LAPORAN AKHIR

PERTEMUAN 5

(SQL SERVER SECURITY)

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

Kelompok : (Opsional)

LEMBAGA PENGEMBANGAN KOMPUTERISASI

UNIVERSITAS GUNADARMA

2021

PERTEMUAN : 5
Tujuan Aktivitas :
1. Memahami security principals dan securables di SQL Server
2. Memahami mode otentikasi di SQL Server
3. Memahami users dan logins istimewa di SQL Server
4. Dapat menerapkan password policy yang aman di SQL Server
5. Memahami dan mencegah SQL Injection di SQL Server

TAHAPAN PENGERJAAN

1. Ringkasan Materi

- Terdapat tiga level keamanan yang dapat diolah pada security principals SQL Server, yaitu Windows, SQL Server, dan database. Pada masing-masing level terdapat lagi security principals yang dapat diolah. Level-level tersebut adalah: Windoes level, SQL server level dan Database level

- Terdapat dua macam mode otentikasi pada SQL Server yaitu:

  1) Mode otentikasi Windows: Penggunaan mode ini paling cocok ketika database hanya diakses dalam satu lingkup organisasi.

  2) Mode otentikasi SQL Server dan Windows (mixed mode): Penggunaan mode ini paling cocok ketika database juga diakses oleh pengguna yang berada di luar lingkup suatu organisasi (diluar domain Windows) atau pun pengguna yang tidak menggunakan perangkat Windows.

- Logins dan Users Istimewa di SQL Server:

  1) Administrators group, sebuah grup lokal dalam server database. Anggota dari grup ini biasanya akun local Administrator user dan user lainnya yang telah diatur sebagai administrator dalam lokal sistem.

  2) Administrator, sebuah user account lokal di server. Akun ini memberikan hak akses sebagai administrator pada sistem. Jika SQL Server terpasang di Windows domain, maka administrator account biasanya memiliki hak akses secara domain juga.

  3) Sa login, adalah aku sistem administrator dengan model keamanan baru yang telah terintegrasi dan diperluas, sa tidak lagi dibutuhkan. Sa disediakan untuk kompabilitas terhadap SQL Server versi sebelumnya.

  4) Guest User adalah pengguna khusus yang dapat Anda tambahkan ke database untuk memungkinkan seseorang dengan login SQL Server yang valid untuk mengakses database.

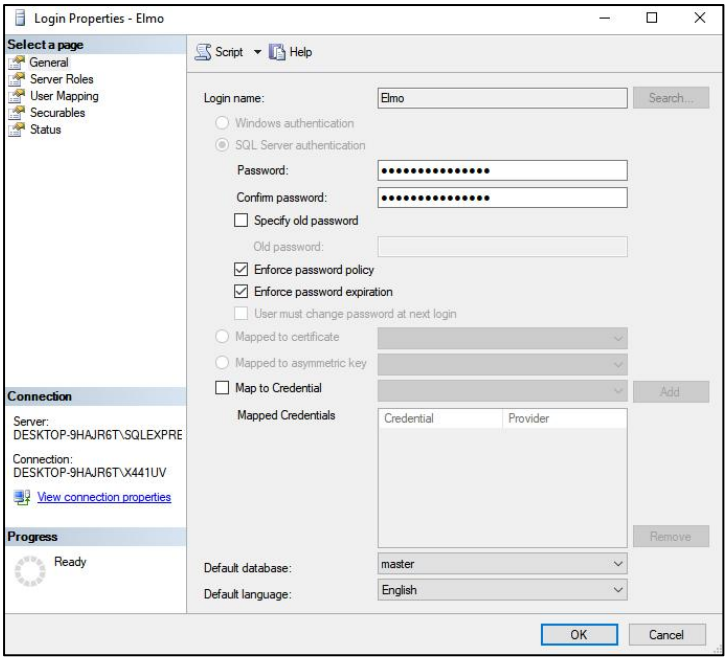
2. **Langkah-Langkah**

   a. **Manakah mode otentikasi yang lebih baik antara mode Windows dan mode campuran? Sertakan alasannya!**

      Mode otentikasi Windows, manfaat utama mode ini adalah menggunakan enkripsi untuk mengotentikasi pengguna SQL Server, selain itu penggunaan mode ini memungkinkan pengguna untuk memusatkan administrasi akun untuk seluruh perusahaan di satu direktori.

   b. **Buatlah password policy untuk perusahaan x pada Local/Domain Security Policy, berikan alasan yang konkrit untuk masing-masing kebijakan/policy yang telah Anda tentukan.**

      Domain policy digunakan untuk memastikan bahwa password login telah memenuhi kebijakan yang ada memiliki beberapa keuntungan, terutama dalam proses audit, Administrator tidak perlu memeriksa secara manual apakah password login telah memenuhi standar yang telah ditetapkan atau belum.



c. **Buatlah store procedure yang dapat diinjeksi dan aman dari sql injection. Store procedure harus mengembalikan tabel yang berisi nama pekerja dan nama departemen berdasarkan manager pekerja tersebut. Injeksi yang dilakukan yaitu injeksi yang mengembalikan daftar semua akun login yang ada dalam SQL Server instance yang sedang diserang.**

Membuat Stored Procedure:

```
CREATE PROCEDURE [sp_demo_injection01]
    @first_name NVARCHAR(MAX)
AS
BEGIN
DECLARE @sqlcmd NVARCHAR(MAX);

SET @sqlcmd = N'SELECT employee_id,first_name,last_name,email
                FROM dbo.employees
                WHERE first_name= '''+@first_name+''''';

                EXECUTE(@sqlcmd)
END
```

Messages

```
Command(s) completed successfully.
```

Mengeksekusi Procedure

```
Declare @var sysname
set @var = 'Alexander'
EXEC sp_demo_injection01 @var
```

Results | Messages

	employee_id	first_name	last_name	email
1	103	Alexander	Hunold	AHUNOLD
2	115	Alexander	Khoo	AKHOO

Menguji keamanan Store Procedure dari sqli njection.

```
CREATE PROCEDURE [sp_demo_injection02]
    @first_name NVARCHAR(MAX)
AS
BEGIN
    DECLARE @SQLCMD NVARCHAR(MAX);
    DECLARE @PARAMS NVARCHAR(MAX);
    SET @SQLCMD = N'SELECT employee_id,first_name,last_name,email
                FROM dbo.employees
                WHERE first_name= @first_name';
                SET @PARAMS = N'@first_name NVARCHAR(MAX)';
                EXECUTE sp_executesql @sqlcmd, @params, @first_name;
END
```

Messages

```
Command(s) completed successfully.
```

```
DECLARE @var sysname

SET @var = 'name employee''; delete from dbo.employees;
PRINT ''table employees telah dikosongkan!'';
            -- data on table employees has been deleted!!'
EXEC sp_demo_injection02 @var
```

Results Messages

employee_id	first_name	last_name	email

```
DECLARE @var sysname

SET @var = 'name employee''; delete from dbo.employees;
PRINT ''table employees telah dikosongkan!'';
            -- data on table employees has been deleted!!'
EXEC sp_demo_injection01 @var
```

Results Messages

```
(0 row(s) affected)

(107 row(s) affected)
table employees telah dikosongkan!
```

Menampilkan data security login

```
DECLARE @var sysname
SET @var = 'nama employee''; select * from master.sys.sql_logins;
PRINT '' TABLE USER Login'';
            -- Table Selected!!'
EXEC sp_demo_injection01 @var
```

Results Messages

employee_id	first_name	last_name	email

	name	principal_id	sid	type	type_desc
1	sa	1	0x01	S	SQL_LO
2	##MS_PolicyEventProcessingLogin##	256	0x0A6983CDF023464B9E86E4EEAB92C5DA	S	SQL_LO
3	##MS_PolicyTsqlExecutionLogin##	257	0x8F651FE8547A4644A0C06CA83723A876	S	SQL_LO

Menampilkan data login user

```
DECLARE @var sysname
SET @var = 'nama employee''; select * from master.sys.sql_logins;
PRINT '' TABLE USER Login'';
            -- Table Selected!!'
EXEC sp_demo_injection02 @var
```

Results Messages

employee_id	first_name	last_name	email