 了解网络搜索引擎对系统安全的威胁。

四、实验步骤和结果

1. 利用各种工具软件进行信息收集

(1) ping 命令

因为本次实验的被攻击方绑定了域名 is202.top，所以下面直接对域名 is202.top 进行 ping 操作。实验在 Windows Terminal 中利用 Windows Powershell 进行操作

(a) 终端输入 ping 查看该命令详细参数说明

```
PS C:\Users\李天昊> ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
           [-r count] [-s count] [[-j host-list] | [-k host-list]]
           [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
           [-4] [-6] target_name

Options:
    -t           Ping the specified host until stopped.
                  To see statistics and continue - type Control-Break;
                  To stop - type Control-C.
    -a           Resolve addresses to hostnames.
    -n count     Number of echo requests to send.
    -l size      Send buffer size.
    -f           Set Don't Fragment flag in packet (IPv4-only).
    -i TTL       Time To Live.
    -v TOS       Type Of Service (IPv4-only. This setting has been deprecated
                  and has no effect on the type of service field in the IP
                  Header).
    -r count     Record route for count hops (IPv4-only).
    -s count     Timestamp for count hops (IPv4-only).
    -j host-list Loose source route along host-list (IPv4-only).
    -k host-list Strict source route along host-list (IPv4-only).
    -w timeout   Timeout in milliseconds to wait for each reply.
    -R           Use routing header to test reverse route also (IPv6-only).
                  Per RFC 5095 the use of this routing header has been
                  deprecated. Some systems may drop echo requests if
                  this header is used.
    -S srcaddr   Source address to use.
    -c compartment Routing compartment identifier.
    -p           Ping a Hyper-V Network Virtualization provider address.
    -4           Force using IPv4.
    -6           Force using IPv6.
```

(b) ping 被攻击方，加入参数-a，将地址解析为主机名

```
PS C:\Users\李天昊> ping -a is202.top

Pinging is202.top [47.97.4.98] with 32 bytes of data:
Reply from 47.97.4.98: bytes=32 time=28ms TTL=51
Reply from 47.97.4.98: bytes=32 time=29ms TTL=51
Reply from 47.97.4.98: bytes=32 time=29ms TTL=51
Reply from 47.97.4.98: bytes=32 time=28ms TTL=51

Ping statistics for 47.97.4.98:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 28ms, Maximum = 29ms, Average = 28ms
```

- 得到的信息：
 - ip 地址：47.97.4.98
 - 回应报文大小：32bytes
 - 平均回应所花费时间（平均网络延迟）：28ms（北京-杭州）
 - 生存时间 TTL：51（可以初步判断被攻击方安装了 linux 系统）

(2) Tracert 命令

Tracert 为路由跟踪程序，跟踪从本地开始到达某一目标地址所经过的路由设备，并显示出这些路由设备的 IP、连接时间等信息。请求超时是由于安全考虑。

```
PS C:\Users\李天昊> tracert is202.top

Tracing route to is202.top [47.97.4.98]
over a maximum of 30 hops:

  1  13 ms  8 ms  1 ms  bogon [192.168.1.1]
  2   2 ms  1 ms  4 ms  bogon [10.70.0.1]
  3  11 ms  5 ms  *    123.126.26.77
  4   5 ms  6 ms  6 ms  125.33.186.85
  5  29 ms  28 ms  29 ms  219.158.100.174
  6  27 ms  29 ms  *    124.160.189.98
  7  25 ms  25 ms  25 ms  124.160.190.222
  8   *    *    *    Request timed out.
  9  27 ms  30 ms  28 ms  119.38.213.125
 10   *    *    *    Request timed out.
 11   *    *    *    Request timed out.
 12  28 ms  28 ms  27 ms  47.97.4.98

Trace complete.
```

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
123.125.195.0	123.126.67.255	中国 北京市 北京市	联通	255.252.0.0	123.124.0.0/14

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
125.33.184.0	125.33.191.255	中国 北京市 北京市	联通	255.255.248.0	125.33.184.0/21

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
219.158.0.0	219.158.255.255	中国 北京市 北京市	联通	255.255.0.0	219.158.0.0/16

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
124.160.160.0	124.160.195.255	中国 浙江省 杭州市	联通	255.255.128.0	124.160.128.0/17

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
124.160.160.0	124.160.195.255	中国 浙江省 杭州市	联通	255.255.128.0	124.160.128.0/17

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
124.160.160.0	124.160.195.255	中国 浙江省 杭州市	联通	255.255.128.0	124.160.128.0/17

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
119.38.208.0	119.38.218.255	中国 浙江省 杭州市	阿里云 数据中心	255.255.240.0	119.38.208.0/20

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
47.96.0.0	47.99.255.255	中国 浙江省 杭州市	阿里云 数据中心	255.252.0.0	47.96.0.0/14

- 得到的信息:

- 从攻击方到被攻击方经过了 12-1=11 跳
- 从数据包从本地局域网出发后, 经过联通网络, 直连杭州阿里云数据中心
- 每个跃点的 windows 子网掩码和 linux 子网掩码

(3) net 命令

(a)使用 net view 命令查看远程主机的共享资源, 命令格式 net view [\\ip](#)

```
PS C:\Users\李天昊> net view 47.97.4.98
System error 53 has occurred.

The network path was not found.
```

原因分析:

- 如果 Microsoft 网络的文件和打印机共享被禁用或未安装, 可能会出现此问题。

- 得到的信息:

- 远程主机没有开放共享资源或设置了防火墙禁止未授权 ip 访问共享资源
(注: 我确实在服务器部署了防火墙并添加了相关规则)

(b)使用 net use 命令把远程主机的某个资源映射为本地盘符, 结合其他命令就可以达到入侵的效果, 命令格式 net use x:\\ip\sharename

因为步骤(a)显示被攻击方并没有开启相关服务, 故不进行此步骤。

(c)使用 net time 命令查看远程主机当前的时间, 入侵成功后进一步渗透时需要远程主机当前的时间

```
PS C:\Users\李天昊> net time \\47.97.4.98
The service has not been started.

More help is available by typing NET HELPMSG 2184.
```

- 得到的信息:

- 被攻击方并没有开启相关服务或设置了防火墙禁止相关操作

(d)使用 net user 命令查看和账户有关的信息，包括新建账户、删除账户、查看特定账户、顾客账户，以上信息对入侵很有利，为账户克隆提供了前提

```
PS C:\Users\李天昊> net user

User accounts for \\THINKPADI480S

----- 中 -----
Administrator          DefaultAccount          Guest
WDAGUtilityAccount     zhaox                  李天昊
The command completed successfully.
```

(e)使用 ipconfig 命令显示所有 TCP/IP 网络配置信息，刷新动态主机配置协议 (DHCP) 和域名系统 (DNS) 设置

```
PS C:\Users\李天昊> ipconfig

Windows IP Configuration

Ethernet adapter 以太网:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter 本地连接* 9:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter 本地连接* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::a134:5bcb:dd2f:ee8b%43
    IPv4 Address. . . . . : 192.168.93.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::dc98:64c0:397b:b1f6%9
    IPv4 Address. . . . . : 192.168.244.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

```
Wireless LAN adapter WLAN:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::a18e:1c72:6da3:8e3f%33
    IPv4 Address. . . . . : 192.168.1.106
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Ethernet adapter 以太网 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Mobile Broadband adapter 手机网络:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2409:8900:4900:1349:d803:a4df:708a:ebb5
    Temporary IPv6 Address. . . . . : 2409:8900:4900:1349:fd0:67b:8d6a:c2b0
    Link-local IPv6 Address . . . . . : fe80::1:2:b489:4056%23
    Link-local IPv6 Address . . . . . : fe80::d803:a4df:708a:ebb5%23
    IPv4 Address. . . . . : 10.146.129.3
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1:2:b489:4001%23
                                fe80::5%23
                                10.146.129.1

Ethernet adapter 蓝牙网络连接 2:

    Media State . . . . . 英 . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

(f) 使用 nslookup 命令查询 DNS 记录，查询域名解析是否正常，可以在网络故障时来诊断网络问题

```
PS C:\Users\李天昊> nslookup
Default Server:  c1-tucheng-pengbs-ns1
Address:  202.106.46.151

> is202.top
Server:  c1-tucheng-pengbs-ns1
Address:  202.106.46.151

Non-authoritative answer:
DNS request timed out.
    timeout was 2 seconds.
Name:    is202.top
Address:  47.97.4.98

> baidu.com
Server:  c1-tucheng-pengbs-ns1
Address:  202.106.46.151

Non-authoritative answer:
Name:    baidu.com
Addresses:  220.181.38.148
            39.156.69.79

> google.com
Server:  c1-tucheng-pengbs-ns1
Address:  202.106.46.151

Non-authoritative answer:
Name:    google.com
Address:  8.7.198.46
> |
```

有意思的事情来了：发现域名 is202.top 的 DNS 响应超时，这时候我想起当时使用的是 Cloudflare（国外服务商）的 DNS 域名解析服务器（名称服务器），猜测可能是因为这个导致了响应超时，为了验证这一猜想，进行了如下测试：

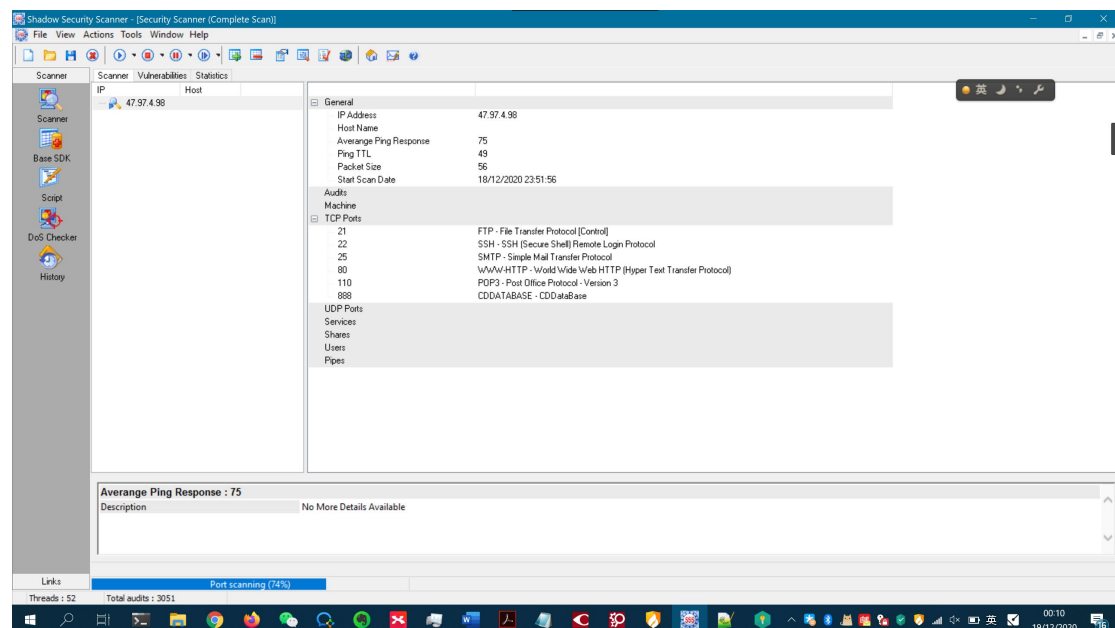
```
> is202.top
Server:  c1-xfdj-pengbs-ns1
Address:  202.106.46.151
Non-authoritative answer:
Name:    is202.top
Address:  47.97.4.98
测试 1: 挂载美国中继服务器的网络

> is202.top
Server:  c1-xfdj-pengbs-ns1
Address:  202.106.46.151
Non-authoritative answer:
DNS request timed out.
    timeout was 2 seconds.
Name:    is202.top
Address:  47.97.4.98
测试 2: 直连网络
```

结果显而易见，这里不作过多解释。

2. 扫描目标主机的开放端口、服务与协议

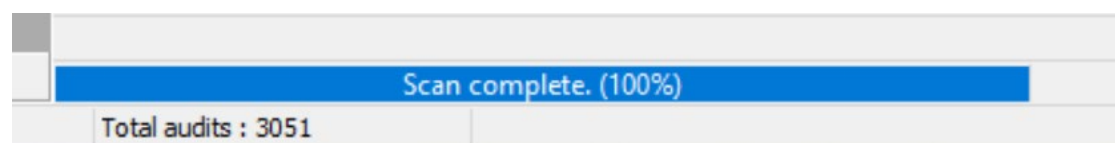
(1) Shadow Security Scanner



TCP Ports	
21	FTP - File Transfer Protocol [Control]
22	SSH - SSH (Secure Shell) Remote Login Protocol
25	SMTP - Simple Mail Transfer Protocol
80	WWW-HTTP - World Wide Web HTTP (Hyper Text Transfer Protocol)
110	POP3 - Post Office Protocol - Version 3
888	CDDATABASE - CDDatabase
8888	SiteScope - SiteScope Remote Server Monitoring
UDP Ports	

21 : FTP - File Transfer Protocol [Control]	
Banner	220----- Welcome to Pure-FTPd [privsep] [TLS] ----- 220-You are user number 1 of 50 allowed. 220-Local time is now 23:53. Server port: 21. 220-This is a private system - No anonymous login 220-IPv6 connections are also welcome on this server. 220 You will be disconnected after 15 minutes of inactivity.
Protocols	FTP
SYST	215 UNIX Type: L8

22 : SSH - SSH (Secure Shell) Remote Login Protocol	
Banner	SSH-2.0-OpenSSH_7.4



(2) Shodan (website)

其实以前我一直使用的并不是 Shadow Security Scanner，而是一款网页端的在线扫描工具，这里也会进行一次实验操作，对 is202.top 进行漏洞扫描。

(a) 信息概览

General Information	
Country	China
Organization	Hangzhou Alibaba Advertising Co.,Ltd.
ISP	Hangzhou Alibaba Advertising Co.,Ltd.
ASN	AS37963

(b) 开放端口

这里扫描出来了三个开放的端口，经验告诉我，21 端口用于 FTP (File Transfer Protocol, 文件传输协议) 服务，80 端口用于网页服务，而 8888 端口恰好是访问 BTpanel 的默认端口！（可以知道，站长很懒，并没有修改默认端口）

Open Ports

21808888

// 8888 / TCP

-164026536 | 2020-11-25T13:10:43.730601

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 802
Set-Cookie: SESSIONID=82a099cd-b831-4791-863e-0bdddef0db28.gb-r1BEh0toremTTjnw1fEURmh8; Expires=Fri, 25-Dec-2020 13:10:43 GMT; HttpOnly; Path=/
Date: Wed, 25 Nov 2020 13:10:43 GMT

由于在网页端查看更多信息需要充值，我又不想花钱，所以我再 kali 子系统中安装了 shodan，安装及操作过程如下：

(3) shodan(kali wsl)

(a) 首先更换国内镜像（这里选择阿里源）

(b) 更新包，得到超级权限后依次使用 apt-get update 和 apt-get upgrade 命令

```
(root@ThinkpadT480s)-[/etc/apt]
# ls
apt.conf.d auth.conf.d preferences.d sources.list sources.list.d trusted.gpg.d
(root@ThinkpadT480s)-[/etc/apt]
# vi sources.list
(root@ThinkpadT480s)-[/etc/apt]
# apt-get update
Get:1 http://mirrors.aliyun.com/kali kali-rolling InRelease [30.5 kB]
Get:2 http://mirrors.aliyun.com/kali kali-rolling/main amd64 Packages [17.1 MB]
Get:3 http://mirrors.aliyun.com/kali kali-rolling/non-free amd64 Packages [202 kB]
Get:4 http://mirrors.aliyun.com/kali kali-rolling/contrib amd64 Packages [105 kB]
Fetched 17.5 MB in 14s (1,224 kB/s)
Reading package lists... Done
(root@ThinkpadT480s)-[/etc/apt]
# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  cpp-10 gcc-10-base libgcc-s1 libstdc++6
The following packages will be upgraded:
  apt apt-utils bash bsdxattrutils bsdtar dash dmsetup fdisk gcc-9-base init init-system-helpers iptables
  isc-dhcp-client isc-dhcp-common kali-defaults libapparmor1 libapt-pkg6.0 libaudit1 libblkid1 libbrotli1 libc-bin
  libc-l10n libc6 libcapi-ng0 libdevmapper1.02.1 libelf1 libfdisk1 libffi7 libgcrpt20 libgmp10 libgssapi-krb5-2
  libidn2-0 libip4tc2 libip6tc2 libk5crypto3 libkrb5-3 libkrb5support0 libldap-2.4-2 libldap-common
  libmail-gettext-perl libmaxminddb0 libmount1 libncurses6 libncursesw6 libnewt0.52 libnftables2-14 libpcre2-8-0
  libpython3.9-minimal libpython3.9-stdlib libseccomp2 libselinux1 libsemanage1 libsmartcols1 libssh2-1 libsystemd0 libtext-charwidth-perl
  libtext-iconv-perl libtinfo6 libudev1 libuuid1 libxml2 libxtables12 locales-all mlocate mount nano ncurses-base
```

(c) 安装 git，使用命令 apt-get install -y git

```
(root@ThinkpadT480s)-[/etc/apt]
# apt-get install -y git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man libcbor0 libcurl3-gnutls liberror-perl libexpat1 libfido2-1 libgdbm-compat4 libgdbm6 libperl5.32 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6
  libxext6 libxmuu1 openssl-client patch perl perl-modules-5.32 xauth
Suggested packages:
  gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn gdbm-l10n keychain
  libpam-ssh monkeysphere ssh-keypass ed diffutils-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  git git-man libcbor0 libcurl3-gnutls liberror-perl libexpat1 libfido2-1 libgdbm-compat4 libgdbm6 libperl5.32 libx11-6 libx11-data libxau6 libxcb1
  libxdmcp6 libxext6 libxmuu1 openssl-client patch perl perl-modules-5.32 xauth
0 upgraded, 22 newly installed, 0 to remove and 4 not upgraded.
Need to get 17.5 MB of archives.
After this operation, 94.4 MB of additional disk space will be used.
Get:1 http://mirrors.aliyun.com/kali kali-rolling/main amd64 perl-modules-5.32 all 5.32.0-5 [2,821 kB]
Get:2 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libgdbm6 amd64 1.18.1-5.1 [64.4 kB]
Get:3 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libgdbm-compat4 amd64 1.18.1-5.1 [44.4 kB]
Get:4 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libperl5.32 amd64 5.32.0-5 [4,119 kB]
Get:5 http://mirrors.aliyun.com/kali kali-rolling/main amd64 perl amd64 5.32.0-5 [293 kB]
Get:6 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libcbor0 amd64 0.5.0+dfsg-2 [24.0 kB]
Get:7 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libfido2-1 amd64 1.5.0-2 [52.3 kB]
Get:8 http://mirrors.aliyun.com/kali kali-rolling/main amd64 openssl-client amd64 1.8.4p1-3 [929 kB]
Get:9 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libcurl3-gnutls amd64 7.72.0-1 [333 kB]
Get:10 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libexpat1 amd64 2.2.10-1 [96.9 kB]
Get:11 http://mirrors.aliyun.com/kali kali-rolling/main amd64 liberror-perl all 0.17029-1 [31.0 kB]
Get:12 http://mirrors.aliyun.com/kali kali-rolling/main amd64 git-man all 1:2.29.2-1 [1,889 kB]
Get:13 http://mirrors.aliyun.com/kali kali-rolling/main amd64 git amd64 1:2.29.2-1 [5,373 kB]
Get:14 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libxau6 amd64 1:1.0.8-1+b2 [19.9 kB]
Get:15 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libxdmcp6 amd64 1:1.1.2-3 [26.3 kB]
Get:16 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libxcb1 amd64 1.14-2 [139 kB]
Get:17 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libx11-data all 2:1.6.12-1 [311 kB]
Get:18 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libx11-6 amd64 2:1.6.12-1 [770 kB]
Get:19 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libxext6 amd64 2:1.3.3-1+b2 [52.5 kB]
Get:20 http://mirrors.aliyun.com/kali kali-rolling/main amd64 libxmuu1 amd64 2:1.1.2-2+b3 [23.9 kB]
Get:21 http://mirrors.aliyun.com/kali kali-rolling/main amd64 patch amd64 2.7.6-6 [126 kB]
```

(d) 找到 github 上的相关开源项目并使用 git 命令安装 shodan

```
(root@ThinkpadT480s)-[/etc/apt/shodan-python]
# ls
AUTHORS      dist          LICENSE      requirements.txt  setuptools-0.6c11.tar.gz  shodan-python
build        docs          MANIFEST.in  setup.py          shodan                    tests
CHANGELOG.md get-pip.py    README.rst    setuptools-0.6c11  shodan.egg-info          tox.ini
(root@ThinkpadT480s)-[/etc/apt/shodan-python]
# shodan -h
Usage: shodan [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  alert      Manage the network alerts for your account
  convert    Convert the given input data file into a different format.
  count      Returns the number of results for a search
  data       Bulk data access to Shodan
```

(e) 执行命令 `shodan host 47.94.4.98`, 调用 API 扫描被攻击方主机, 结果如下:
发现该服务器存在大量的 CVE 漏洞, 并探测到 SSL 版本

```
(root@ThinkpadT480s)-[/etc/apt/shodan-python]
# shodan host 47.94.4.98
47.94.4.98
Country: China
Organization: Hangzhou Alibaba Advertising Co.,Ltd.
Updated: 2020-12-20T22:07:00.710770
Number of open ports: 3
Vulnerabilities: CVE-2018-10549 CVE-2014-5459 CVE-2018-10545 CVE-2018-10547 CVE-2018-10546 CVE-2011-0755 CVE-2013-1635 CVE-2019-9638 CVE-2011-4885 CVE-2018-17082 CVE-2018-10548 CVE-2018-19520 CVE-2018-19396 CVE-2016-7478 CVE-2012-2376 CVE-2011-1092 CVE-2012-2143 CVE-2019-9023 CVE-2012-2336 CVE-2014-2497 CVE-2012-1171 CVE-2019-9639 CVE-2013-4635 CVE-2011-0708 CVE-2011-1468 CVE-2011-1469 CVE-2011-0421 CVE-2012-2688 CVE-2013-4248 CVE-2019-9637 CVE-2011-1467 CVE-2011-1464 CVE-2017-16642 CVE-2013-2110 CVE-2018-20783 CVE-2011-1466 CVE-2018-14883 CVE-2018-19395 CVE-2019-6977 CVE-2012-0057 CVE-2019-9641 CVE-2012-2386 CVE-2006-7243 CVE-2011-4718 CVE-2012-1172 CVE-2012-2311 CVE-2014-0237 CVE-2015-8994 CVE-2012-1823 CVE-2018-19935 CVE-2010-4699 CVE-2014-9427 CVE-2014-0238 CVE-2010-3870 CVE-2019-9020 CVE-2019-9021 CVE-2012-0789 CVE-2012-0788 CVE-2019-9024 CVE-2012-3365 CVE-2011-1470 CVE-2018-15132 CVE-2013-1643
Ports:
  21/tcp
  80/tcp Apache httpd
  443/tcp Apache httpd
  |-- SSL Versions: -SSLv2, -SSLv3, -TLSv1, -TLSv1.3, TLSv1.1, TLSv1.2
(root@ThinkpadT480s)-[/etc/apt/shodan-python]
# |
```

● 得到的信息: (非常关键)

CVE-2018-10549 CVE-2014-5459 CVE-2018-10545 CVE-2018-10547 CVE-2018-10546 CVE-2011-0755 CVE-2013-1635 CVE-2019-9638 CVE-2011-4885 CVE-2018-17082 CVE-2018-10548 CVE-2018-19520 CVE-2018-19396 CVE-2016-7478 CVE-2012-2376 CVE-2011-1092 CVE-2012-2143 CVE-2019-9023 CVE-2012-2336 CVE-2014-2497 CVE-2012-1171 CVE-2019-9639 CVE-2013-4635 CVE-2011-0708 CVE-2011-1468 CVE-2011-1469 CVE-2011-0421 CVE-2012-2688 CVE-2013-4248 CVE-2019-9637 CVE-2011-1467 CVE-2011-1464 CVE-2017-16642 CVE-2013-2110 CVE-2018-20783 CVE-2011-1466 CVE-2018-14883 CVE-2018-19395 CVE-2019-6977 CVE-2012-0057 CVE-2019-9641 CVE-2012-2386 CVE-2006-7243 CVE-2011-4718 CVE-2012-1172 CVE-2012-2311 CVE-2014-0237 CVE-2015-8994 CVE-2012-1823 CVE-2018-19935 CVE-2010-4699 CVE-2014-9427 CVE-2014-0238 CVE-2010-3870 CVE-2019-9020 CVE-2019-9021 CVE-2012-0789 CVE-2012-0788 CVE-2019-9024 CVE-2012-3365 CVE-2011-1470 CVE-2018-15132 CVE-2013-1643

=====现在, 已经完成了一次基础的信息收集=====

补充:

这个软件突然让我想起我在初学 python 时候 (大概是 2018 年) 写过的一个端口扫描程序, 下面将会附上源代码与运行时截图:

- Python3 源代码:

```
#TCP server scanner
#UTF-8
import socket
import struct
import os

print("TCP Host Netscanener ver.1.2 (By LTH)")

#初始化
tgtHostS = str(input("Enter host ip START (IPv4) :"))
tgtHostE = str(input("Enter host ip END (IPv4) :")) + 1
tgtPortS = int(input("Enter host port START :"))
tgtPortE = int(input("Enter host port END :")) + 1
f = 0
c = 0

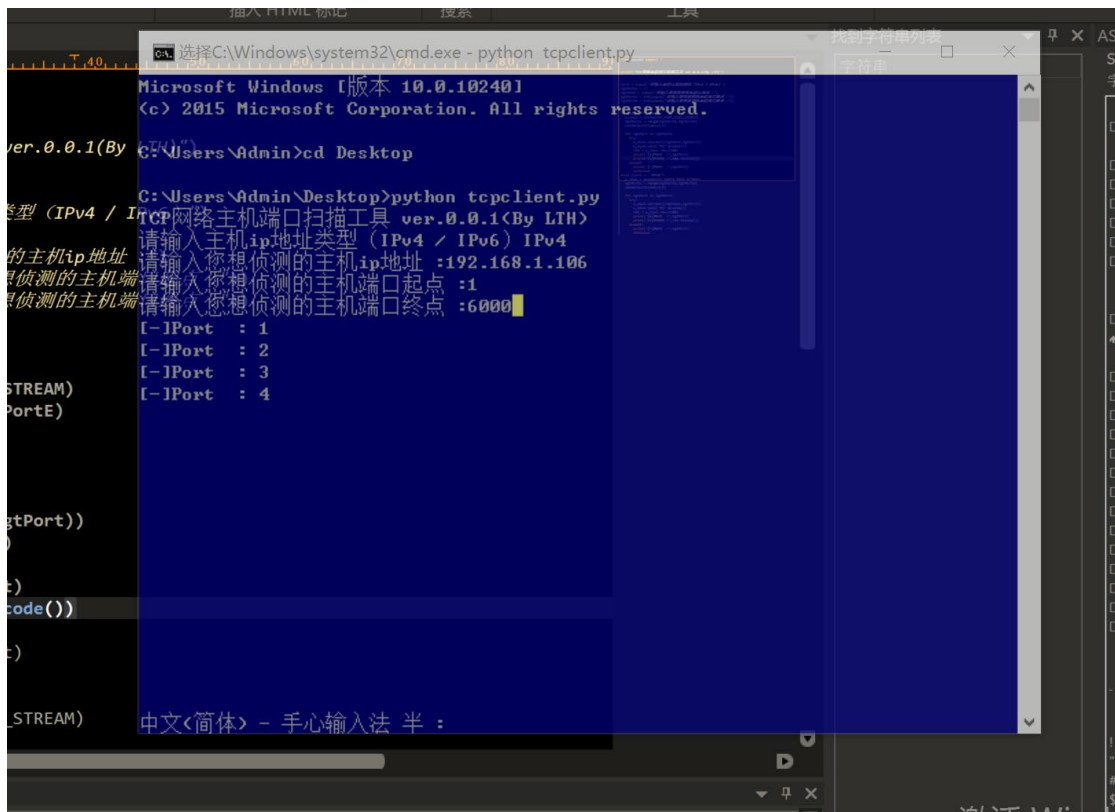
#ip -> num
num_tgtHostS = socket.ntohl(struct.unpack("I",socket.inet_aton(str(tgtHostS)))[0])
num_tgtHostE = socket.ntohl(struct.unpack("I",socket.inet_aton(str(tgtHostE)))[0])
print("[START] ip -> num :",num_tgtHostS)
print("[E N D] ip -> num :",num_tgtHostE)

#net
f = open("Scan.txt")
for i in range(num_tgtHostS,num_tgtHostE+1):
    import socket
    tgtHost = socket.inet_ntoa(struct.pack('I',socket.htonl(i)))
    print("Host ip detecting : ",tgtHost)
    print("")
    from socket import *

    c_sock = socket(AF_INET,SOCK_STREAM)
    tgtPorts = range(tgtPortS,tgtPortE)
    setdefaulttimeout(1)

    for tgtPort in tgtPorts:
        try:
            c_sock.connect((tgtHost,tgtPort))
            c_sock.send("hi".encode())          #编码并发送数据
            res = c_sock.recv(100)              #接收数据
            TP = res.decode()
            print("[+]Port :",tgtPort)
            print("[+]Proto :",res.decode())    #数据解码
            seek(0,2)
            a.write('from: ',tgtHost,' port: ',tgtPort,' ',TP,'\n')
            close()
        except:
            print("[-]Port :",tgtPort)
            continue
```

- 用户输入扫描指令信息

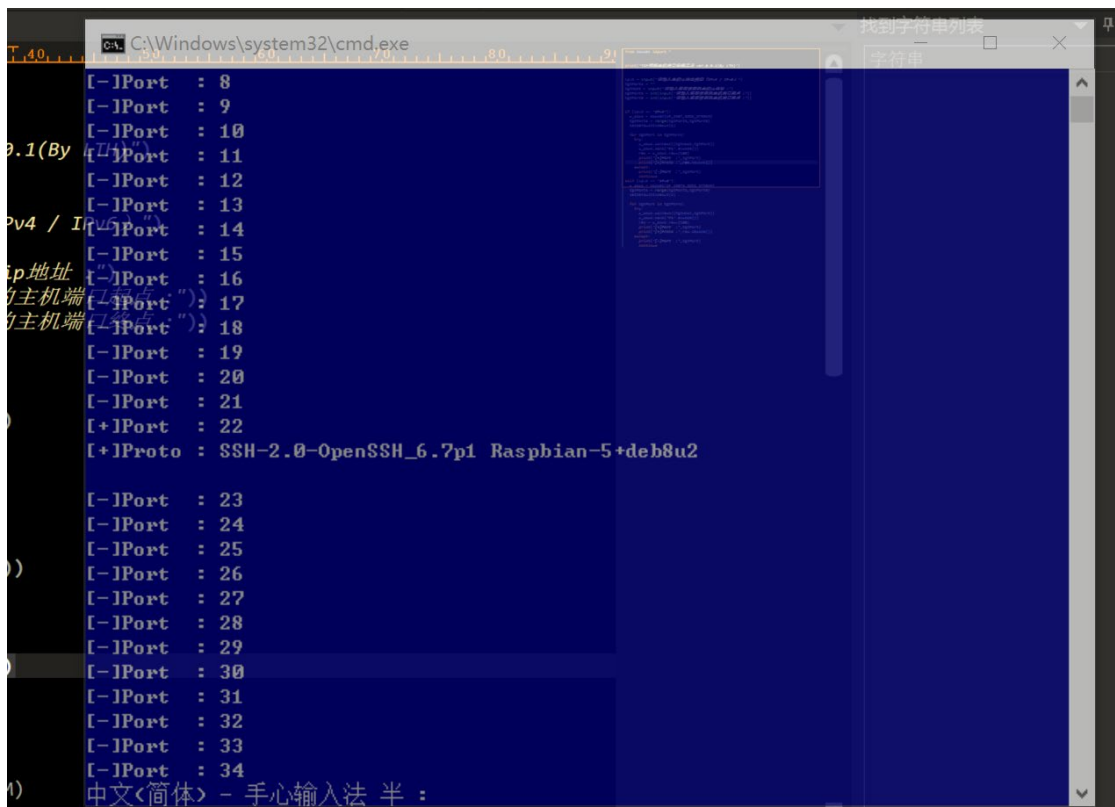


```
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>python tcpclient.py
TCP网络主机端口扫描工具 ver.0.0.1(By LTH)
请输入主机ip地址类型 (IPv4 / IPv6) IPv4
请输入您想侦测的主机ip地址 :192.168.1.106
请输入您想侦测的主机端口起点 :1
请输入您想侦测的主机端口终点 :6000
[-]Port : 1
[-]Port : 2
[-]Port : 3
[-]Port : 4
```

- 部分扫描结果（其中[+]表示端口开放，若检测到开放，则打印出服务与协议的相关信息，便于寻找漏洞，以便进一步入侵）



```
[-]Port : 8
[-]Port : 9
[-]Port : 10
[-]Port : 11
[-]Port : 12
[-]Port : 13
[-]Port : 14
[-]Port : 15
[-]Port : 16
[-]Port : 17
[-]Port : 18
[-]Port : 19
[-]Port : 20
[-]Port : 21
[+]Port : 22
[+]Proto : SSH-2.0-OpenSSH_6.7p1 Raspbian-5+deb8u2
[-]Port : 23
[-]Port : 24
[-]Port : 25
[-]Port : 26
[-]Port : 27
[-]Port : 28
[-]Port : 29
[-]Port : 30
[-]Port : 31
[-]Port : 32
[-]Port : 33
[-]Port : 34
```


五、实验心得和思考

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

本次实验只是进行了一些基础的信息收集与漏洞分析，涉及到域名信息收集、服务器信息收集、端口信息收集。