Linux内核调试

Edvison / 2018-02-06 09:22:03 / 浏览数 4690 技术文章 技术文章 顶(0) 踩(0)

0x00 前言

这段时间开始学习了linux内核提权，也跟进看了一些漏洞。但是由于linux系统版本、内核版本的不同，驱动模块或者是某个函数的加载地址都是不同的，如果不能自己亲自
本文测试系统是Ubuntu12.04 64位。

0x01 准备环境

首先当然是准备需要调试的内核版本，linux的所有历史版本都可以在这里找到。

down下来后进入源码树根目录，开始配置内核，这里使用基于ncurse库编制的图形界面工具：

```
$ make menuconfig
```

由于我们需要使用kgdb调试内核，注意下面这几项一定要配置好：
KernelHacking -->

- 选中Compile the kernel with debug info
- 选中Compile the kernel with frame pointers
- 选中KGDB:kernel debugging with remote gdb，其下的全部都选中。

Processor type and features-->

- 去掉Paravirtualized guest support

KernelHacking-->

- 去掉Writeprotect kernel read-only data structures（否则不能用软件断点）

保存config文件之后make、make modules_install、make install编译安装就好。
具体安装编译内核可以看我另一篇笔记

0x02 使用kvm、gdb调试内核

先介绍一下现在用得比较多的调试内核方法吧，简单地说就是在linux系统里再装一个kvm虚拟机配置好gdb远程调试支持，然后在linux主机上连接调试。但是我因为电脑配
总之还是讲一下怎么调试吧。
查看cpu是否支持，如果没有输出，要在虚拟机设置里选上Inter VT:

```
$ grep vmx /proc/cpuinfo
```

安装：

```
$ sudo apt-get install kvm qemu libvirt-bin virtinst virt-manager virt-viewer
```

需要将自己添加进kvm、libvirtd用户组：

```
$ sudo adduser ■■■ kvm
$ sudo adduser ■■■ libvirtd
$ groups
```

打开libvirt-bin服务：

```
$ sudo service libvirt-bin start
```

然后打开virt-manager开始装虚拟机:

```
$ gksudo virt-manager
```

这个图形界面的virt-manager和vmware差不多，就这样直接装虚拟机就好了。
然后添加下面配置使虚拟机支持gdb调试：

```
<qemu:commandline>
   <qemu:arg value='-S'/>
   <qemu:arg value='-gdb'/>
   <qemu:arg value='tcp::1234'/>
</qemu:commandline>
```

将要调试的内核源码树复制进虚拟机，准备调试。
之后在主机端进入gdb调试：

```
$ gdb vmlinux
(gdb) target remote 127.0.0.1:1234 //■■■■■,■■■■■■■■■■
(gdb) b drv_open //■■■■■■■■■■■■■■■
(gdb) c
Countinuing.  //■■■■■■■■■■
```

如果要调试驱动模块的话，要在虚拟机里加载该模块：
```
$ insmod drv.ko
```
然后再回到主机开始调试。

0x03 用gdb和qemu调试内核

用qemu其实和上面kvm的差不多（qemu也已经在上面那步装好了），这里介绍一个比较快速的方法。直接在qemu中启动已经编译好的Linux内核，不用再制作完整的启动
```
qemu-system-x86_64 -s -S -kernel arch/i386/boot/bzImage -hda /boot/initrd.img -append "root=/dev/hda" -gdb tcp::1234
```
命令中的bzImage和initrd.img就是内核源码树中对应目录下的文件。
如果不需要图形界面的话可以加上下面的参数：
```
-nographic
```

在主机用gdb调试连接：

```
$ gdb vmlinux
(gdb) target remote:1234
```

和上面的差不多可以开始调试了。

0x04 用kdb和kgdb调试内核驱动模块

接下来介绍如何用kgdb和gdb来调试内核驱动。
kgdb 是一个在 Linux 内核上提供完整的 gdb 调试器功能的补丁。通过串口连线以钩子的形式挂入目标调试系统进行工作，然后远程运行 gdb。使用 kgdb
时需要两个系统——一个用于运行调试器，另一个用于运行待调试的内核。
当然，用两台VM虚拟机也是可以调试内核的，渣电脑开心地都哭了(つд⊂)

首先准备两台VMware虚拟机，直接复制之前配置好的虚拟机来创建一个新的就好。
配置双机通信：
将准备好的两台虚拟机添加串行端口，如果有并行端口记得先删除。注意箭头处的设置，一个得是客户端一个是服务端。

然后来验证一下是否成功连接。
先在客户机使用下面命令，准备接收消息：
```
$ cat /dev/ttyS0
```
然后在目标机（开启服务端的那个）输入下面命令：
```
$ echo edvison > /dev/ttyS0
```
如果没有显示就试下ttyS[0~3]

配置串口调试：
两个虚拟机都更改/etc/default/grub文件，在GRUB_CMDLINE_LINUX_DEFAULT这项后面加上kgdboc=ttyS1,115200，如果不需要图形界面启动的话可以加上text（最好

然后update-grub一下，重启系统。
进入grub界面，选择要调试的内核版本。（没有进grub的话去把gurb文件里GRUB_HIDDEN_TIMEOUT这行注释掉）
出现下面的信息就说明配置成功了。

开始调试：
打开目标机，可以看到已经显示命令行界面了（之前配置里加了text）
编译、载入要调试的驱动（这里使用内核ROP的那个栗子，源码在这里可以找到。
打开客户机，使用gdb准备开始连接调试：

编译加载好驱动后，在目标机里查看我们调试的驱动的加载基址，由于我们的驱动程序能直接打印基址，所以用dmesg命令查看就好：

另外还可以用下面的命令查看模块基址：
```
cat /proc/modules | grep drv
```

然后在目标机下输入下面的命令，使目标机进入被调试的假死状态：
```
# echo g > /proc/sysrq-trigger
```

接下来就可以在客户机里连接上目标机了：

之后在客户机里加载符号文件，并给驱动的入口函数device_ioctl()和device_open()设置断点：

输入c让目标机继续运行。
然后我们在目标机调试trigger程序，来调用内核模块的device_ioctl函数。

```
#define _GNU_SOURCE
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "drv.h"
#define DEVICE_PATH "/dev/vulndrv"
int main(int argc, char **argv) {
   int fd;
   struct drv_req req;
   req.offset = atoll(argv[1]);
   //map = mmap((void *)..., ..., 3, 0x32, 0, 0);
   fd = open(DEVICE_PATH, O_RDONLY);
   if (fd == -1) {
       perror("open");
   }
   ioctl(fd, 0, &req);
   return 0;
}
```

在open函数和ioctl下下断：

调试到这里后断下（直接运行./trigger [offset]也能在客户机里断下），可以看到客户机已经进入了驱动程序的调试模式，接下来就可以随意进行调试了；）

0x05 参考链接

http://blog.nsfocus.net/gdb-kgdb-debug-application/
https://qemu.weilnetz.de/doc/qemu-doc.html#direct_005flinux_005fboot
https://www.ibm.com/developerworks/cn/linux/1508_zhangdw_gdb/index.html

点击收藏 | 2 关注 | 3
上一篇：渗透测试 -- VulnHub –... 下一篇：【代码审计】yxcms从伪XSS到...

1. 2 条回复

bsauce 2019-09-03 16:36:00

大佬，请问一下，调试CVE一般用哪种环境配置方法呢？

0 回复Ta


Edvison 2019-09-17 14:43:38

@bsauce 我一般是用的gdb+qemu的方法

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板