PANDUAN PENANGANAN INSIDEN SERANGAN SQL INJECTION

MAHKAMAH AGUNG
COMPUTER SECURITY INCIDENT RESPONSE TEAM
[MA-CSIRT]

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, Tuhan Yang Maha Esa, atas segala limpahan rahmat, nikmat serta karunia-Nya yang tak ternilai dan tak dapat dihitung sehingga kami dapat menyelesaikan penyusunan "Panduan Penanganan Insiden SQL Injection". Panduan ini disusun dalam rangka memberikan acuan bagi pihak yang berkepentingan dalam penanganan insiden serangan SQL Injection. Panduan ini berisikan langkah-langkah yang harus diambil apabila terjadi serangan SQL Injection, yang dimulai dari tahap persiapan sampai dengan tahap pembuatan laporan dari penanganan serangan. Panduan ini tentu saja masih banyak kekurangan dan masih jauh dari kesempurnaan karena keterbatasan ilmu dan referensi kami. Untuk itu, kami selalu berusaha melakukan evaluasi dan perbaikan secara berkala agar bisa mencapai hasil yang lebih baik lagi.

Akhir kata, kami ucapkan terima kasih kepada segala pihak yang telah membantu dalam penyusunan penduan ini.

Jakarta, Maret 2023

MA-CSIRT,

KEPALA MA-CSIRT

DAFTAR ISI

PROSEDUR PENANGANAN SERANGAN SQL INJECTION

1. PENDAHULUAN

Serangan *SQL Injection* merupakan jenis eksploitasi keamanan halaman web, dimana penyerang menyisipkan kode-kode SQL melalui *formulir/form* kemudian memanipulasi URL berdasarkan pada parameter sql. Serangan *SQL Injection* adalah serangan yang berupa menginjeksi perintah SQL melalui *form input* data, yang kemudian diteruskan menuju *database* untuk dieksekusi, dengan tujuan mengakses data sensitif pada *database.*

2. TUJUAN

Secara umum, tujuan panduan ini dimaksudkan untuk membantu organisasi memahami tentang penanganan serangan *SQL Injection.* Sedangkan secara khusus adalah sebagai berikut:

a) Memastikan adanya sumber daya yang memadai untuk menangani serangan yang terjadi;
b) Melakukan pengumpulan informasi yang akurat;
c) Meminimalisir dampak dari serangan yang terjadi;
d) Mencegah adanya serangan lanjutan dan mencegah kerusakan agar tidak lebih meluas.

3. RUANG LINGKUP

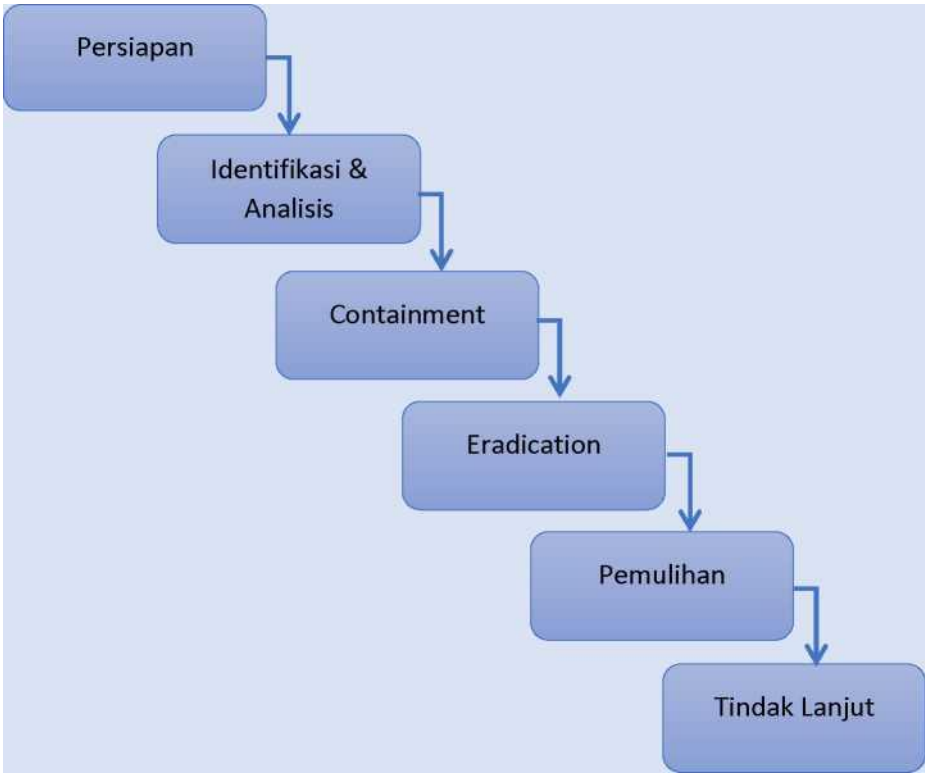
Panduan ini berisi langkah-langkah yang harus diambil apabila terjadi serangan *SQL Injection,* yang dimulai dari tahap persiapan sampai dengan tahap pembuatan laporan dari penanganan serangan. Panduan ini dapat dijadikan acuan bagi semua individu atau tim *(administrator,* pengelola TI, tim respon insiden keamanan komputer) yang bertanggung jawab untuk mencegah, mempersiapkan atau menangani serangan *SQL Injection* pada suatu *website.*

4. PROSEDUR PENANGANAN SERANGAN SQL INJECTION

Penanganan serangan *SQL Injection* ditujukan utuk mencapai hal-hal sebagai berikut:

a) Mengumpulkan informasi sebanyak mungkin tentang serangan *SQL Injection*;
b) Menghalangi atau mencegah eskalasi kerusakan yang disebabkan oleh serangan tersebut;
c) Mengumpulkan bukti terkait serangan *SQL Injection*;
d) Mengambil langkah-langkah proaktif untuk mengurangi kemungkinan terjadinya serangan *SQL Injection* di masa depan.

Supaya tujuan di atas dapat terlaksana dengan baik, maka penanganan terhadap serangan *SQL Injection* dilakukan dalam beberapa tahap sebagai berikut:



Gambar 1. Tahap Penanganan Serangan *SQL Injection*

4.1. Persiapan

Dalam melakukan penanganan serangan *SQL Injection,* perlu adanya tahap persiapan dengan prosedur sebagai berikut :

a) Pembentukan tim respon. Tim dapat berasal dari institusi yang mengalami

serangan (internal) atau juga bisa berasal dari luar institusi (eksternal) jika memang diperlukan. Anggota tim memiliki pengetahuan tentang *SQL Injection* dan memiliki kemampuan penanganannya;

b) Menyiapkan dokumen yang dibutuhkan dalam proses penanganan serangan *SQL Injection.* Dokumen ini antara lain adalah :

- Panduan penanganan insiden serangan siber;
- Formulir penanganan insiden serangan siber;
- Diagram yang menggambarkan hubungan antar komponen-komponen aplikasi yang membangun *website* (web *server,* aplikasi web, daftar user, diagram *network*).

c) Menyiapkan tool dan media yang dibutuhkan untuk penanganan. Misalnya Notepad ++ untuk membaca log, IDS/IPS, SQL Map, Accunetix /Nessus.

4.2. Identifikasi dan Analisis

Tujuan dari proses identifikasi dan analis adalah:

a) Memahami sifat dan ruang lingkup kejadian;

b) Mengumpulkan informasi yang cukup tentang serangan *SQL Injection* sehingga tim respon dapat memprioritaskan langkah selanjutnya dalam menangani serangan tersebut, yang biasanya diikuti dengan penahanan sistem.

Pada tahap ini dilakukan proses identifikasi untuk memastikan telah terjadi serangan *SQL Injection* dan mendeteksi sumbernya.

Langkah-langkah yang dapat diambil pada tahap identifikasi dan analisis antara lain:

a) Memeriksa *alert* dan *anomalies* dari perangkat IDS atau IPS;

b) Melakukan *error checking* melalui form atau url dengan memberikan karakter atau sebuah simbol. Misalnya:

- Melalui *form login,* memasukan pada *username* dan *password* berupa karakter-karakter yang digunakan *SQL Injection,* seperti:

  OR 1=1 --

  OR 1=2 --

  OR 'a'='a'

- Melalui ur/, menambahkan karakter-karakter yang digunakan *SQL*

Injection, seperti *sing/e quote, doub/e minus.*

c) Memeriksa semua */og (error /og, access /og, database /og, firewa// /og).* Lokasi */og f/e* secara *defau/t* berada pada var//og, */og* tersebut menyimpan seluruh aktivitas yang terjadi pada sistem;

d) Memeriksa adanya *command /ine*, *string-string* yang digunakan untuk menyerang;

e) Memeriksa isi *database* untuk mencari *script* yang berbahaya, dan mengecek apakah ada penambahan user secara tidak sah;

f) Memeriksa apakah ada file atau script berbahaya (*trojan*, *ma/icious fi/e*, *backdoor)* yang ditanamkan pada web *server*;

g) Menggunakan tool untuk memeriksa kerentanan. *Too/* yang dapat digunakan diantaranya Acunetix, SQLMap, *SQL Injection too/s.*

Mengukur dampak dari terjadinya *SQL Injection* adalah

a) Terhadap kelangsungan proses bisnis, indikatornya adalah seberapa besar dari fungsi-fungsi bisnis yang terdapat pada website mengalami gangguan;

b) Terhadap sistem dan informasi, apakah penyerang melakukan distribusi *ma/ware*, membuat *backdoor* atau melakukan *web defacement.* Selain itu, apakah ada data dan informasi yang berubah atau terhapus.

4.3. Containment

Setelah dipastikan bahwa memang benar telah terjadi serangan *SQL Injection*, maka dilakukan proses berikutnya dengan tujuan:

1. Tidak terjadi kerusakan lebih dalam;
2. Mencegah penyerang masuk lebih dalam ke sistem yang terkena dampak;
3. Melindungi *server-server* lain yang terhubung dengan aplikasi web.

Prosedur yang dilakukan pada tahap ini adalah:

a) Melakukan proses backup semua data yang terdapat pada web server. untuk keperluan forensik dan pengumpulan bukti-bukti. Backup sebaiknya ditempatkan pada hard disk eksternal;

b) Jika sumber penyerangan berasal dari sistem lain pada jaringan, maka

putuskan secara fisik koneksi tersebut dan lakukan investigasi sumber tersebut.

4.4. Eradication

Tahap *Eradication* pada penanganan serangan *SQL Injection* adalah untuk menghapus *file* / *script* serta menutup sumber serangan.

Prosedur untuk melakukan proses ini dapat dilakukan dengan cara berikut:

a) Memeriksa apakah terdapat *malicious file, backdoor, rootkit* atau kode-kode berbahaya lainnya yang berhasil ditanamkan pada server dan segera menghapusnya;

b) Jika terdapat kode SQL yang mengakses IP tertentu maka perlu melakukan block /menutup sumber serangan (block IP dan *Port).*

4.5. Pemulihan

Pemulihan merupakan tahap untuk mengembalikan seluruh sistem bekerja normal seperti semula. Prosedur yang dapat dilakukan sebagai berikut:

a) Mengubah kredential password pengguna. Hal ini untuk mengantisipasi apabila password pengguna telah diketahui oleh penyerang;

b) Melakukan recovery database pada aplikasi web;

c) Jika SQL Injection menyebabkan web defacement, gunakan panduan penanganan insiden web defacement;

d) Jika SQL Injection menyebabkan insiden malware, gunakan panduan penanganan insiden malware;

e) Menutup semua kerentanan yang telah diketahui;

f) Membatasi akses root langsung ke database;

g) Melakukan filter terhadap input yang dimasukkan oleh pengguna;

h) Mematikan atau menyembunyikan pesan-pesan *error* yang keluar dari SQL *Server* yang berjalan;

i) Patching terhadap aplikasi yang rentan, melakukan *upgrade* terhadap aplikasi web yang masih memiliki kerentanan;

j) Melakukan *penetration testing* untuk mengetahui celah-celah keamanan yang mungkin masih terdapat pada *website.*

4.6. Tindak Lanjut

Tahap ini adalah fase di mana semua dokumentasi kegiatan yang dilakukan dicatat sebagai referensi untuk di masa mendatang. Tujuan dari tahap ini adalah untuk:

a) Pelaporan, membuat laporan mengenai langkah-langkah dan hasil yang telah didapatkan pada penanganan serangan *SQL Injection*;

b) Mengambil pelajaran dan membuat rekomendasi untuk mencegah terjadi lagi.

c) Prosedur yang dapat dilakukan adalah sebagai berikut:

d) Membuat dokumentasi dan laporan terkait penanganan serangan SQL Injection;

e) Menuliskan *tools* apa saja yang digunakan dalam penanganan serangan injeksi sql;

f) Menuliskan bukti-bukti yang ditemukan, hal ini terkait dengan proses hukum kedepannya;

g) Memberikan analisa dan penjelasan apa yang harus dilakukan sehingga serangan serupa tidak terulang kembali;

h) Membuat evaluasi dan rekomendasi.