MANAJEMEN KERENTANAN DAN PEMULIHAN PADA BASIS DATA RELASIONAL

Holong Marisi Simalango, A.Md., S.T., M.Kom,
Gilang Bagus Ramadhan, A.Md.Kom.
Ahmadi Irmansyah Lubis, S.Kom., M.Kom.

Tujuan Pembelajaran:

Mahasiswa mampu menerapkan praktis dari konsep standar keamanan yang relevan pada basis data relasional

Setelah mempelajari control akses, otentikasi, dasar sql injection, enkripsi, dan control integritas data,

HAL KRUSIAL yang harus difokuskan → PROAKTIF dan REAKTIF dari keamanan basis data yaitu Pencegahan serangan dan memulihkan sistem.

Maka perdalam dan focus pada, sebagai berikut:

- Manajemen kerentanan,
- Pencegahan SQLi
- BackUp dan recovery
- Perencanaan Respon Insiden

MANAJEMEN KERENTANAN

Mempelajari cara mengidentifikasi, menilai, dan mengatasi kerentanan dalam basis data.

Tahap manajemen kerentanan:

- **IDENTIFIKASI**

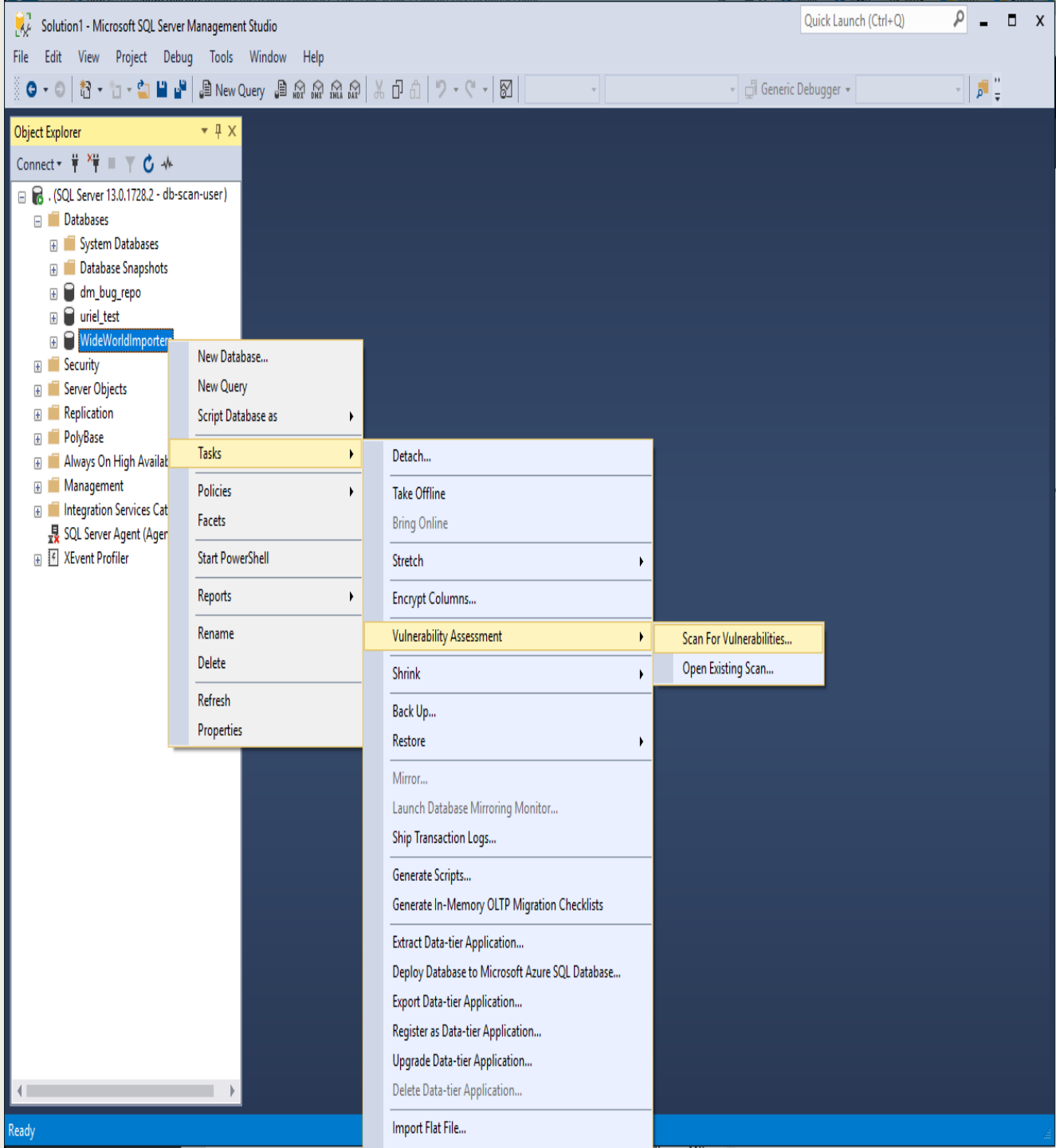
Proses tahap ini dimulai dengan pemindaian otomatis maupun audit secara manual, yang bertujuan menemukan **kelemahan** pada tingkat server (misalnya, patch yang hilang) dan **tingkat basis data (misalnya, pengaturan hak akses yang terlalu lemah)**

Hal-hal yang perlu dilakukan, yaitu:

1. *Audit* dan *Scanning*: menggunakan *vulnerability scanner* atau *manual review.*
2. *Database audit log*: mendeteksi query abnormal.
3. *Misconfiguration check*: port default, password lemah, user root aktif.

Vulnerability scanning

Mysql belum menyediakan Vulnerability scanning secara otomatis, maka menggunakan tools seperti Nikto (yang dijalankan melalui OS Linux), Acuneti, atau owasp ZAP. Namun, pada Ms. SQL Server menyediakan fitur ini.



Database audit log: query abnormal

- Penarikan / pengembalian data berlebihan

Biasanya data (Misalnya pengguna) hanya menarik data 10-100 baris data, namun muncul query baru mencoba seluruh isi baris.

SELECT * FROM T_USERS;

- Frekuensi query tidak wajar

Dalam periode waktu singkat dalam satu menit, banyak pengguna hingga ribuan mencoba akses. Biasanya hal ini terindikasi brute force.

Database audit log: query abnormal

- Pola query yang mencurigakan

Terdapat sisipan query mencurigakan.

SELECT * FROM T_USERS

WHERE

USERNAME = ‘ADMIN’ OR ‘1’=‘1’;

- Akses yang tidak sesuai privilege maupun role

Pengguna biasa (misalnya role staf) mencoba perintah DROP TABLE, ALTER USER, atau GRANT ALL PRIVILEGES

Misconfiguration check:

Resiko yang sering ditemukan pada misconfiguration check yaitu **port default, password lemah**, dan **user root aktif.**

Port default pada Mysql yaitu 3306 dapat dilakukan perubahan (*security-by-obscurity*) misalnya 3307.

NAMUN!
Pastikan bahwa nomor port yang diubah tidak bentrok dengan service lain.

Jika MySQL yang dikelola seperti RDS, Cloud SQL, mungkin tidak bisa menggantikan port

Misconfiguration check:

User root aktif:
akun root (super admin DB) yang masih aktif untuk login maupun dapat diakses dari host remote (‘root’@’%’), sehinga memberi celah bila database dicuri, dan dapat melakukan control penuh pada database seperti DROP DB maupun hapus log

Periksa akun yang berhubungan dengan root.

Hapus (rekomendasi nonaktifkan) akun root yang remote.

Buat akun admin account secara terpisah untuk pengaksesan secara remote.

Tahap manajemen kerentanan: (cont.)

- **PENILAIAN**

Proses tahap ini setelah kerentanan diidentifikasi, kerentanan tersebut dikategorikan berdasarkan tingkat keparahannya (*high, medium, low*) untuk menentukan mana yang harus diperbaiki terlebih dahulu

Contoh penentuan kategori

1. High : SQLi
2. Medium : akun default tidak bisa dihapus
3. Low : Log file tanpa enkripsi

Tahap manajemen kerentanan: (cont.)

- **MITIGASI**

Proses tahap ini menindaklanjuti dari mengatasi ancaman maupun masalah tersebut.

Kegiatan yang dilakukan, misalnya:

1. Patching: memperbaharui DBMS secara berkala
2. Hardening: menutup port yang tidak perlu, menonaktifkan akun default, lakukan enkripsi pada backup.
3. Penerapan Role base access control (RBAC)

PERENCANAAN RESPON INSIDEN / *INCIDENT RESPOND PLAN*

Meminimal jika suatu insiden terjadi

1. Isolasi sistem dan database

Jika dari awal terdeteksi adanya aktivitas abnormal pada basis data yang bisa dilihat dari audit log maupun alert dari DBA.

Kemudian lakukan pemutusan koneksi ke pengguna yang mencurigakan, blokir IP pengguna.

Dokumentasikan segera dari aktifitas pengguna tersebut, dan nonaktifkan / ubah akun user yang digunakan sebagai mediator akses sistem maupun DBMS

2. Analisis dan Investigasi

Pada tahap lakukan pencarian sumber dari masalah sesuai tahap manajemen kerentanan.

Kemudian hapus hal maupun aktifitas yang mencurigakan ataupun berbahaya, biasanya hal / aktifitas tersebut, yaitu:

- Basis data baru illegal
- Query injeksi yang masih berjalan di background
- Adanya backdoor / trigger mencurigakan.

Dokumentasikan segala aktifitas tersebut sebagai bukti digital untuk pelaporan berwajib

3. Pelaporan Insiden

Lakukan pelaporan insiden terhadap pihak berwewenang yang relevan, misalnya otoritas keamanan siber di Indonesia yaitu

- Badan Siber dan Sandi Negara (BSSN),
- Polri (pada direktorat Tindak Pidana Siber-Bareskrim), dan
- TNI(pada Satuan Siber TNI jika berhubungan dengan kedaulatan dan keamanan nasional)

4. Pemulihan Data

Pemulihan data ini dilakukan hampir sama dengan import data dari backup data yang terakhir dilakukan

Di tahap ini juga lakukan meeting dengan pihak internal yaitu petinggi Perusahaan dan tim IT internal untuk membahas kejadian ini, seperti:

- Kronologi insiden,
- Apa yang telah dilakukan
- Apa yang harus diperbaiki
- Persiapan untuk pelatihan ulang untuk tim

SQL INJECTIONS PREVENTIONS / PENCEGAHAN SQLI

PRAKTIKUM

Teknik : Prepared Statements ((Parameterized Queries)

- Buat tabel sederhana

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50),
  password VARCHAR(50)
);
```

- Isikan data username & password : admin, 1234

- Lakukan prepared statement

```
PREPARE stmt FROM 'SELECT * FROM users WHERE username = ? AND password = ?';
SET @user = 'admin';
SET @pass = '1234';
EXECUTE stmt USING @user, @pass;
```

→ Jika terjadi percobaan user ‘ADMIN’ OR ‘1’=‘1’, hasilnya gagal karena dianggap string, bukan bagian query

Teknik : Prepared Statements (Stored Procedures)

- Gunakan DELIMIITER

```
DELIMITER //

CREATE PROCEDURE checkUser(IN
uname VARCHAR(50), IN upass
VARCHAR(50))

BEGIN

    SELECT * FROM users WHERE
username = uname AND password =
upass;

END //

DELIMITER ;
```

- Lakukan pemanggilan prosedur

```
CALL checkUser('admin', '1234’);
```

→ Jika terjadi percobaan user ’ OR ‘1’=‘1’, hasilnya diproses sebagai string, bukan query injeksi sehingga tidak dianggap untuk manipulasi input langsung

BACKUP DAN RECOVERY

PRAKTIKUM

Aktifitas backup dan recovery bertujuan untuk Langkah terakhir untuk mempertahankan dari kehilangan data akibat :

- Kegagalan sistem,
- Bencana alam,
- Serangan siber
- Serangan terhadap backup

Sehingga, basis data relasional memiliki fungsionalitas import dan export data yang memungkinkan pembuatan backup data dengan cepat.

Backup

Backup basis data pada MySQL dapat dilakukan dengan beberapa cara yaitu mysqldump, phpMyAdmin, ataupun dari Mysql Workbench.

Pada phpMyAdmin:

1. Login ke phpMyAdmin.
2. Pilih database yang ingin dibackup.
3. Klik tab Export.
4. Pilih metode:

Quick → lebih sederhana, langsung ekspor semua.

Custom → bisa pilih tabel tertentu, format (SQL, CSV, dll).

5. Klik Go, lalu pilih file .sql akan terunduh.

Recovery

Sama halnya seperti backup, ada beberapa metode dilakukan untuk recovery, menggunakan mysqlimport, phpMyAdmin, mySql Workbench

Pada phpMyAdmin:

1. Login ke phpMyAdmin.
2. Buat database baru (misalnya sistemPBL_restore).
3. Pilih database tersebut.
4. Klik tab Import.
5. Pilih file backup .sql → klik Go.

Tunggu sampai proses selesai.