ctf-pwn环境配置

说明

因为没有一个很好的环境录制人声，所以就没有人声了。

文档在博客上 https://gnol3.top/

BGM：https://c.y.qq.com/base/fcgi-bin/u?__=G6bsxblT4hNA

俺不是嘉心糖，单纯觉得这个主题css样式很好看而已。

开场白

本次安装环境选用的是kubuntu20.04，大家可以按照自习喜好来选择自己的ubuntu版本。推荐20.04，18.04，当然22.04也不是不行，就是可能会有些小问题。

大家如果没有Linux使用基础也不要紧，本次录制会把命令都解释解释

文章里面写到可以使用deepin，实际上deepin还是没有ubuntu好用。

下面我们就正式开始吧。

源的介绍

源是什么？在Linux中有一个叫软件源的东西，软件源里面有软件包，我们可以使用ubuntu的apt包管理器下载软件。

安装一般使用如下命令。

```
apt install gdb
```

上述是一个安装gdb的命令，但是如果在普通用户下会报错。我们在虚拟机里看看会报怎么样的错吧。

报的错为：

```
 gnol@kubt20  ~  apt install gdb
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13:
Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend),
are you root?
```

可以看到，实际上报的是说我们权限不够，我们不是root用户。所以大家要注意在Linux下安装软件需要root权限。

怎么获取root权限呢，下面介绍两个方法：

1. 使用`sudo su`切换root用户(使用`whoami`命令可以查看当前用户是谁)
2. 使用`sudo apt install gdb`，即在安装命令前使用sudo，来说明我们要以root身份运行此命令

因为我已经安装了gdb所以限使是没有需要重新安装的内容。

ubuntu换源

因为ubuntu的官方源在国外，我们访问速度很慢，装软件很费劲，所以我们需要把apt源换为国内的源，比如阿里源或者清华源

博客中写道，已经将常用源上传到服务器上了，所以我们可以使用`wget`来下载记录着源的文本。

1. 首先我们切换root用户`sudo su`
2. 因为我们是20.04所以复制20.04的内容

```
## ubuntu20.04
cd /etc/apt # cd 命令是切换目录的命令，我们可以使用cd切换到任意目录下
# 在学习中我们会看到~这个波浪线，如果我们是普通用户gnol,那么~表示/home/gnol,如果我们是root用户那么~表示/root
# pwd命令可以查看当前路径
# ls命令可以查看当前路径下的文件与文件夹
cp sources.list sources.list.bak # cp命令就是copy，我们这把原来的源打一个备份，以防外一之后需要用。
# cat命令可以输出文件中的内容，其中在终端中复制粘贴是ctrl+shift+c/v。
wget http://gnol3.top/doc/ubuntu20-src.txt # 下载一下源
# 显示saving to ubuntu20-src.txt.1，这是因为我之前已经下过一个了
cat ubuntu18-src.txt > sources.list
# cat命令之前说可以输出文件内容，实际上默认是输出到我们的终端上的，但是我们可以指定输出地址
# 这里我们就指定输出到了sources.list里面 > 就是重定向的意思，且为擦除写，与C语言中的fopen函数中的r模式相似
# >> 是追加写，与fopen中的a相似
# 因为我换过源了所以可以看到两者文件中的内容是一样的


# 现在我把这个文件中的内容全部删除了
# 卧槽，发现文档中的最后一行都是错的，现场改一下博客内容
cat ubuntu20-src.txt > sources.list # 这样才是对的，版本一定不能搞错

#好了现在我们的源列表就换好了。然后我们需要更新一下源内容

apt update && apt full-upgrade
```

3. 更新源内容

```
apt update && apt full-upgrade
```

因为我没有什么需要更新的，所以很快就执行完了。大家可能需要更新个几分钟(看个人网速)。撕，似乎我也有要更新的内容。

听一下歌曲吧，等一等。在更新xserver什么的，其实可以不更新，我就中断更新过程了，使用`ctrl+c`中断。

4. 安装vmtools，让我们窗口能够自适应我们的屏幕（并且与物理机共享剪切板，如果没有vmtools我们就不能从物理机里面复制粘贴文件或者文本到虚拟机里面，别看我之前复制粘贴了一些命令，那是因为我的虚拟机已经配置完了）

```
apt install open-vm-tools-desktop
```

5. 此时可以重启一下虚拟机，使vm-tools生效，从实现虚拟机与物理机的共享剪切板(重启很重要的，重新加载一些服务)

```
reboot
```

因为我这台机子重启有点慢，就不重启了。大家记得重启一下

安装常用build工具

因为安装了vmtools，复制粘贴就可以了(用虚拟机的火狐浏览器打开gnol3.top也可以复制粘贴)

```
sudo apt install build-essential gcc-multilib g++-multilib
```

这台虚拟机好像正好没装这个。好了装完了。

安装git

复制粘贴

```
sudo apt install git
```

设置git时候不检查ssl，因为有时候你**开代理**，他会说你无ssl证书

```
git config --global https.sslverify "false" && git config --global
http.sslverify "false"
```

vim

选装，平常我使用的是vim，大家也可以用别的，这里就不做讲解了，大家可以自行去看theCW大佬的视频。

zsh

大家可以看到我的这个终端shell是zsh 5.8,而不是默认的bash

使用zsh并用oh-my-zsh，设置你喜欢的主题，你的终端会更漂亮

注意一点，这个更改默认shell文档里的命令

```
chsh -s /bin/zsh gnol # 将zsh设置为gnol(大家换成自己的)下的默认shell
# 这里应该写的很清楚了，把gnol换成自己的id，但是还是有同学复制粘贴也不管是不是自己的id ^…^
```

简单设置代理

如果你会用clash，可以简单设置代理

```
alias proxy="export http_proxy=http://127.0.0.1:7890;export
https_proxy=http://127.0.0.1:7890"
alias unproxy="unset http_proxy;unset https_proxy;"
```

python与pip

注意下，这个文档里下来python2在20.04下需要手动安装，这里就自学一下把，点击那个超链接，复制粘贴里面的命令就差不多了

python3-pip也是一样的，直接复制粘贴命令

```
sudo apt install python3-pip
```

pip换源

这个事情和上面的apt换源差不多，也是因为网速的问题，我们用国内的镜像源(一不小心暴露了我拼音垃圾的事实，sad)

pip换源，在家目录下和/root目录下都建一个.pip文件夹,创建一个pip.conf文件

翻译一下这句话

```
mkdir .pip # 使用mkdir出啊关键文件夹
vim pip.conf# 然后再创建pip.conf
```

文件内容就是复制粘贴解决

```
[global]
index-url =http://mirrors.aliyun.com/pypi/simple/
[install]
trusted-host = mirrors.aliyun.com
```

可以看到，在用户家目录下和root目录下都创建了这个文件

安装pwntools

```
pip install --upgrade pwntools # 选装
pip3 install --upgrade pwntools
```

复制粘贴上述两行。其中pip2的pwntools里面集成了checksec，如果我们选择手动安装checksec也可以不装pip2版本的pwntools

验证是否安装成功

```
 gnol@kubt20   ~/.pip   python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from pwn import *
>>>

 ✗ gnol@kubt20   ~/.pip   python2
Python 2.7.18 (default, Jul  1 2022, 12:27:04)
[GCC 9.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from pwn import *
>>>

```

安装checksec

```
# 直接装
sudo apt install checksec

 源码安装
git clone https://github.com/slimm609/checksec.sh
cd checksec.sh
sudo ln -s checksec /usr/local/bin/checksec

 如果已经存在checksec就只需要创建软连接即可,其实使用alias也是一样的
sudo ln -sf checksec /usr/local/bin/checksecv
```

这个使用方法会有一点点不一样，如果是pip2安装的就直接`checksec filename`如果是apt里面安装的就要`checksec --file=filename`好像是这样，主要是我常用pip2安装，所以大家可以使用`--help`来 查看帮助，看看具体怎么用这个命令。

ROPgadget

这个可以不用手动装，因为pwntools无论是pip2还是pip3，好像都自带了，会帮你一起装好，当然你想要最新版本的，可以直接clone源码，手动安装

```
sudo apt-get install python-capstone
git clone https://github.com/JonathanSalwan/ROPgadget.git
cd ROPgadget
sudo python setup.py install
```

查询gadget

```
ROPgadgete --bianry sort --only "pop|ret"
```

大家也可以用--help来看如何使用这个命令，他举了一些例子

```
examples:
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --ropchain
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --depth 3
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --string "main"
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --string "m..n"
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --opcode c9c3
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --only "mov|ret"
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --only
"mov|pop|xor|ret"
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --filter
"xchg|add|sub|cmov.*"
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --norop --nosys
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --range
0x08041000-0x08042000
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --string main --
range 0x080c9aaa-0x080c9aba
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --memstr
"/bin/sh"
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --console
  ROPgadget.py --binary ./test-suite-binaries/elf-Linux-x86 --badbytes
"00|01-1f|7f|42"
```

```
  ROPgadget.py --binary ./test-suite-binaries/Linux_lib64.so --offset
0xdeadbeef00000000
  ROPgadget.py --binary ./test-suite-binaries/elf-ARMv7-ls --depth 5
  ROPgadget.py --binary ./test-suite-binaries/elf-ARM64-bash --depth 5
  ROPgadget.py --binary ./test-suite-binaries/raw-x86.raw --rawArch=x86 --
rawMode=32
```

gdb动态调试

gdb是用来动态调试的，一般gdb都是自带的，如果没有自带，请用下述命令安装

```
sudo apt install gdb
```

使用`gdb filename`格式即可对`filename`进行调试

不过我的gdb花里胡哨的原因是装了`pwndbg`这个插件，看所写的文档， 可以知道`pwndbg`，`gef`，`peda`三个插件，我觉得都挺好使的。pwndbg属于是把两者优点全吸收了。

颜值这个东西就见仁见智了(没装peda，8好意思，展示不了了，可以自己去项目仓库里面看,peda颜值有点低的，不过现阶段ubuntu16还能装，pwndbg和gef已经抛弃ubuntu16了)

安装也很容易，基本上项目仓库里面都有安装命令。比如pwndbg安装使用

```
git clone https://github.com/pwndbg/pwndbg
cd pwndbg
./setup.sh
```

然后pwnGDB大家自行了解吧，配置gdb都在`~/.gdbinit`里面配置。

稍微了解一下也会配置了，source就是加载资源，其实就是在本进程中执行这个程序，大家可以去了解一下，这个source的作用，总之就是不会产生子进程。

安装one_gadget与seccomp-tools

这两者都需要用到ruby这个程序语言的包管理器`gem`

所以按照文档来安装ruby并安装他们两者

```
sudo apt install ruby ruby-dev
sudo gem install one_gadget
sudo gem install seccomp-tools
```

不过安装挺慢的，可以自行了解一下gem换源，俺妹深入了解，因为这两个包挺小的。

安装main_arena_offset

这个是用来查询main_arena到libc加载地址之间的偏移的，这个使用方法与one_gadget类似，他就是一个shell脚本很简单，高版本可能不能用，但是这个工具给我们提供了一个思路，大家可以尝试自己实现

```
 gnol@kubt20   ~/ctf/buu_libc   main_arena libc-2.27-64.so
[+]libc version : glibc 2.27
[+]build ID : BuildID[sha1]=b417c0ba7cc5cf06d1d1bed6652cedb9253c60d0
[+]main_arena_offset : 0x3ebc40
 gnol@kubt20   ~/ctf/buu_libc   main_arena libc-2.27-86.so
[+]libc version : glibc 2.27
[+]build ID : BuildID[sha1]=63b3d43ad45e1b0f601848c65b067f9e9b40528b
[+]main_arena_offset : 0x1d57a0
```

一共100行的脚本，可以学习学习

安装patchelf与glibc-all-in-one

patchelf可以修改elf中的ld和libc路径。

```
sudo apt intall patchelf
```

源里的不是最新的，大家也可以去GitHub仓库里面找项目源码

使用方法主要就是`--set-interpreter`&&`--set-rpath`两个参数

```
patchelf --set-interpreter [ld.so路径] --set-rpath [libc所在目录] [elf文件]
 举例说明(命令很长，可以因此本代码块用\来换行)
patchelf --set-interpreter ~/glibc-all-in-one/libs/2.23-0ubuntu3_amd64/ld-2.23.so \
 --set-rpath ~/glibc-all-in-one/libs/2.23-0ubuntu3_amd64 ./question3
```

`--set-interpreter`后面接ld的路径

`--set-rpath`后面接libc所在目录

为了替换libc所以我们在使用glibc-all-in-one这个项目，来快速下载我们的libc库

安装glibc-all-in-oen

```
git clone https://github.com/matrix1001/glibc-all-in-one.git
```

使用演示

列出来的都是可以下载的版本

使用download下载

下载完之后去libs目录下找ld与libc，2.35是空的，害，不过有debug的源码，好吧啥也不是

看看之前下载的2.23的libc

尝试给程序换一下libc

首先sort的libc是系统默认的

现在我们的sort的动态链接库就被替换了

后记

在这里没有介绍qume等一些内核pwn或者交叉编译所需要的工具，大家往后自己了解就ok了。

BYE BYE