漏洞评估是所有渗透测试环节中非常重要的一部分，也是识别和评估目标系统漏洞的一个过程。本篇文章中，我们将从 Metasploitable 2 虚拟机的网络侧，对可获取的漏洞进行评估。在之后的教程中，我们还将对 Metasploitable 2 的 web 应用进行评估。

在上篇关于 Metasploitable 2 的指纹获取和信息枚举的教程中我们可以得知 Metasploitable 2 存在许多的漏洞。基于这些我们以收集的信息我们可以找出目标系统线上或是线下的漏洞。关于这些漏洞的具体利用方法，我们将会在下篇文章中讲解。这篇文章主要重点在对这些漏洞的分析上。我们将手动寻找漏洞，结合一些工具例如：Nmap 的一些脚本，同时我们还将结合自动化扫描工具例如： Open-Vas 。每一种扫描技术，都有它的优点和缺点，在下面的内容中你将体会到这种差异性！

如上所诉，有许多方式可以进行脆弱性分析。从手动检索漏洞数据库，在到全自动化的漏洞扫描工具例如： Open-Vas

及 Nessus等。而自动漏扫工具是一种非常主动的扫描方式，它通常会向目标系统发送大量的请求流量，从而获取扫描结果。正因为这种特性的存在，因此当我们对某个目标进行扫描时，一定要小心再小心！否则可能会给目标系统造成 DDos 甚至奔溃！ 当我们使用自动漏扫工具时，一定要适当的结合一些手动的方式，因为谁也不敢保证自动化工具的准确性。

Metasploitable 2 信息枚举

现在，让我们从之前收集的关于 Metasploitable 2 的信息，开始我们的漏洞评估工作。

```
1.其运行的操作系统版本为 Linux 2.6.9 - 2.6.33
2.服务器名称为 METASPLOITABLE
3.可获取 35 个账户信息
4.msfadmin 为管理员账户
5.msfadmin 管理员账户密码没有过期时间
6.我们知道哪些服务在跑，及这些服务的版本及侦听端口
7.有一个 webserver 和 SQL server 在 Metasploitable 下运行
```

从 Nmap 的扫描中，我们得到以下端口及服务信息：

Vsftpd 2.3.4 21 Open

OpenSSH 4.7p1 Debian 8ubuntu 1 (protocol 2.0) 22

Open

Linux telnetd service 23 Open

Postfix smtpd 25 Open

ISC BIND 9.4.2 53 Open

Apache httpd 2.2.8 Ubuntu DAV/2 80 Open

A RPCbind service 111 Open

Samba smbd 3.X 139 & 445 Open

3 r services 512, 513 & 514 Open

GNU Classpath grmiregistry 1099 Open

Metasploitable root shell 1524 Open

A NFS service 2048 Open

ProFTPD 1.3.1 2121 Open

MySQL 5.0.51a-3ubuntu5 3306 Open

PostgreSQL DB 8.3.0 - 8.3.7 5432 Open

VNC protocol v1.3 5900 Open

X11 service 6000 Open

Unreal ircd 6667 Open

Apache Jserv protocol 1.3 8009 Open

Apache Tomcat/Coyote JSP engine 1.1 8180 Open

从以上信息我们可以看出，这些服务包含许多可被利用的已知漏洞存在。下一步我们要找出哪些服务是有漏洞的，并手机其相关的利用方法。我们可以从以下提供的服务上知道哪些存在漏洞。例如现在最流行的 exploit-db 及开源漏洞库(OSVDB) 。我们还可以通过 kali 上的 searchsploit 来查找和评估已知漏洞及漏洞的利用代码。

这篇文章主要是要教大家学会怎么评估漏洞，而不是 Metasploitable 2 使用指南。因此我们只会评估其中一些服务漏洞。剩下的漏洞评估，大家可以自己去实践练习。下面让我们继续我们本篇文章的评估内容，同时我们来启动之前我们枚举发现的 Vsftpd 2.3.4 服务。

VSFTPD v2.3.4 漏洞

当我们在谷歌上搜索关于 Vsftpd 2.3.4 的已知漏洞后我们可以得知该版本有一个已知后门漏洞。

这意味着 Vsftpd 2.3.4 及之前的版本都存在这个 backdoor 漏洞。因此这是值得一试的在 Metasploitable 2 上。在 Metasploit 中，也有关于这个后门漏洞的利用模块。

CVE：CVE-2011-02523

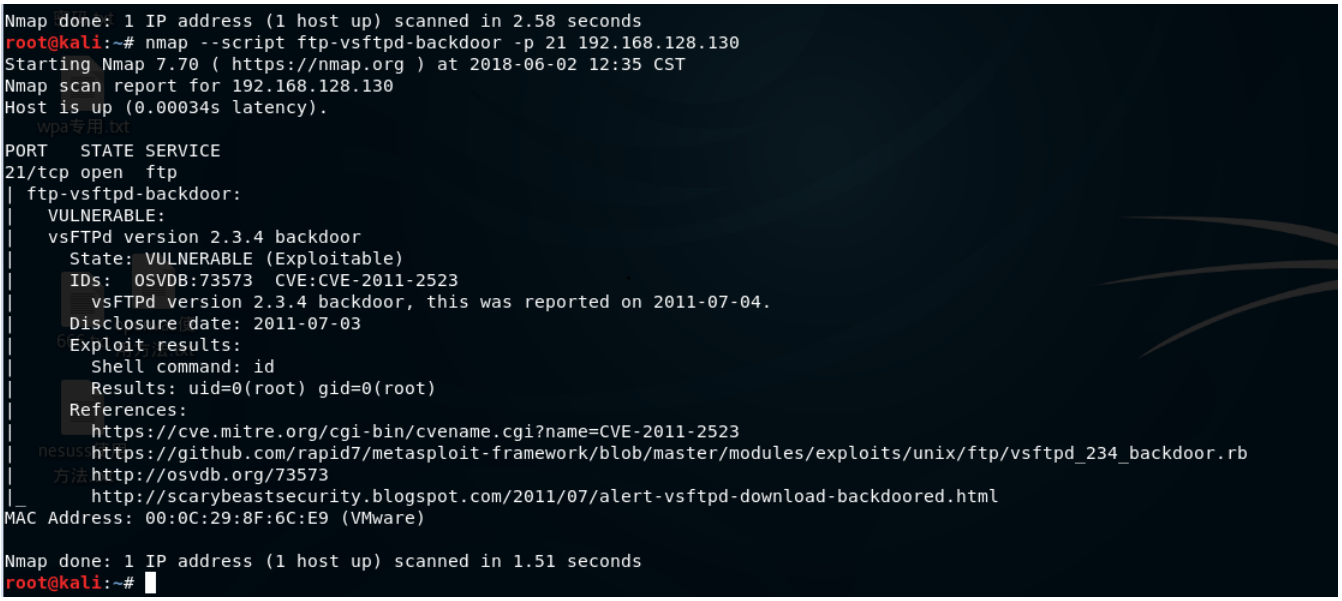
OSVDB：73573

Nmap 脚本扫描 VSFTPD v2.3.4

我们可以启动 Metasploit 来得知，目标系统上运行的服务是否是存在漏洞的。还有另一种方法就是通过 Nmap 的 ftp-vsftpd-backdoor 脚本来进行扫描探测。下面我们来启动 Nmap 并用以下命令，对目标主机执行扫描：

nmap -script ftp-vsftpd-backdoor -p 21 [目标主机]

从检测结果我们可以得知，目标使用的 vsFTPd 服务，为存在后门漏洞的版本。



Unreal ircd 漏洞

Metasploitable 运行着 UnrealIRCD IRC 后台守护程序，在某个版本上存在一个自动向其他正在监听状态中的端口发送一个紧跟“AB”信件的系统命令的漏洞。下面，我们看看目标主机上的 Unreal ircd 服务。从 Nmap 的扫描结果看，

它运行在 6667 端口上。但是我们还不知道它的具体版本号。这里我们可以使用一种通常的 banner 抓取技术。我们可以使用 Netcat 来抓取。命令如下：

Unreal ircd 漏洞

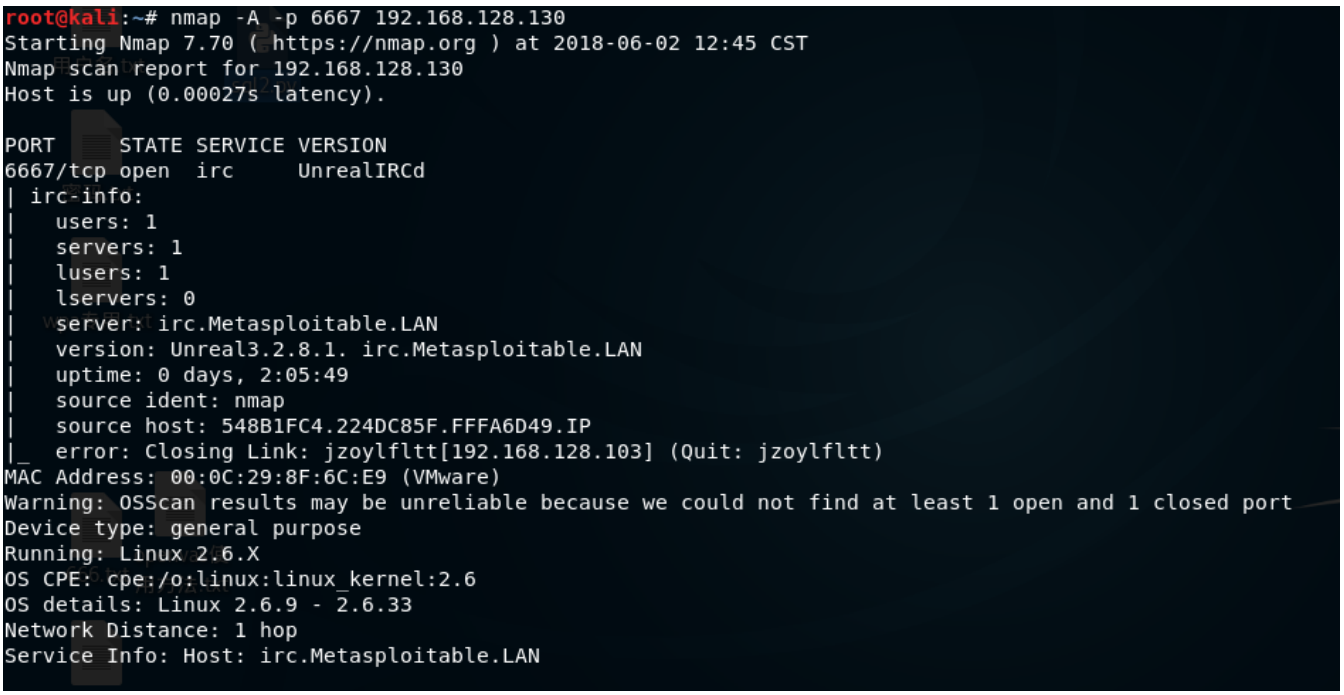
Metasploitable 运行着 UnrealRCD IRC 后台守护程序，在某个版本上存在一个自动向其他正在监听状态中的端口发送一个紧跟“AB”信件的系统命令的漏洞。下面，我们看看目标主机上的 Unreal ircd 服务。从 Nmap 的扫描结果看，它运行在 6667 端口上。但是我们还不知道它的具体版本号。这里我们可以使用一种通常的 banner 抓取技术。我们可以使用 Netcat 来抓取。命令如下：

nc [目标主机]6667

不幸的是 banner 信息并没有如愿返回给我们：

```
root@kali:~# nc 192.168.128.130 6667
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
whoani
:irc.Metasploitable.LAN 451 whoani :You have not registered
id
:irc.Metasploitable.LAN 451 id :You have not registered
ERROR :Closing Link: [192.168.128.103] (Ping timeout)
root@kali:~# nc 192.168.128.130 6667
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
```

那么让我们回到 Nmap 使用 Nmap 对 6667 端口进行一次完整的扫描，命令如下：

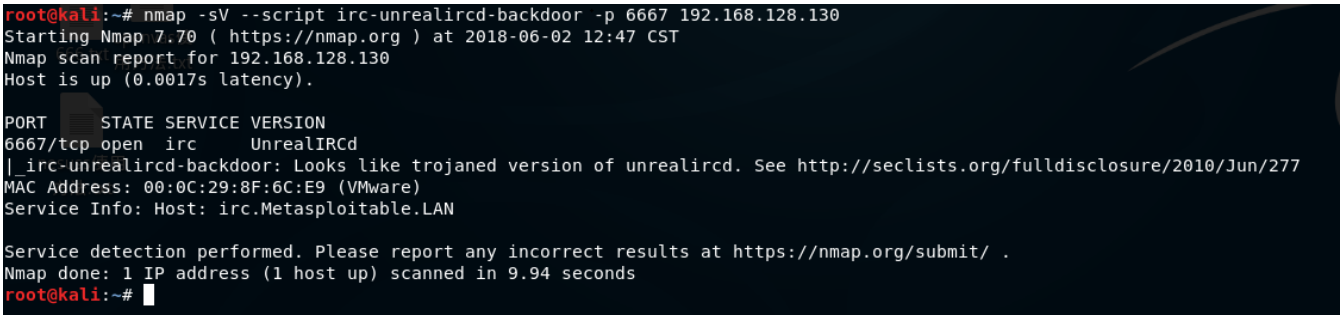


很幸运， Nmap 如我们所愿返回了 unreal ircd 服务相关的版本号，其版本为 unreal ircd 3.2.8.1 。当我们在谷歌上对该版本的 unreal ircd 进行查询后，我们可以得知，该版本存在一个后门漏洞。

在 Nmap 中同样有个脚本，可以用来检测该服务是否存在漏洞。命令如下：

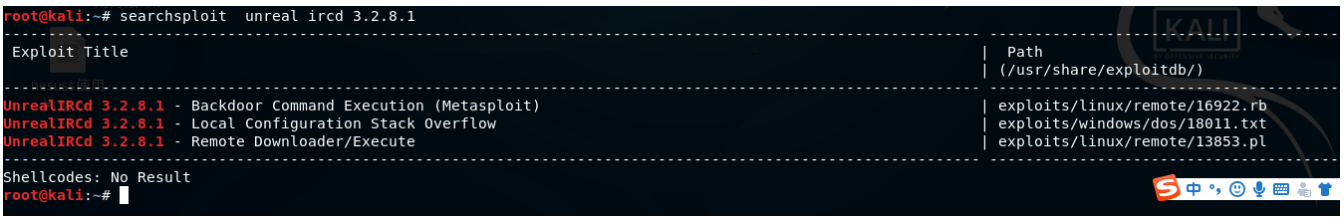
nmap -sV --script irc-unrealircd-backdoor -p 6667 [目标主机]

我们可以从输出结果得知，该版本是否为漏洞版本。由于脚本执行的命令无法返回到我们终端，因此我们无法百分百确定其是否存在漏洞。



使用 Exploit-db 评估漏洞

Exploit-db 是一个非常好的，查找已知漏洞的地方。它为我们提供了大量的漏洞利用细节，详细说明文档，shellcodes 等重要信息资源。我们可以使用关键字 CVE 或 OSVDB 来进行相关的查找工作。当我们搜索关于 Unreal IRCD 的漏洞时我们得到以下返回结果：



第一行的那个漏洞是针对 windows 平台下的，因此在 Metasploitable 2 Linux 上无法利用。当我们点击它们，我们就可以下载到它的漏洞利用代码。同时 Exploit-db 还为我们提供了，该漏洞版本软件的下载，以便于我们实验环境下的学习测试使用。

一般这些利用代码，都是由 Ruby (Metasploit 模块)，C，Perl 或者 Python 这些编程语言所编写。这里需要说明的是，我们使用这些 shellcode 常常不都不自己对其做一些代码或参数上的修改，只有修改成符合我们当前环境的代码，才有可能成功利用到。因此这就要求我们使用者有一定得编程能力和代码的阅读能力。许多安全研究员，为了避免一些脚本小子的恶意使用，往往只提供漏洞的 POC 概念验证代码。

小心，我们下载的 exploits ！

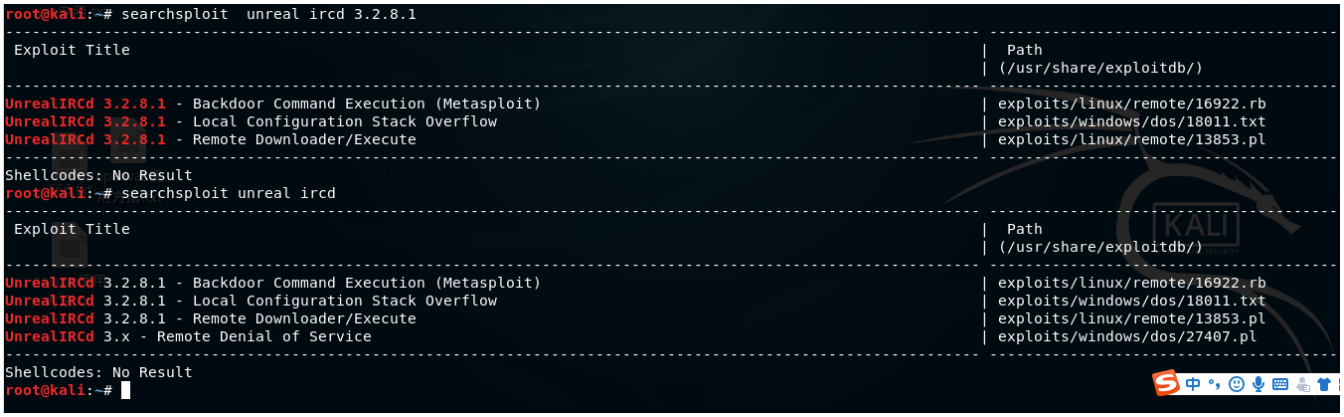
我们一定要小心从 Exploit-db 以外的地方下载的 exploits ！你可能会下载到带壳的恶意编码的后门程序，

并对你的计算机系统造成隐私及完整性的损害。为了避免这种情况发生，我们不得不对所下载的代码进行一次审计。不久前，我遇到一个 Apache 漏洞，被宣传为零日 exploit ，而且还是最近版本的没有打过补丁的 Apache 。经过对代码的分析，我得知那个 exploit 只是检查当前帐户权限，和格式化整个硬盘驱动器的！

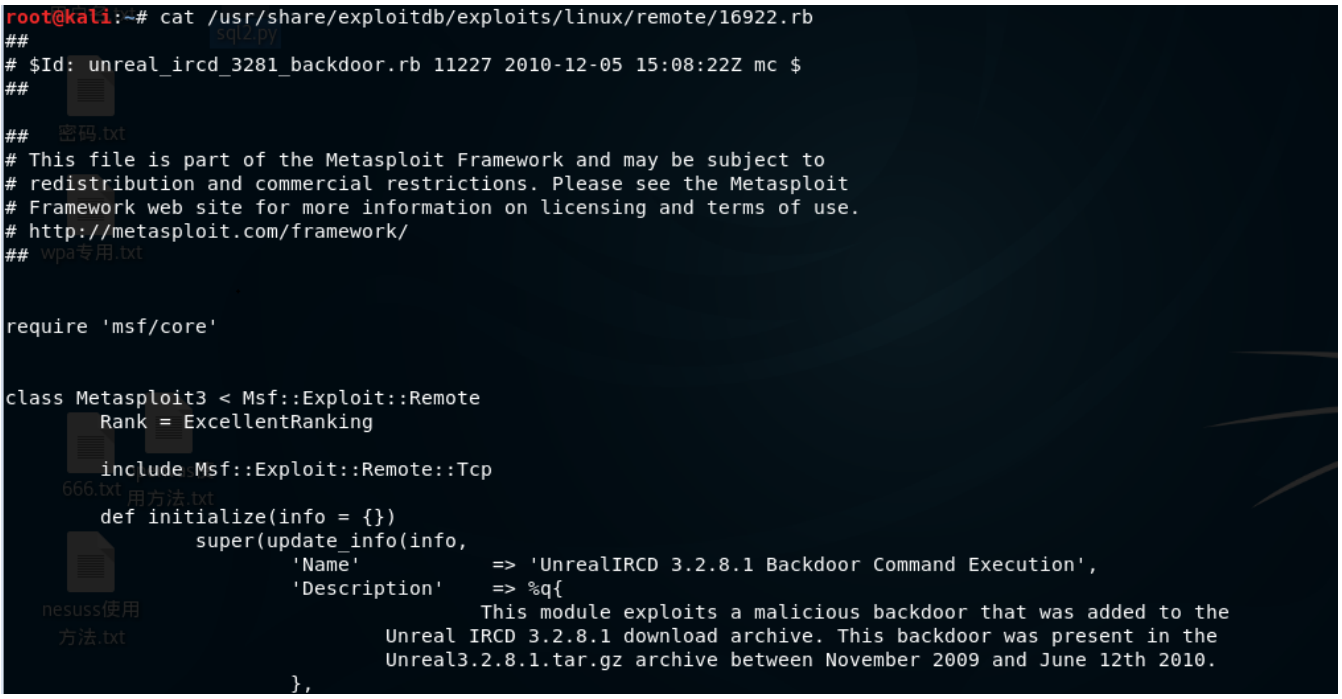
Kali Linux 下的 Searchsploit

在 kali 上其实也为我们默认集成了，一款专用于漏洞信息查询的工具 searchsploit 。使用命令如下：

searchsploit unreal ircd



我们可以使用 cat 命令，来查看其内容：



```
root@kali:~# cat /usr/share/exploitdb/exploits/linux/remote/16922.rb
##
# $Id: unreal_ircd_3281_backdoor.rb 11227 2010-12-05 15:08:22Z mc $
##

##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
# http://metasploit.com/framework/
##


require 'msf/core'


class Metasploit3 < Msf::Exploit::Remote
        Rank = ExcellentRanking

        include Msf::Exploit::Remote::Tcp
        def initialize(info = {})
                super(update_info(info,
                        'Name'           => 'UnrealIRCD 3.2.8.1 Backdoor Command Execution',
                        'Description'    => %q{
                                        This module exploits a malicious backdoor that was added to the
                                Unreal IRCD 3.2.8.1 download archive. This backdoor was present in the
                                Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th 2010.
                        },
```

```
                'References'     =>
                        [
                                [ 'CVE', '2010-2075' ],
                                [ 'OSVDB', '65445' ],
                                [ 'URL', 'http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt' ]
                        ],
                'Platform'       => ['unix'],
                'Arch'           => ARCH_CMD,
                'Privileged'     => false,
                'Payload'        =>
                        {
                                'Space'       => 1024,
                                'DisableNops' => true,
                                'Compat'      =>
                                        {
                                                'PayloadType' => 'cmd',
                                                'RequiredCmd' => 'generic perl ruby bash telnet',
                                        }
                        },
                'Targets'        =>
                        [
                                [ 'Automatic Target', { }]
                        ],
                'DefaultTarget' => 0,
                'DisclosureDate' => 'Jun 12 2010'))

        register_options(
                [
                        Opt::RPORT(6667)
```

Open-Vas 漏洞扫描器

到目前为止，我只介绍了关于使用 Nmap 及手动进行漏洞评估及查询的方法。其实除此之外我们还可以使用高度自动化

的扫描器，来进行漏洞的评估发现。例如： Open-Vas 和 Nessus 。这里需要注意的是，使用这类扫描器会产生大量的请求流量，使用不当将会对目标主机产生极大的影响。同时，随意使用扫描器去扫描那些未经授权的主机，是属于违法的行为。

我们启动 Open-Vas 漏洞扫描，完成一次完整的扫描，我们需要花点时间等待。之后扫描结果如下：

从扫描结果我们可以看到，已经发现许多严重的漏洞。最好我们结合使用多个扫描器进行扫描，这样就能在最大程度上，避免自动化扫描器带来的误报及漏报情况！

到此为止，我们就对 Metasploitable 2 做了一次相对全面的漏洞评估。接下来要做的就是如何利用这些漏洞了！

参考文章：

http://www.secist.com/archives/1994.html