
SeedUbuntu 虚拟机使用手册

目录

1	Seed Ubuntu 实验环境搭建.....	2
1.1	实验环境	2
1.2	虚拟机软件: vmware.....	2
1.3	Ubuntu 用户账号	2
1.4	docker 安装及镜像创建(可跳过).....	2
1.4.1	给 SEEDUbuntu 扩容	2
1.4.2	安装 docker	3
1.4.3	安装 docker-compose.....	5
1.4.4	打包 SEEDUbuntu 镜像	5
1.5	docker 容器使用	6
1.5.1	使用镜像创建容器	6
1.5.2	容器操作	7
1.5.3	telnet 服务启动	7
1.6	可能出现的问题	7
2	应用软件.....	9
3	服务器软件.....	11
3.1	Apache HTTP Server	11
3.2	MySQL Server	11
3.3	Bind9 DNS Server	12
3.4	Other Servers	12
4	其他.....	13
4.1	VM Customization Folder	13
4.2	Package List	13
4.3	Software Security Lab Tools.....	13
	参考.....	14

1 Seed Ubuntu 实验环境搭建

1.1 实验环境

提供的 SEEDUbuntu16.04 虚拟机(已安装 docker)。

1.2 虚拟机软件：vmware

将虚拟机文件解压缩，用 vmware 打开虚拟机 Ubuntu-seed 目录下的
Ubuntu-seed.vmx

(该虚拟机用 vmware 15.5.0 版本创建的，如果出现不兼容的情况，可以使用此版本)

1.3 Ubuntu 用户账号

虚拟机里创建了两个帐户。用户名和密码如下：

1.用户名：root，密码：seedubuntu。注意：Ubuntu 不允许 root 从登录窗口直接登录。您必须以普通用户身份登录，然后使用命令 su 登录 root 帐户。

2.用户 ID：seed，密码：dees。此帐户已获得 root 权限，但要使用 root 权限，需要用 sudo 命令。

1.4 docker 安装及镜像创建(可跳过)

此步骤针对原来就已经下载了 seed 虚拟机，但是虚拟机里面没有 docker 的同学。如果下载了本次实验提供的 seed 虚拟机，可以跳过此小节。

1.4.1 给 SEEDUbuntu 扩容

首先在虚拟机设置里面，选择硬盘，然后扩容。(使用 vmware-diskmanager 命令行工具也可)

打开虚拟机，apt-get install gparted

运行 gparted，将除了/dev/sda1 外的分区格式化，再进行扩展，建议

80G（40G 以上），这是因为打包生成的镜像导入创建的容器时，一个容器占用 8G 左右的空间。分好的分区如图 1 所示。

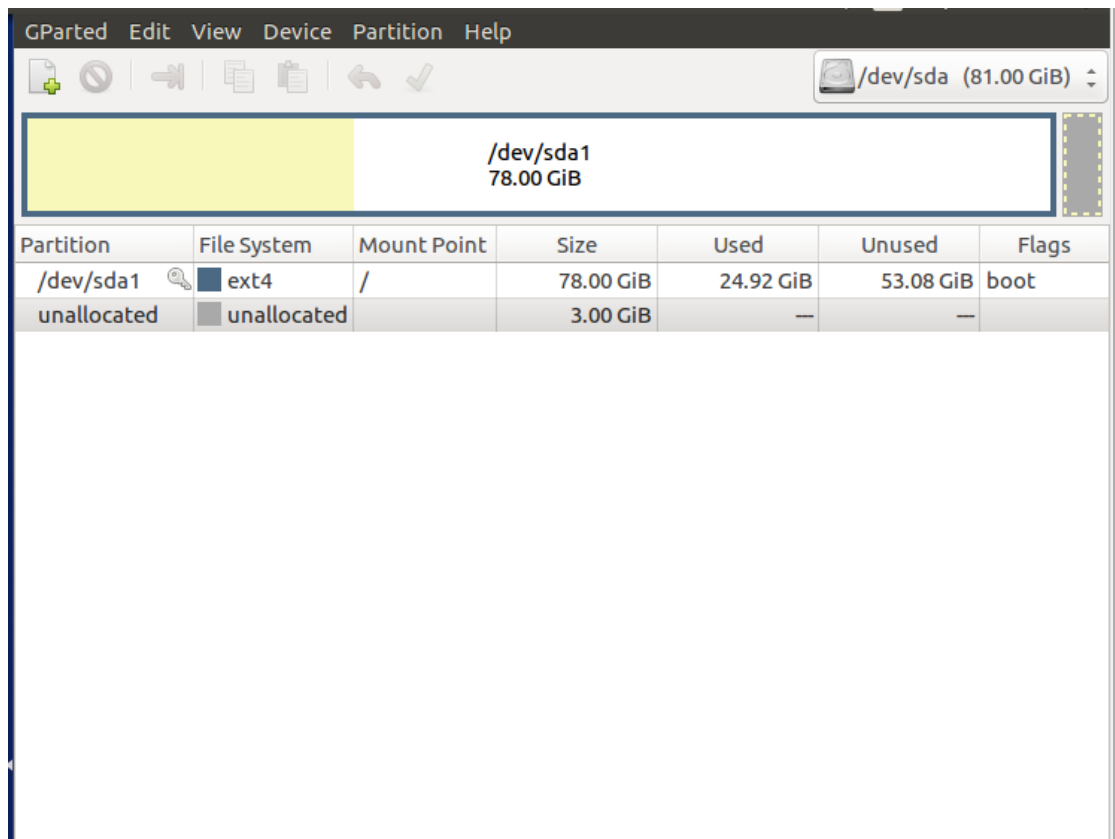


图 1 扩容后分区

1.4.2 安装 docker

由于 seedubuntu1604 是 i386 架构的 32 位机器，不能安装 docker-ce-cli、docker-ce，解决方法是安装 docker.io，这区别在于 docker.io 是 ubuntu 团队进行维护的版本，版本较低，但不影响实验。

如果在之前有安装旧版本的 docker 的话，需要卸载之后才能安装：

```
$ sudo apt-get remove docker docker-engine docker.io
```

更新

```
sudo apt-get update
```

下载相关工具

添加
docker
GPG 密

```
sudo apt-get install \  
apt-transport-https \  
ca-certificates \  
software-properties-common
```

官方
钥

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
apt-key add -
```

设立仓库

```
$ sudo add-apt-repository \  
"deb [arch=amd64]  
https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) \  
stable"
```

安装 docker

```
$ sudo apt-get update  
$ sudo apt-get install docker.io
```

运行 `docker version` 可检验是否完成安装，如图 2 所示。

```
root@VM:/home/seed# docker version
Client:
Version:      19.03.6
API version:  1.40
Go version:   go1.12.17
Git commit:   369ce74a3c
Built:        Fri Feb 28 23:46:10 2020
OS/Arch:      linux/386
Experimental: false

Server:
Engine:
Version:      19.03.6
API version:  1.40 (minimum version 1.12)
Go version:   go1.12.17
Git commit:   369ce74a3c
Built:        Wed Feb 19 01:06:16 2020
OS/Arch:      linux/386
Experimental: false
containerd:
Version:      1.3.3-0ubuntu1~18.04.2
GitCommit:
```

图 2 docker version

1.4.3 安装 docker-compose

```
$sudo apt-get install docker-compose
```

安装成功后运行 docker-compose version，如图 3 所示。

```
root@VM:/home/seed# docker-compose version
docker-compose version 1.17.1, build unknown
docker-py version: 2.5.1
CPython version: 2.7.12
OpenSSL version: OpenSSL 1.0.2g 1 Mar 2016
```

图 3 docker-compose version

1.4.4 打包 SEEDUbuntu 镜像

通过 tar 备份目录

```
#tar -cvpf /home/buildrpm.tar --directory=/ --exclude=proc --exclude=sys --
exclude=dev --exclude=run /
```

导入镜像：

```
#cat buildrpm.tar | docker import - seedubuntu
```

1.5 docker 容器使用

虚拟机里面已经安装了 docker，seed 虚拟机压缩文件在
/home/contain/buildrpm.tar

镜像也已经导入，镜像名字为 seedubuntu

因为 docker 容器里面的系统没有图形界面，需要图形界面的主机可以直接用虚拟机，而不用容器。比如攻击机需要运行 wireshark 监听报文的话，攻击机可以直接用虚拟机，user 和 server 机器可以用 docker 容器。

系统里面已经创建了一个名字为“server”的镜像，该镜像中已经安装了 telnet 服务。

1.5.1 使用镜像创建容器

如果需要创建一个新的名字为“user”的镜像，则可以使用以下命令：

```
# docker run -it --name=user --privileged "seedubuntu" /bin/bash
```

常用可选参数说明：

-i 表示以“交互模式”运行容器

-t 表示容器启动后会进入其命令行。加入这两个参数后，容器创建就能登录进去。即 分配一个伪终端。

--name 为创建的容器命名

-v 表示目录映射关系(前者是宿主机目录，后者是映射到宿主机上的目录，即 宿主机目录:容器中目录)，可以使用多个-v 做多个目录或文件映射。
注意:最好做目录映射，在宿主机上做修改，然后 共享到容器上。

-d 在 run 后面加上-d 参数,则会创建一个守护式容器在后台运行(这样创建容器后不会自动登录容器，如果只加-i -t 两个参数，创建后就会自动进去容器)。

-p 表示端口映射，前者是宿主机端口，后者是容器内的映射端口。可以使用多个-p 做多个端口映射

-e 为容器设置环境变量

--network=host 表示将主机的网络环境映射到容器中，容器的网络与主机相

同

-it 创建一个伪终端交互界面，name 指定容器名称，否则随机一个名字

--privileged 为了后续实验能够修改系统变量，需要添加上这个参数

docker [container] ps -a 可以查看所有容器 (在新的 docker 版本中需要带 container 参数，因为此虚拟机的 docker 版本比较老，不需要此参数，其它命令类似)

1.5.2 容器操作

docker ps 可以查看正在运行的容器

docker exec -it <容器名/id> <运行后第一个命令>

#一般使用/bin/bash，否则无法进入交互界面

比如：进入容器 user 的命令：

```
# docker exec -it user /bin/bash
```

停止一个已经在运行的容器

sudo docker stop 容器名或容器 id

启动一个已经停止的容器

sudo docker start 容器名或容器 id

#删除容器

sudo docker rm 容器名或容器 id

1.5.3 telnet 服务启动

重启 openbsd-inetd

```
$ sudo /etc/init.d/openbsd-inetd restart
```

查看 telnet 运行状态

```
$ sudo netstat -a | grep telnet
```

1.6 可能出现的问题

1. 新装的 ubuntu 虚拟机，执行 sudo apt update,总会报(appstreamcli:21755):

GLib-CRITICAL **: g_strchomp: assertion 'string != NULL' failed

解决:

```
$ sudo apt-get install libappstream4
```

2. find: '/run/user/1000/gvfs': 权限不够 的解决办法

解决: 其实这个目录是空的, 查不查都没关系。所以, 以下解决方式比较简粗暴:

```
umount /run/user/1000/gvfs // 卸载该文件
```

```
rm -rf /run/user/1000/gvfs // 删除该文件
```

3. 回显功能没启动

在 telnet 中使用回显功能, 首先需要启动 echo 服务。默认情况下 ubuntu 是不开启 telnet 的回显功能的。

开启方法如下:

```
#apt-get install xinetd
```

之后, 需要配置在 xinetd 中, 启动 echo 服务,

```
$vi /etc/xinetd.d/echo
```

把 disable 字段的值从 yes 改成 no

接着, 启动 xinetd 服务

```
$sudo service xinetd start
```


2 应用软件

第 4.2 节提供了安装的完整软件包列表。在这里，我们只强调一些最重要的，SEED 实验常用的工具。为了便于操作，已经将它们固定在启动器上（见图 4）。



图 4：启动器中的应用程序快捷方式

Terminator。这是一个终端应用程序，提供了一种管理多个终端窗口的便捷方式。使用 VM 时，应记住两个重要功能：分屏和配置文件。通过右键单击窗口并选择水平分割或垂直分割，可以分割终端的屏幕。此外，我们在 Terminator 中设置了三种颜色/字体配置文件，可以通过右键单击窗口并从配置文件菜单中选择。图 5 显示了分屏和配置文件的使用。

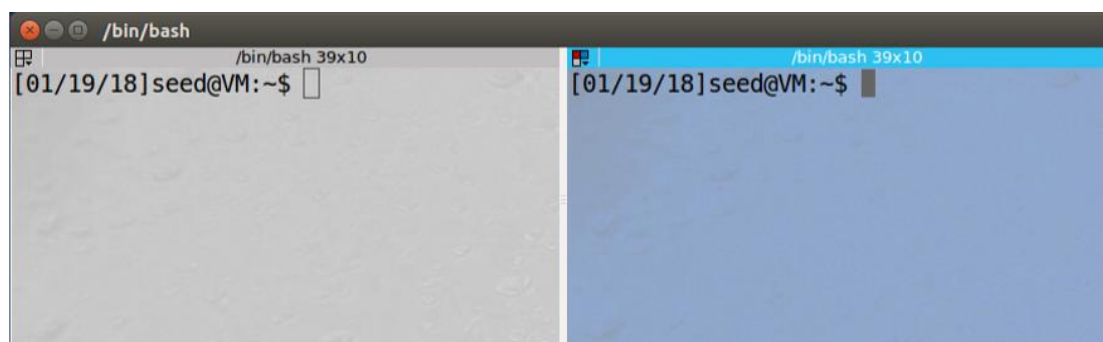


图 5：具有分屏和不同配置文件的 Terminator

Text Editors. 我们在 VM 中提供了两个文本编辑器，即 gedit 和 sublime。gedit 是 Ubuntu OS 附带的默认文本编辑器。与 gedit 相比，sublime 提供了一些额外的功能。这些工具的比较可以在[4]中找到。

Firefox Extensions. Firefox (version 60) 已安装在 VM 中。我们已经安装了 HTTP Header Live extension [5]以检查 Web 安全实验中的 HTTP 数据包。此扩展可以通过浏览器右上角的侧边栏图标访问。我们还安装了一个时间戳扩展名，可以通过右上角的时钟图标访问。

Networking. 我们已经安装了三个工具来协助网络安全实验（所有工具都安装在/usr/bin/中）：

1. **Netwox:** 这是一个网络工具箱，可用于生成不同类型的数据包。它包含 222 个网络功能。netwag 是 netwox 的图形前端。运行 netwox /netwag 需要 root 权限。

2. **Wireshark:** 这个工具是一种流行的网络协议分析器。它在检查网络数据包时很有用。

3. **Scapy:** 这个工具是一个交互式数据包操作程序。

GDB-PEDA. 使用 gdb 调试程序时，此工具[1]提供了更多信息。它在使用 gdb 时自动运行。

Mobile Security Lab Software. 为了支持移动安全实验，我们在文件夹 /home/seed/android 下设置了 Android SDK 和 NDK。为了允许 Android 应用程序的逆向工程，我们安装了 apktool。还安装了 Oracle Java 8。

3 服务器软件

本节提到的所有服务都是由 VM 自动开启的。这可以通过在终端运行：
`service --status-all`，进行验证。

3.1 Apache HTTP Server

Apache2 是一个开源 HTTP 服务器。它用于托管 Web 安全 SEED 实验的所有网站。我们使用 `virtual host` 功能，从而允许我们在同一台机器上运行多个网站。SEED VM 中的所有网站都使用端口 80。`virtual host` 配置可以在以下文件中找到：`/etc/apache2/sites-enabled/000-default.conf`。以下代码段显示了 `VirtualHost` 的示例。

```
<VirtualHost *:80>
    ServerName http://www.xsslabelgg.com
    DocumentRoot /var/www/XSS/Elgg/
</VirtualHost>
```

上面的代码段是用于 XSS 实验的 Elgg 网站的 `VirtualHost` 条目。

`DocumentRoot` 指网站源代码所在的目录。与上述类似，我们在配置文件中有以下网站的条目：

<code>www.xsslabelgg.com</code>	<code>/var/www/XSS/Elgg</code>
<code>www.csrflabelgg.com</code>	<code>/var/www/CSRF/Elgg</code>
<code>www.csrfattacklab.com</code>	<code>/var/www/CSRF/Attacker</code>
<code>www.seedlabsqlinjection.com</code>	<code>/var/www/SQLInjection</code>
<code>www.repackagingattacklab.com</code>	<code>/var/www/RepackagingAttack</code>

我们还配置了 `/etc/hosts` 文件，以将虚拟机的本地 IP 地址与网站主机名相关联。下面的代码段显示了 `/etc/hosts` 条目：

```
127.0.0.1 www.xsslabelgg.com
127.0.0.1 www.csrflabelgg.com
127.0.0.1 www.csrfattacklab.com
127.0.0.1 www.seedlabsqlinjection.com
127.0.0.1 www.repackagingattacklab.com
```

3.2 MySQL Server

MySQL 是一个开源数据库管理软件。它在 VM 中用于管理与已安装网站相对应的数据库。我们在 `mysql` 服务器中有以下数据库：

- 1.用于 SQL 注入站点的用户数据库

2.用于 XSS 站点的 elgg_xss 数据库

3. 用于 CSRF 站点的 elgg_csrf 数据库

您可以通过客户端应用程序/usr/bin/mysql，访问 MySQL 数据库服务器。

以下是关于如何使用 mysql 的简单演示。

```
$ mysql -u root -pseedubuntu
mysql> show databases;
mysql> use Users;
mysql> show tables;
mysql> select * from credential;
mysql> quit
```

MySQL Accounts。目前，在 MySQL 服务器中有两个账户。 用户名和密码如下所示。

1.用户： root， 密码： seedubuntu

2.用户： elgg_admin， 密码： seedubuntu（Web 应用程序使用此帐户连接到 mysql 服务器）

phpMyAdmin。 我们还安装了 phpMyAdmin，这是一个允许通过浏览器管理 MySQL 的 PHP 工具。可以通过导航到 <http://localhost/phpmyadmin> 来访问它。phpmyadmin 的帐户,用户名为 root，密码为： seedubuntu。

3.3 Bind9 DNS Server

Bind9 是域名系统组件的开源实现。 它主要用于 SEED DNS 网络安全实验室。 Bind9 的主要配置文件位于/etc/bind/named.conf.options 中。您还需要了解文件/var/cache/bind/dump.db，它是当前配置的 Dump 文件。

3.4 Other Servers

我们还安装了一个 ftp 服务器（vsftpd），一个 telnet 服务器（openbsd-inetd）和一个 ssh 服务器（ssh）。

4 其他

4.1 VM Customization Folder

在一些实验中，尤其是网络安全实验，我们必须运行多个 VM，并在它们之间来回切换。由于所有这些虚拟机看起来都一样，因此很难知道我们所在的虚拟机。我们提供了一个自定义文件夹来修改虚拟机的外观，使其更易于管理多个虚拟机。该文件夹可以在 `/home/seed/Customization` 中找到。由于网络实验最多涉及三个 VM，因此自定义文件夹为三个角色提供图标和桌面背景，即用户，代理/服务器和攻击者。

4.2 Package List

除了 Ubuntu16.04 安装附带的软件包外，还使用“`apt-get install`”命令安装了以下附加软件包。

```
terminator, curl, sublime-text, bless, ghex, vim,  
libssl-dev, openbsd-inetd, telnetd, openssh-server,  
vsftpd, bind9, libnet1-dev, apache2, mysql-server,  
php, libapache2-mod-php, php-mysqldb, wireshark,  
netwox, libpcap-dev, zsh, git, python-pip, capstone,  
squid, scrapy, oracle-java8-installer, adb
```

4.3 Software Security Lab Tools

我们安装了两个工具来帮助我们探索软件安全实验：

- Shellnoob 这个工具[3]帮助我们编写像缓冲区溢出实验的 shellcode。例如，它可以将汇编指令转换为 32 位和 64 位架构的 shellcode。它可以在 `/home/seed/source/shellnoob` 中找到。

- RoPGadget 这个工具[2]涉及面向回归的编程。它允许在二进制文件中搜索 ROP 小工具以促进 ROP 开发。它可以在 `/home/seed/source/ropgadgets` 中找到。

参考

- [1] Gdbpeda Github. <https://github.com/longld/peda>, 2017.
- [2] RopGadget Github. <https://github.com/JonathanSalwan/ROPgadget>, 2017.
- [3] Shellnoob Github. <https://github.com/reyammer/shellnoob>, 2017.
- [4] Comparison of gedit and sublime. https://web.archive.org/save/https://www.slant.co/versus/40/58/~sublime-text_vs_gedit, 2018.