Report

CVE-2022-21664

Requested by guest
Request date 2022-04-24

What is CVE ?

CVE, kependekan dari Common Vulnerabilities and Exposures, adalah daftar kelemahan keamanan komputer yang diungkapkan kepada publik. Ketika seseorang merujuk ke CVE, itu berarti kelemahan keamanan yang telah diberi nomor ID CVE. Penasihat keamanan yang dikeluarkan oleh vendor dan peneliti hampir selalu menyebutkan setidaknya satu ID CVE. CVE membantu profesional TI mengoordinasikan upaya mereka untuk memprioritaskan dan mengatasi kerentanan ini untuk membuat sistem komputer lebih aman. more

What is CWE ?

CWE™ adalah daftar jenis kelemahan perangkat lunak dan perangkat keras yang dikembangkan komunitas. Ini berfungsi sebagai bahasa umum, tongkat pengukur untuk alat keamanan, dan sebagai dasar untuk identifikasi kelemahan, mitigasi, dan upaya pencegahan. more

What is CVSS ?

Common Vulnerability Scoring System (CVSS) adalah kerangka kerja terbuka untuk mengkomunikasikan karakteristik dan tingkat keparahan kerentanan perangkat lunak. CVSS terdiri dari tiga kelompok metrik: Basis, Temporal, dan Lingkungan. Metrik Dasar menghasilkan skor mulai dari 0 hingga 10, yang kemudian dapat dimodifikasi dengan menilai metrik Temporal dan Lingkungan. Skor CVSS juga direpresentasikan sebagai string vektor, representasi tekstual terkompresi dari nilai yang digunakan untuk mendapatkan skor. Dengan demikian, CVSS sangat cocok sebagai sistem pengukuran standar untuk industri, organisasi, dan pemerintah yang membutuhkan skor tingkat keparahan kerentanan yang akurat dan konsisten. Dua penggunaan umum CVSS adalah menghitung tingkat keparahan kerentanan yang ditemukan pada sistem seseorang dan sebagai faktor dalam memprioritaskan aktivitas remediasi kerentanan. Database Kerentanan Nasional (NVD) memberikan skor CVSS untuk hampir semua kerentanan yang diketahui. more

What is CAPEC ?

Upaya Common Attack Pattern Enumeration and Classification (CAPEC™) menyediakan katalog pola serangan umum yang tersedia untuk umum yang membantu pengguna memahami bagaimana musuh mengeksploitasi kelemahan dalam aplikasi dan kemampuan lain yang mendukung dunia maya. more

CVE ID - CVE-2022-21664

CWE ID - CWE-89

CVSS SC- 6.5

CAPEC ID	Name	Description	Solution
7	Blind SQL Injection	Blind SQL Injection dihasilkan dari mitigasi yang tidak memadai untuk SQL Injection. Meskipun menekan pesan kesalahan database dianggap sebagai praktik terbaik, penekanan saja tidak cukup untuk mencegah SQL Injection. Blind SQL Injection adalah bentuk SQL Injection yang mengatasi kurangnya pesan kesalahan. Tanpa pesan kesalahan yang memfasilitasi SQL Injection, musuh membuat string input yang menyelidiki target melalui ekspresi SQL Boolean sederhana. Musuh dapat menentukan apakah sintaks dan struktur injeksi berhasil berdasarkan apakah kueri dijalankan atau tidak. Diterapkan secara berulang, musuh menentukan bagaimana dan di mana target rentan terhadap SQL Injection.	Security by Obscurity bukanlah solusi untuk mencegah SQL Injection. Daripada menekan pesan kesalahan dan pengecualian, aplikasi harus menanganinya dengan baik, mengembalikan halaman kesalahan khusus atau mengarahkan pengguna ke halaman default, tanpa mengungkapkan informasi apa pun tentang database atau internal aplikasi. Validasi input yang kuat - Semua input yang dapat dikontrol pengguna harus divalidasi dan difilter untuk karakter ilegal serta konten SQL. Kata kunci seperti UNION, SELECT atau INSERT harus difilter selain karakter seperti tanda kutip tunggal (') atau komentar SQL (--) berdasarkan konteks kemunculannya.

109	Object Relational Mapping Injection	Penyerang memanfaatkan kelemahan yang ada dalam kode lapisan akses basis data yang dihasilkan dengan alat Pemetaan Relasional Objek (ORM) atau kelemahan dalam cara pengembang menggunakan kerangka kerja ketekunan untuk menyuntikkan perintah SQL-nya sendiri untuk dieksekusi terhadap basis data yang mendasarinya . Serangan di sini mirip dengan injeksi SQL biasa, kecuali bahwa aplikasi tidak menggunakan JDBC untuk berbicara langsung ke database, tetapi menggunakan lapisan akses data yang dihasilkan oleh alat atau kerangka kerja ORM (misalnya Hibernate). Sementara sebagian besar kode waktu yang dihasilkan oleh alat ORM berisi metode akses aman yang kebal terhadap injeksi SQL, kadang-kadang karena beberapa kelemahan dalam kode yang dihasilkan atau karena fakta bahwa pengembang gagal menggunakan metode akses yang dihasilkan dengan benar, SQL injeksi masih memungkinkan.	Ingatlah untuk memahami cara menggunakan metode akses data yang dihasilkan oleh alat / kerangka kerja ORM dengan benar dengan cara yang akan memanfaatkan mekanisme keamanan bawaan dari kerangka kerja Pastikan untuk tetap up to date dengan pembaruan keamanan yang relevan dengan kerangka kerja persistensi yang digunakan dalam Anda aplikasi.
110	SQL Injection through SOAP Parameter Tampering	Penyerang memodifikasi parameter pesan SOAP yang dikirim dari konsumen layanan ke penyedia layanan untuk memulai serangan injeksi SQL. Di sisi penyedia layanan, pesan SOAP diuraikan dan parameter tidak divalidasi dengan benar sebelum digunakan untuk mengakses database dengan cara yang tidak menggunakan pengikatan parameter, sehingga memungkinkan penyerang untuk mengontrol struktur kueri SQL yang dieksekusi. Pola ini menjelaskan serangan injeksi SQL dengan mekanisme pengiriman menjadi pesan SOAP.	Memvalidasi dan membersihkan/menolak input pengguna dengan benar di penyedia layanan. Pastikan bahwa pernyataan yang disiapkan atau mekanisme lain yang memungkinkan pengikatan parameter digunakan saat mengakses database dengan cara yang akan mencegah data yang diberikan penyerang untuk mengontrol struktur kueri yang dieksekusi. Pada tingkat basis data, pastikan bahwa pengguna basis data yang digunakan oleh aplikasi dalam konteks tertentu memiliki hak istimewa minimum yang diperlukan untuk basis data yang diperlukan untuk melakukan operasi. Jika memungkinkan, jalankan kueri terhadap tampilan yang dibuat sebelumnya, bukan tabel secara langsung.

66	SQL Injection	Serangan ini mengeksploitasi perangkat lunak target yang membuat pernyataan SQL berdasarkan input pengguna. Penyerang membuat string input sehingga ketika perangkat lunak target membuat pernyataan SQL berdasarkan input, pernyataan SQL yang dihasilkan melakukan tindakan selain yang dimaksudkan oleh aplikasi. SQL Injection dihasilkan dari kegagalan aplikasi untuk memvalidasi input dengan tepat. Ketika input yang dikontrol pengguna yang dibuat khusus yang terdiri dari sintaks SQL digunakan tanpa validasi yang tepat sebagai bagian dari kueri SQL, dimungkinkan untuk mengumpulkan informasi dari database dengan cara yang tidak dipertimbangkan selama desain aplikasi. Tergantung pada database dan desain aplikasi, mungkin juga untuk memanfaatkan injeksi agar database menjalankan perintah terkait sistem pilihan penyerang. SQL Injection memungkinkan penyerang untuk berbicara langsung ke database, sehingga melewati aplikasi sepenuhnya. Injeksi yang berhasil dapat menyebabkan pengungkapan informasi serta kemampuan untuk menambah atau mengubah data dalam database. Agar berhasil menyuntikkan SQL dan mengambil informasi dari database, penyerang:	Validasi input yang kuat - Semua input yang dapat dikontrol pengguna harus divalidasi dan disaring untuk karakter ilegal serta konten SQL. Kata kunci seperti UNION, SELECT atau INSERT harus difilter selain karakter seperti tanda kutip tunggal (') atau komentar SQL (--) berdasarkan konteks kemunculannya. Penggunaan kueri berparameter atau prosedur tersimpan - Parameterisasi menyebabkan input dibatasi ke domain tertentu, seperti string atau bilangan bulat, dan input apa pun di luar domain tersebut dianggap tidak valid dan kueri gagal. Perhatikan bahwa SQL Injection dimungkinkan bahkan dengan adanya prosedur tersimpan jika kueri akhirnya dibuat secara dinamis. Penggunaan halaman kesalahan khusus - Penyerang dapat mengumpulkan informasi tentang sifat kueri dari pesan kesalahan deskriptif. Validasi input harus digabungkan dengan halaman kesalahan khusus yang menginformasikan tentang kesalahan tanpa mengungkapkan informasi tentang database atau aplikasi.
470	Expanding Control over the Operating System from the Database	Seorang penyerang dapat memanfaatkan akses yang diperoleh ke database untuk membaca / menulis data ke sistem file, kompromi sistem operasi, membuat terowongan untuk mengakses mesin host, dan menggunakan akses ini untuk berpotensi menyerang mesin lain di jaringan yang sama dengan jaringan yang sama. mesin basis data. Secara tradisional serangan injeksi SQL dipandang sebagai cara untuk mendapatkan akses baca yang tidak sah ke data yang disimpan dalam database, memodifikasi data dalam database, menghapus data, dll. Namun, hampir setiap sistem sistem manajemen basis data (DBMS) menyertakan fasilitas yang jika disusupi memungkinkan penyerang akses lengkap ke sistem file, sistem operasi, dan akses penuh ke host yang menjalankan database. Penyerang kemudian dapat menggunakan akses istimewa ini untuk meluncurkan serangan berikutnya. Fasilitas ini termasuk memasukkan ke dalam shell perintah, membuat fungsi yang ditentukan pengguna yang dapat memanggil perpustakaan tingkat sistem yang ada di mesin host, prosedur tersimpan, dll.	Implementasi: Hapus / nonaktifkan semua fungsi sistem DBMS yang tidak dibutuhkan / tidak digunakan yang memungkinkan penyerang meningkatkan hak istimewa jika dikompromikan

108		Command Line Execution through SQL Injection	Penyerang menggunakan metode injeksi SQL standar untuk menyuntikkan data ke baris perintah untuk dieksekusi. Ini dapat dilakukan secara langsung melalui penyalahgunaan arahan seperti MSSQL_xp_cmdshell atau secara tidak langsung melalui injeksi data ke dalam database yang akan ditafsirkan sebagai perintah shell. Beberapa waktu kemudian, aplikasi backend yang tidak bermoral (atau bisa menjadi bagian dari fungsionalitas aplikasi yang sama) mengambil data yang dimasukkan yang disimpan dalam database dan menggunakan data ini sebagai argumen baris perintah tanpa melakukan validasi yang tepat. Data berbahaya lolos dari bidang data itu dengan memunculkan perintah baru untuk dieksekusi di host.	Nonaktifkan MSSQL xp_cmdshell directive pada database Validasi data dengan benar (secara sintaksis dan semantik) sebelum menulisnya ke database. Jangan secara implisit mempercayai data yang disimpan dalam database. Validasi ulang sebelum digunakan untuk memastikan aman digunakan dalam konteks tertentu (misalnya sebagai argumen baris perintah).

CAPEC ID	Name	STEP 1	STEP 2	STEP 3
569	Blind SQL Injection	[Hipotesiskan kueri SQL dalam aplikasi]Hipotesis yang dihasilkan terkait kueri SQL dalam aplikasi. Misalnya, musuh dapat berhipotesis bahwa input mereka diteruskan langsung ke kueri yang terlihat seperti: SELECT * FROM orders WHERE ordernum = ___atauSELECT * FROM orders WHERE ordernum IN (___)orSELECT * FROM orders WHERE ordernum in (_____) ORDER BY _____ Tentu saja, ada banyak kemungkinan lain.. - Meneliti jenis kueri SQL dan menentukan yang mana yang dapat digunakan di berbagai tempat dalam sebuah aplikasi.	[Tentukan cara memasukkan informasi ke dalam kueri]Tentukan cara memasukkan informasi ke dalam kueri dari langkah sebelumnya sehingga injeksi tidak memengaruhi logikanya. Misalnya, berikut ini adalah kemungkinan suntikan untuk kueri tersebut: 5' OR 1=1; --dan5) ATAU 1=1; --andordernum DESC; --. - Tambahkan klausa ke kueri SQL sehingga logika kueri tidak berubah.. - Tambahkan penundaan ke kueri SQL jika server tidak memberikan pesan kesalahan yang jelas (mis. WAITFOR DELAY '0:0:10' di SQL Server atau BENCHMARK(1000000000,MD5(1) di MySQL).Jika ini dapat disuntikkan ke dalam kueri, maka lama waktu yang dibutuhkan server untuk merespons menunjukkan apakah kueri dapat disuntikkan atau tidak.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai musuh rentan terhadap injeksi SQL, coba masukkan nilai yang ditentukan pada langkah sebelumnya. Jika tidak terjadi kesalahan, maka musuh mengetahui bahwa injeksi SQL berhasil.. - Gunakan browser web untuk menginjeksi input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.

569	Object Relational Mapping Injection	[Hipotesiskan kueri SQL dalam aplikasi]Hipotesis yang dihasilkan terkait kueri SQL dalam aplikasi. Misalnya, musuh dapat berhipotesis bahwa input mereka diteruskan langsung ke kueri yang terlihat seperti: SELECT * FROM orders WHERE ordernum = ___atauSELECT * FROM orders WHERE ordernum IN (___)orSELECT * FROM orders WHERE ordernum in (_____) ORDER BY _____ Tentu saja, ada banyak kemungkinan lain.. - Meneliti jenis kueri SQL dan menentukan yang mana yang dapat digunakan di berbagai tempat dalam sebuah aplikasi.	[Tentukan cara memasukkan informasi ke dalam kueri]Tentukan cara memasukkan informasi ke dalam kueri dari langkah sebelumnya sehingga injeksi tidak memengaruhi logikanya. Misalnya, berikut ini adalah kemungkinan suntikan untuk kueri tersebut: 5' OR 1=1; --dan5) ATAU 1=1; --andordernum DESC; --. - Tambahkan klausa ke kueri SQL sehingga logika kueri tidak berubah.. - Tambahkan penundaan ke kueri SQL jika server tidak memberikan pesan kesalahan yang jelas (mis. WAITFOR DELAY '0:0:10' di SQL Server atau BENCHMARK(1000000000,MD5(1) di MySQL).Jika ini dapat disuntikkan ke dalam kueri, maka lama waktu yang dibutuhkan server untuk merespons menunjukkan apakah kueri dapat disuntikkan atau tidak.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai musuh rentan terhadap injeksi SQL, coba masukkan nilai yang ditentukan pada langkah sebelumnya. Jika tidak terjadi kesalahan, maka musuh mengetahui bahwa injeksi SQL berhasil.. - Gunakan browser web untuk menginjeksi input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.
569	SQL Injection through SOAP Parameter Tampering	[Hipotesiskan kueri SQL dalam aplikasi]Hipotesis yang dihasilkan terkait kueri SQL dalam aplikasi. Misalnya, musuh dapat berhipotesis bahwa input mereka diteruskan langsung ke kueri yang terlihat seperti: SELECT * FROM orders WHERE ordernum = ___atauSELECT * FROM orders WHERE ordernum IN (___)orSELECT * FROM orders WHERE ordernum in (_____) ORDER BY _____ Tentu saja, ada banyak kemungkinan lain.. - Meneliti jenis kueri SQL dan menentukan yang mana yang dapat digunakan di berbagai tempat dalam sebuah aplikasi.	[Tentukan cara memasukkan informasi ke dalam kueri]Tentukan cara memasukkan informasi ke dalam kueri dari langkah sebelumnya sehingga injeksi tidak memengaruhi logikanya. Misalnya, berikut ini adalah kemungkinan suntikan untuk kueri tersebut: 5' OR 1=1; --dan5) ATAU 1=1; --andordernum DESC; --. - Tambahkan klausa ke kueri SQL sehingga logika kueri tidak berubah.. - Tambahkan penundaan ke kueri SQL jika server tidak memberikan pesan kesalahan yang jelas (mis. WAITFOR DELAY '0:0:10' di SQL Server atau BENCHMARK(1000000000,MD5(1) di MySQL).Jika ini dapat disuntikkan ke dalam kueri, maka lama waktu yang dibutuhkan server untuk merespons menunjukkan apakah kueri dapat disuntikkan atau tidak.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai musuh rentan terhadap injeksi SQL, coba masukkan nilai yang ditentukan pada langkah sebelumnya. Jika tidak terjadi kesalahan, maka musuh mengetahui bahwa injeksi SQL berhasil.. - Gunakan browser web untuk menginjeksi input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.

569	SQL Injection	[Hipotesiskan kueri SQL dalam aplikasi]Hipotesis yang dihasilkan terkait kueri SQL dalam aplikasi. Misalnya, musuh dapat berhipotesis bahwa input mereka diteruskan langsung ke kueri yang terlihat seperti: SELECT * FROM orders WHERE ordernum = ___atauSELECT * FROM orders WHERE ordernum IN (___)orSELECT * FROM orders WHERE ordernum in (_____) ORDER BY _____ Tentu saja, ada banyak kemungkinan lain.. - Meneliti jenis kueri SQL dan menentukan yang mana yang dapat digunakan di berbagai tempat dalam sebuah aplikasi.	[Tentukan cara memasukkan informasi ke dalam kueri]Tentukan cara memasukkan informasi ke dalam kueri dari langkah sebelumnya sehingga injeksi tidak memengaruhi logikanya. Misalnya, berikut ini adalah kemungkinan suntikan untuk kueri tersebut: 5' OR 1=1; --dan5) ATAU 1=1; --andordernum DESC; --. - Tambahkan klausa ke kueri SQL sehingga logika kueri tidak berubah.. - Tambahkan penundaan ke kueri SQL jika server tidak memberikan pesan kesalahan yang jelas (mis. WAITFOR DELAY '0:0:10' di SQL Server atau BENCHMARK(1000000000,MD5(1) di MySQL).Jika ini dapat disuntikkan ke dalam kueri, maka lama waktu yang dibutuhkan server untuk merespons menunjukkan apakah kueri dapat disuntikkan atau tidak.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai musuh rentan terhadap injeksi SQL, coba masukkan nilai yang ditentukan pada langkah sebelumnya. Jika tidak terjadi kesalahan, maka musuh mengetahui bahwa injeksi SQL berhasil.. - Gunakan browser web untuk menginjeksi input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.
569	Expanding Control over the Operating System from the Database	[Hipotesiskan kueri SQL dalam aplikasi]Hipotesis yang dihasilkan terkait kueri SQL dalam aplikasi. Misalnya, musuh dapat berhipotesis bahwa input mereka diteruskan langsung ke kueri yang terlihat seperti: SELECT * FROM orders WHERE ordernum = ___atauSELECT * FROM orders WHERE ordernum IN (___)orSELECT * FROM orders WHERE ordernum in (_____) ORDER BY _____ Tentu saja, ada banyak kemungkinan lain.. - Meneliti jenis kueri SQL dan menentukan yang mana yang dapat digunakan di berbagai tempat dalam sebuah aplikasi.	[Tentukan cara memasukkan informasi ke dalam kueri]Tentukan cara memasukkan informasi ke dalam kueri dari langkah sebelumnya sehingga injeksi tidak memengaruhi logikanya. Misalnya, berikut ini adalah kemungkinan suntikan untuk kueri tersebut: 5' OR 1=1; --dan5) ATAU 1=1; --andordernum DESC; --. - Tambahkan klausa ke kueri SQL sehingga logika kueri tidak berubah.. - Tambahkan penundaan ke kueri SQL jika server tidak memberikan pesan kesalahan yang jelas (mis. WAITFOR DELAY '0:0:10' di SQL Server atau BENCHMARK(1000000000,MD5(1) di MySQL).Jika ini dapat disuntikkan ke dalam kueri, maka lama waktu yang dibutuhkan server untuk merespons menunjukkan apakah kueri dapat disuntikkan atau tidak.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai musuh rentan terhadap injeksi SQL, coba masukkan nilai yang ditentukan pada langkah sebelumnya. Jika tidak terjadi kesalahan, maka musuh mengetahui bahwa injeksi SQL berhasil.. - Gunakan browser web untuk menginjeksi input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.

569	Command Line Execution through SQL Injection	[Hipotesiskan kueri SQL dalam aplikasi]Hipotesis yang dihasilkan terkait kueri SQL dalam aplikasi. Misalnya, musuh dapat berhipotesis bahwa input mereka diteruskan langsung ke kueri yang terlihat seperti: SELECT * FROM orders WHERE ordernum = ___atauSELECT * FROM orders WHERE ordernum IN (___)orSELECT * FROM orders WHERE ordernum in (_____) ORDER BY _____ Tentu saja, ada banyak kemungkinan lain.. - Meneliti jenis kueri SQL dan menentukan yang mana yang dapat digunakan di berbagai tempat dalam sebuah aplikasi.	[Tentukan cara memasukkan informasi ke dalam kueri]Tentukan cara memasukkan informasi ke dalam kueri dari langkah sebelumnya sehingga injeksi tidak memengaruhi logikanya. Misalnya, berikut ini adalah kemungkinan suntikan untuk kueri tersebut: 5' OR 1=1; --dan5) ATAU 1=1; --andordernum DESC; --. - Tambahkan klausa ke kueri SQL sehingga logika kueri tidak berubah.. - Tambahkan penundaan ke kueri SQL jika server tidak memberikan pesan kesalahan yang jelas (mis. WAITFOR DELAY '0:0:10' di SQL Server atau BENCHMARK(1000000000,MD5(1) di MySQL).Jika ini dapat disuntikkan ke dalam kueri, maka lama waktu yang dibutuhkan server untuk merespons menunjukkan apakah kueri dapat disuntikkan atau tidak.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai musuh rentan terhadap injeksi SQL, coba masukkan nilai yang ditentukan pada langkah sebelumnya. Jika tidak terjadi kesalahan, maka musuh mengetahui bahwa injeksi SQL berhasil.. - Gunakan browser web untuk menginjeksi input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.
12	Blind SQL Injection	[Tentukan Kerangka Kegigihan yang Digunakan] Penyerang mencoba menentukan kerangka kerja ketekunan apa yang digunakan oleh aplikasi untuk memanfaatkan kelemahan dalam kode lapisan akses data yang dihasilkan atau kelemahan dengan cara lapisan akses data mungkin telah digunakan oleh pengembang .. - Penyerang memberikan masukan ke aplikasi dalam upaya untuk menginduksi layar kesalahan yang mengungkapkan jejak tumpukan yang memberikan indikasi lapisan akses data otomatis yang digunakan. Atau penyerang mungkin hanya membuat beberapa tebakan dan berasumsi, misalnya, bahwa Hibernate digunakan dan mencoba membuat serangan dari sana.	[Penyelidikan untuk kerentanan Injeksi ORM] Penyerang menyuntikkan sintaks ORM ke input data yang dapat dikontrol pengguna dari aplikasi untuk menentukan apakah mungkin memodifikasi struktur kueri data dan konten.	[Lakukan Injeksi SQL melalui lapisan akses data yang dihasilkan] Penyerang melanjutkan untuk mengeksploitasi kelemahan dalam metode akses data yang dihasilkan yang tidak memisahkan bidang kontrol dengan benar dari rencana data, atau berpotensi cara tertentu di mana pengembang mungkin telah menyalahgunakan kode yang dihasilkan , untuk memodifikasi struktur kueri SQL yang dieksekusi dan/atau menyuntikkan kueri SQL yang sama sekali baru.. - Penyerang menggunakan teknik injeksi SQL normal dan menyesuaikannya untuk mencerminkan jenis kerangka pembuatan lapisan akses data yang digunakan oleh aplikasi.

12	Object Relational Mapping Injection	[Tentukan Kerangka Kegigihan yang Digunakan] Penyerang mencoba menentukan kerangka kerja ketekunan apa yang digunakan oleh aplikasi untuk memanfaatkan kelemahan dalam kode lapisan akses data yang dihasilkan atau kelemahan dengan cara lapisan akses data mungkin telah digunakan oleh pengembang .. - Penyerang memberikan masukan ke aplikasi dalam upaya untuk menginduksi layar kesalahan yang mengungkapkan jejak tumpukan yang memberikan indikasi lapisan akses data otomatis yang digunakan. Atau penyerang mungkin hanya membuat beberapa tebakan dan berasumsi, misalnya, bahwa Hibernate digunakan dan mencoba membuat serangan dari sana.	[Penyelidikan untuk kerentanan Injeksi ORM] Penyerang menyuntikkan sintaks ORM ke input data yang dapat dikontrol pengguna dari aplikasi untuk menentukan apakah mungkin memodifikasi struktur kueri data dan konten.	[Lakukan Injeksi SQL melalui lapisan akses data yang dihasilkan] Penyerang melanjutkan untuk mengeksploitasi kelemahan dalam metode akses data yang dihasilkan yang tidak memisahkan bidang kontrol dengan benar dari rencana data, atau berpotensi cara tertentu di mana pengembang mungkin telah menyalahgunakan kode yang dihasilkan , untuk memodifikasi struktur kueri SQL yang dieksekusi dan/atau menyuntikkan kueri SQL yang sama sekali baru.. - Penyerang menggunakan teknik injeksi SQL normal dan menyesuaikannya untuk mencerminkan jenis kerangka pembuatan lapisan akses data yang digunakan oleh aplikasi.
12	SQL Injection through SOAP Parameter Tampering	[Tentukan Kerangka Kegigihan yang Digunakan] Penyerang mencoba menentukan kerangka kerja ketekunan apa yang digunakan oleh aplikasi untuk memanfaatkan kelemahan dalam kode lapisan akses data yang dihasilkan atau kelemahan dengan cara lapisan akses data mungkin telah digunakan oleh pengembang .. - Penyerang memberikan masukan ke aplikasi dalam upaya untuk menginduksi layar kesalahan yang mengungkapkan jejak tumpukan yang memberikan indikasi lapisan akses data otomatis yang digunakan. Atau penyerang mungkin hanya membuat beberapa tebakan dan berasumsi, misalnya, bahwa Hibernate digunakan dan mencoba membuat serangan dari sana.	[Penyelidikan untuk kerentanan Injeksi ORM] Penyerang menyuntikkan sintaks ORM ke input data yang dapat dikontrol pengguna dari aplikasi untuk menentukan apakah mungkin memodifikasi struktur kueri data dan konten.	[Lakukan Injeksi SQL melalui lapisan akses data yang dihasilkan] Penyerang melanjutkan untuk mengeksploitasi kelemahan dalam metode akses data yang dihasilkan yang tidak memisahkan bidang kontrol dengan benar dari rencana data, atau berpotensi cara tertentu di mana pengembang mungkin telah menyalahgunakan kode yang dihasilkan , untuk memodifikasi struktur kueri SQL yang dieksekusi dan/atau menyuntikkan kueri SQL yang sama sekali baru.. - Penyerang menggunakan teknik injeksi SQL normal dan menyesuaikannya untuk mencerminkan jenis kerangka pembuatan lapisan akses data yang digunakan oleh aplikasi.

12	SQL Injection	[Tentukan Kerangka Kegigihan yang Digunakan] Penyerang mencoba menentukan kerangka kerja ketekunan apa yang digunakan oleh aplikasi untuk memanfaatkan kelemahan dalam kode lapisan akses data yang dihasilkan atau kelemahan dengan cara lapisan akses data mungkin telah digunakan oleh pengembang .. - Penyerang memberikan masukan ke aplikasi dalam upaya untuk menginduksi layar kesalahan yang mengungkapkan jejak tumpukan yang memberikan indikasi lapisan akses data otomatis yang digunakan. Atau penyerang mungkin hanya membuat beberapa tebakan dan berasumsi, misalnya, bahwa Hibernate digunakan dan mencoba membuat serangan dari sana.	[Penyelidikan untuk kerentanan Injeksi ORM] Penyerang menyuntikkan sintaks ORM ke input data yang dapat dikontrol pengguna dari aplikasi untuk menentukan apakah mungkin memodifikasi struktur kueri data dan konten.	[Lakukan Injeksi SQL melalui lapisan akses data yang dihasilkan] Penyerang melanjutkan untuk mengeksploitasi kelemahan dalam metode akses data yang dihasilkan yang tidak memisahkan bidang kontrol dengan benar dari rencana data, atau berpotensi cara tertentu di mana pengembang mungkin telah menyalahgunakan kode yang dihasilkan , untuk memodifikasi struktur kueri SQL yang dieksekusi dan/atau menyuntikkan kueri SQL yang sama sekali baru.. - Penyerang menggunakan teknik injeksi SQL normal dan menyesuaikannya untuk mencerminkan jenis kerangka pembuatan lapisan akses data yang digunakan oleh aplikasi.
12	Expanding Control over the Operating System from the Database	[Tentukan Kerangka Kegigihan yang Digunakan] Penyerang mencoba menentukan kerangka kerja ketekunan apa yang digunakan oleh aplikasi untuk memanfaatkan kelemahan dalam kode lapisan akses data yang dihasilkan atau kelemahan dengan cara lapisan akses data mungkin telah digunakan oleh pengembang .. - Penyerang memberikan masukan ke aplikasi dalam upaya untuk menginduksi layar kesalahan yang mengungkapkan jejak tumpukan yang memberikan indikasi lapisan akses data otomatis yang digunakan. Atau penyerang mungkin hanya membuat beberapa tebakan dan berasumsi, misalnya, bahwa Hibernate digunakan dan mencoba membuat serangan dari sana.	[Penyelidikan untuk kerentanan Injeksi ORM] Penyerang menyuntikkan sintaks ORM ke input data yang dapat dikontrol pengguna dari aplikasi untuk menentukan apakah mungkin memodifikasi struktur kueri data dan konten.	[Lakukan Injeksi SQL melalui lapisan akses data yang dihasilkan] Penyerang melanjutkan untuk mengeksploitasi kelemahan dalam metode akses data yang dihasilkan yang tidak memisahkan bidang kontrol dengan benar dari rencana data, atau berpotensi cara tertentu di mana pengembang mungkin telah menyalahgunakan kode yang dihasilkan , untuk memodifikasi struktur kueri SQL yang dieksekusi dan/atau menyuntikkan kueri SQL yang sama sekali baru.. - Penyerang menggunakan teknik injeksi SQL normal dan menyesuaikannya untuk mencerminkan jenis kerangka pembuatan lapisan akses data yang digunakan oleh aplikasi.

12	Command Line Execution through SQL Injection	[Tentukan Kerangka Kegigihan yang Digunakan] Penyerang mencoba menentukan kerangka kerja ketekunan apa yang digunakan oleh aplikasi untuk memanfaatkan kelemahan dalam kode lapisan akses data yang dihasilkan atau kelemahan dengan cara lapisan akses data mungkin telah digunakan oleh pengembang .. - Penyerang memberikan masukan ke aplikasi dalam upaya untuk menginduksi layar kesalahan yang mengungkapkan jejak tumpukan yang memberikan indikasi lapisan akses data otomatis yang digunakan. Atau penyerang mungkin hanya membuat beberapa tebakan dan berasumsi, misalnya, bahwa Hibernate digunakan dan mencoba membuat serangan dari sana.	[Penyelidikan untuk kerentanan Injeksi ORM] Penyerang menyuntikkan sintaks ORM ke input data yang dapat dikontrol pengguna dari aplikasi untuk menentukan apakah mungkin memodifikasi struktur kueri data dan konten.	[Lakukan Injeksi SQL melalui lapisan akses data yang dihasilkan] Penyerang melanjutkan untuk mengeksploitasi kelemahan dalam metode akses data yang dihasilkan yang tidak memisahkan bidang kontrol dengan benar dari rencana data, atau berpotensi cara tertentu di mana pengembang mungkin telah menyalahgunakan kode yang dihasilkan , untuk memodifikasi struktur kueri SQL yang dieksekusi dan/atau menyuntikkan kueri SQL yang sama sekali baru.. - Penyerang menggunakan teknik injeksi SQL normal dan menyesuaikannya untuk mencerminkan jenis kerangka pembuatan lapisan akses data yang digunakan oleh aplikasi.
14	Blind SQL Injection	[Detect Incorrect SOAP Parameter Handling] Penyerang merusak parameter pesan SOAP dan mencari indikasi bahwa gangguan tersebut menyebabkan perubahan perilaku aplikasi yang ditargetkan. - Penyerang merusak parameter pesan SOAP dengan menginjeksi beberapa karakter khusus seperti single tanda kutip, tanda kutip ganda, semi kolom, dll. Penyerang mengamati perilaku sistem.	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke dalam parameter SOAP rentan yang diidentifikasi selama fase Jelajahi untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Menyuntikkan SQL melalui Parameter SOAP] Penyerang menyuntikkan SQL melalui parameter SOAP yang diidentifikasi sebagai rentan selama fase Jelajahi untuk meluncurkan serangan injeksi SQL urutan pertama atau kedua.. - Penyerang melakukan serangan injeksi SQL melalui metode biasa yang memanfaatkan parameter SOAP sebagai injeksi vektor. Penyerang harus berhati-hati untuk tidak merusak parser XML di penyedia layanan yang dapat mencegah payload masuk ke kueri SQL. Penyerang juga dapat melihat WSDL untuk layanan web (jika tersedia) untuk lebih memahami apa yang diharapkan oleh penyedia layanan.

14	Object Relational Mapping Injection	[Detect Incorrect SOAP Parameter Handling] Penyerang merusak parameter pesan SOAP dan mencari indikasi bahwa gangguan tersebut menyebabkan perubahan perilaku aplikasi yang ditargetkan. - Penyerang merusak parameter pesan SOAP dengan menginjeksi beberapa karakter khusus seperti single tanda kutip, tanda kutip ganda, semi kolom, dll. Penyerang mengamati perilaku sistem.	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke dalam parameter SOAP rentan yang diidentifikasi selama fase Jelajahi untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Menyuntikkan SQL melalui Parameter SOAP] Penyerang menyuntikkan SQL melalui parameter SOAP yang diidentifikasi sebagai rentan selama fase Jelajahi untuk meluncurkan serangan injeksi SQL urutan pertama atau kedua.. - Penyerang melakukan serangan injeksi SQL melalui metode biasa yang memanfaatkan parameter SOAP sebagai injeksi vektor. Penyerang harus berhati-hati untuk tidak merusak parser XML di penyedia layanan yang dapat mencegah payload masuk ke kueri SQL. Penyerang juga dapat melihat WSDL untuk layanan web (jika tersedia) untuk lebih memahami apa yang diharapkan oleh penyedia layanan.
14	SQL Injection through SOAP Parameter Tampering	[Detect Incorrect SOAP Parameter Handling] Penyerang merusak parameter pesan SOAP dan mencari indikasi bahwa gangguan tersebut menyebabkan perubahan perilaku aplikasi yang ditargetkan. - Penyerang merusak parameter pesan SOAP dengan menginjeksi beberapa karakter khusus seperti single tanda kutip, tanda kutip ganda, semi kolom, dll. Penyerang mengamati perilaku sistem.	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke dalam parameter SOAP rentan yang diidentifikasi selama fase Jelajahi untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Menyuntikkan SQL melalui Parameter SOAP] Penyerang menyuntikkan SQL melalui parameter SOAP yang diidentifikasi sebagai rentan selama fase Jelajahi untuk meluncurkan serangan injeksi SQL urutan pertama atau kedua.. - Penyerang melakukan serangan injeksi SQL melalui metode biasa yang memanfaatkan parameter SOAP sebagai injeksi vektor. Penyerang harus berhati-hati untuk tidak merusak parser XML di penyedia layanan yang dapat mencegah payload masuk ke kueri SQL. Penyerang juga dapat melihat WSDL untuk layanan web (jika tersedia) untuk lebih memahami apa yang diharapkan oleh penyedia layanan.
14	SQL Injection	[Detect Incorrect SOAP Parameter Handling] Penyerang merusak parameter pesan SOAP dan mencari indikasi bahwa gangguan tersebut menyebabkan perubahan perilaku aplikasi yang ditargetkan. - Penyerang merusak parameter pesan SOAP dengan menginjeksi beberapa karakter khusus seperti single tanda kutip, tanda kutip ganda, semi kolom, dll. Penyerang mengamati perilaku sistem.	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke dalam parameter SOAP rentan yang diidentifikasi selama fase Jelajahi untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Menyuntikkan SQL melalui Parameter SOAP] Penyerang menyuntikkan SQL melalui parameter SOAP yang diidentifikasi sebagai rentan selama fase Jelajahi untuk meluncurkan serangan injeksi SQL urutan pertama atau kedua.. - Penyerang melakukan serangan injeksi SQL melalui metode biasa yang memanfaatkan parameter SOAP sebagai injeksi vektor. Penyerang harus berhati-hati untuk tidak merusak parser XML di penyedia layanan yang dapat mencegah payload masuk ke kueri SQL. Penyerang juga dapat melihat WSDL untuk layanan web (jika tersedia) untuk lebih memahami apa yang diharapkan oleh penyedia layanan.

14	Expanding Control over the Operating System from the Database	[Detect Incorrect SOAP Parameter Handling] Penyerang merusak parameter pesan SOAP dan mencari indikasi bahwa gangguan tersebut menyebabkan perubahan perilaku aplikasi yang ditargetkan. - Penyerang merusak parameter pesan SOAP dengan menginjeksi beberapa karakter khusus seperti single tanda kutip, tanda kutip ganda, semi kolom, dll. Penyerang mengamati perilaku sistem.	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke dalam parameter SOAP rentan yang diidentifikasi selama fase Jelajahi untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Menyuntikkan SQL melalui Parameter SOAP] Penyerang menyuntikkan SQL melalui parameter SOAP yang diidentifikasi sebagai rentan selama fase Jelajahi untuk meluncurkan serangan injeksi SQL urutan pertama atau kedua.. - Penyerang melakukan serangan injeksi SQL melalui metode biasa yang memanfaatkan parameter SOAP sebagai injeksi vektor. Penyerang harus berhati-hati untuk tidak merusak parser XML di penyedia layanan yang dapat mencegah payload masuk ke kueri SQL. Penyerang juga dapat melihat WSDL untuk layanan web (jika tersedia) untuk lebih memahami apa yang diharapkan oleh penyedia layanan.
14	Command Line Execution through SQL Injection	[Detect Incorrect SOAP Parameter Handling] Penyerang merusak parameter pesan SOAP dan mencari indikasi bahwa gangguan tersebut menyebabkan perubahan perilaku aplikasi yang ditargetkan. - Penyerang merusak parameter pesan SOAP dengan menginjeksi beberapa karakter khusus seperti single tanda kutip, tanda kutip ganda, semi kolom, dll. Penyerang mengamati perilaku sistem.	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke dalam parameter SOAP rentan yang diidentifikasi selama fase Jelajahi untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Menyuntikkan SQL melalui Parameter SOAP] Penyerang menyuntikkan SQL melalui parameter SOAP yang diidentifikasi sebagai rentan selama fase Jelajahi untuk meluncurkan serangan injeksi SQL urutan pertama atau kedua.. - Penyerang melakukan serangan injeksi SQL melalui metode biasa yang memanfaatkan parameter SOAP sebagai injeksi vektor. Penyerang harus berhati-hati untuk tidak merusak parser XML di penyedia layanan yang dapat mencegah payload masuk ke kueri SQL. Penyerang juga dapat melihat WSDL untuk layanan web (jika tersedia) untuk lebih memahami apa yang diharapkan oleh penyedia layanan.

543	Blind SQL Injection	[Aplikasi survei] Penyerang pertama-tama menginventarisasi fungsionalitas yang diekspos oleh aplikasi.. - Situs web laba-laba untuk semua tautan yang tersedia. - Mengendus komunikasi jaringan dengan aplikasi menggunakan utilitas seperti WireShark.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai penyerang rentan terhadap injeksi SQL, coba masukkan karakter yang memiliki arti khusus dalam SQL (seperti karakter tanda kutip tunggal, karakter tanda kutip ganda, dua tanda hubung, tanda kurung, dll.). Tujuannya adalah untuk membuat kueri SQL dengan sintaks yang tidak valid.. - Gunakan browser web untuk memasukkan input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.	[Eksperimen dengan kerentanan SQL Injection] Setelah menentukan bahwa input yang diberikan rentan terhadap SQL Injection, buat hipotesis seperti apa kueri yang mendasarinya. Coba tambahkan logika ke kueri secara berulang untuk mengekstrak informasi dari database, atau untuk mengubah atau menghapus informasi dalam database.. - Gunakan sumber daya publik seperti Lembar Cheat Injeksi SQL di http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/, dan coba pendekatan berbeda untuk menambahkan logika ke kueri SQL.. - Tambahkan logika ke kueri, dan gunakan pesan kesalahan terperinci dari server untuk men-debug kueri. Misalnya, jika menambahkan satu kutipan ke kueri menyebabkan pesan kesalahan, coba : ' OR 1=1; --, atau hal lain yang secara sintaksis akan menyelesaikan kueri yang dihipotesiskan. Perbaiki kueri secara berulang.. - Gunakan teknik Blind SQL Injection untuk mengekstrak informasi tentang skema basis data.. - Jika serangan penolakan layanan adalah tujuannya, coba susun kueri. Ini tidak bekerja pada semua platform (terutama, tidak bekerja pada Oracle atau MySQL). Contoh masukan untuk dicoba meliputi: '; DROP TABLE SYSOBJECT; -- dan '); DROP TABLE SYSOBJECT; --. Kueri khusus ini kemungkinan besar tidak akan berfungsi karena tabel SYSOBJECTS umumnya dilindungi.

543	Object Relational Mapping Injection	[Aplikasi survei] Penyerang pertama-tama menginventarisasi fungsionalitas yang diekspos oleh aplikasi.. - Situs web laba-laba untuk semua tautan yang tersedia. - Mengendus komunikasi jaringan dengan aplikasi menggunakan utilitas seperti WireShark.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai penyerang rentan terhadap injeksi SQL, coba masukkan karakter yang memiliki arti khusus dalam SQL (seperti karakter tanda kutip tunggal, karakter tanda kutip ganda, dua tanda hubung, tanda kurung, dll.). Tujuannya adalah untuk membuat kueri SQL dengan sintaks yang tidak valid.. - Gunakan browser web untuk memasukkan input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.	[Eksperimen dengan kerentanan SQL Injection] Setelah menentukan bahwa input yang diberikan rentan terhadap SQL Injection, buat hipotesis seperti apa kueri yang mendasarinya. Coba tambahkan logika ke kueri secara berulang untuk mengekstrak informasi dari database, atau untuk mengubah atau menghapus informasi dalam database.. - Gunakan sumber daya publik seperti Lembar Cheat Injeksi SQL di http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/, dan coba pendekatan berbeda untuk menambahkan logika ke kueri SQL.. - Tambahkan logika ke kueri, dan gunakan pesan kesalahan terperinci dari server untuk men-debug kueri. Misalnya, jika menambahkan satu kutipan ke kueri menyebabkan pesan kesalahan, coba : ' OR 1=1; --, atau hal lain yang secara sintaksis akan menyelesaikan kueri yang dihipotesiskan. Perbaiki kueri secara berulang.. - Gunakan teknik Blind SQL Injection untuk mengekstrak informasi tentang skema basis data.. - Jika serangan penolakan layanan adalah tujuannya, coba susun kueri. Ini tidak bekerja pada semua platform (terutama, tidak bekerja pada Oracle atau MySQL). Contoh masukan untuk dicoba meliputi: '; DROP TABLE SYSOBJECT; -- dan '); DROP TABLE SYSOBJECT; --. Kueri khusus ini kemungkinan besar tidak akan berfungsi karena tabel SYSOBJECTS umumnya dilindungi.

543	SQL Injection through SOAP Parameter Tampering	[Aplikasi survei] Penyerang pertama-tama menginventarisasi fungsionalitas yang diekspos oleh aplikasi.. - Situs web laba-laba untuk semua tautan yang tersedia. - Mengendus komunikasi jaringan dengan aplikasi menggunakan utilitas seperti WireShark.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai penyerang rentan terhadap injeksi SQL, coba masukkan karakter yang memiliki arti khusus dalam SQL (seperti karakter tanda kutip tunggal, karakter tanda kutip ganda, dua tanda hubung, tanda kurung, dll.). Tujuannya adalah untuk membuat kueri SQL dengan sintaks yang tidak valid.. - Gunakan browser web untuk memasukkan input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.	[Eksperimen dengan kerentanan SQL Injection] Setelah menentukan bahwa input yang diberikan rentan terhadap SQL Injection, buat hipotesis seperti apa kueri yang mendasarinya. Coba tambahkan logika ke kueri secara berulang untuk mengekstrak informasi dari database, atau untuk mengubah atau menghapus informasi dalam database.. - Gunakan sumber daya publik seperti Lembar Cheat Injeksi SQL di http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/, dan coba pendekatan berbeda untuk menambahkan logika ke kueri SQL.. - Tambahkan logika ke kueri, dan gunakan pesan kesalahan terperinci dari server untuk men-debug kueri. Misalnya, jika menambahkan satu kutipan ke kueri menyebabkan pesan kesalahan, coba : ' OR 1=1; --, atau hal lain yang secara sintaksis akan menyelesaikan kueri yang dihipotesiskan. Perbaiki kueri secara berulang.. - Gunakan teknik Blind SQL Injection untuk mengekstrak informasi tentang skema basis data.. - Jika serangan penolakan layanan adalah tujuannya, coba susun kueri. Ini tidak bekerja pada semua platform (terutama, tidak bekerja pada Oracle atau MySQL). Contoh masukan untuk dicoba meliputi: '; DROP TABLE SYSOBJECT; -- dan '); DROP TABLE SYSOBJECT; --. Kueri khusus ini kemungkinan besar tidak akan berfungsi karena tabel SYSOBJECTS umumnya dilindungi.

543	SQL Injection	[Aplikasi survei] Penyerang pertama-tama menginventarisasi fungsionalitas yang diekspos oleh aplikasi.. - Situs web laba-laba untuk semua tautan yang tersedia. - Mengendus komunikasi jaringan dengan aplikasi menggunakan utilitas seperti WireShark.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai penyerang rentan terhadap injeksi SQL, coba masukkan karakter yang memiliki arti khusus dalam SQL (seperti karakter tanda kutip tunggal, karakter tanda kutip ganda, dua tanda hubung, tanda kurung, dll.). Tujuannya adalah untuk membuat kueri SQL dengan sintaks yang tidak valid.. - Gunakan browser web untuk memasukkan input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.	[Eksperimen dengan kerentanan SQL Injection] Setelah menentukan bahwa input yang diberikan rentan terhadap SQL Injection, buat hipotesis seperti apa kueri yang mendasarinya. Coba tambahkan logika ke kueri secara berulang untuk mengekstrak informasi dari database, atau untuk mengubah atau menghapus informasi dalam database.. - Gunakan sumber daya publik seperti Lembar Cheat Injeksi SQL di http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/, dan coba pendekatan berbeda untuk menambahkan logika ke kueri SQL.. - Tambahkan logika ke kueri, dan gunakan pesan kesalahan terperinci dari server untuk men-debug kueri. Misalnya, jika menambahkan satu kutipan ke kueri menyebabkan pesan kesalahan, coba : ' OR 1=1; --, atau hal lain yang secara sintaksis akan menyelesaikan kueri yang dihipotesiskan. Perbaiki kueri secara berulang.. - Gunakan teknik Blind SQL Injection untuk mengekstrak informasi tentang skema basis data.. - Jika serangan penolakan layanan adalah tujuannya, coba susun kueri. Ini tidak bekerja pada semua platform (terutama, tidak bekerja pada Oracle atau MySQL). Contoh masukan untuk dicoba meliputi: '; DROP TABLE SYSOBJECT; -- dan '); DROP TABLE SYSOBJECT; --. Kueri khusus ini kemungkinan besar tidak akan berfungsi karena tabel SYSOBJECTS umumnya dilindungi.

543	Expanding Control over the Operating System from the Database	[Aplikasi survei] Penyerang pertama-tama menginventarisasi fungsionalitas yang diekspos oleh aplikasi.. - Situs web laba-laba untuk semua tautan yang tersedia. - Mengendus komunikasi jaringan dengan aplikasi menggunakan utilitas seperti WireShark.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai penyerang rentan terhadap injeksi SQL, coba masukkan karakter yang memiliki arti khusus dalam SQL (seperti karakter tanda kutip tunggal, karakter tanda kutip ganda, dua tanda hubung, tanda kurung, dll.). Tujuannya adalah untuk membuat kueri SQL dengan sintaks yang tidak valid.. - Gunakan browser web untuk memasukkan input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.	[Eksperimen dengan kerentanan SQL Injection] Setelah menentukan bahwa input yang diberikan rentan terhadap SQL Injection, buat hipotesis seperti apa kueri yang mendasarinya. Coba tambahkan logika ke kueri secara berulang untuk mengekstrak informasi dari database, atau untuk mengubah atau menghapus informasi dalam database.. - Gunakan sumber daya publik seperti Lembar Cheat Injeksi SQL di http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/, dan coba pendekatan berbeda untuk menambahkan logika ke kueri SQL.. - Tambahkan logika ke kueri, dan gunakan pesan kesalahan terperinci dari server untuk men-debug kueri. Misalnya, jika menambahkan satu kutipan ke kueri menyebabkan pesan kesalahan, coba : ' OR 1=1; --, atau hal lain yang secara sintaksis akan menyelesaikan kueri yang dihipotesiskan. Perbaiki kueri secara berulang.. - Gunakan teknik Blind SQL Injection untuk mengekstrak informasi tentang skema basis data.. - Jika serangan penolakan layanan adalah tujuannya, coba susun kueri. Ini tidak bekerja pada semua platform (terutama, tidak bekerja pada Oracle atau MySQL). Contoh masukan untuk dicoba meliputi: '; DROP TABLE SYSOBJECT; -- dan '); DROP TABLE SYSOBJECT; --. Kueri khusus ini kemungkinan besar tidak akan berfungsi karena tabel SYSOBJECTS umumnya dilindungi.

543	Command Line Execution through SQL Injection	[Aplikasi survei] Penyerang pertama-tama menginventarisasi fungsionalitas yang diekspos oleh aplikasi.. - Situs web laba-laba untuk semua tautan yang tersedia. - Mengendus komunikasi jaringan dengan aplikasi menggunakan utilitas seperti WireShark.	[Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi] Tentukan input yang dapat dikontrol pengguna yang rentan terhadap injeksi. Untuk setiap input yang dapat dikontrol pengguna yang dicurigai penyerang rentan terhadap injeksi SQL, coba masukkan karakter yang memiliki arti khusus dalam SQL (seperti karakter tanda kutip tunggal, karakter tanda kutip ganda, dua tanda hubung, tanda kurung, dll.). Tujuannya adalah untuk membuat kueri SQL dengan sintaks yang tidak valid.. - Gunakan browser web untuk memasukkan input melalui bidang teks atau melalui parameter HTTP GET.. - Gunakan alat debugging aplikasi web seperti Tamper Data, TamperIE, WebScarab, dll. untuk mengubah parameter HTTP POST, bidang tersembunyi, bidang non-bentuk bebas, dll. - Gunakan alat injeksi paket tingkat jaringan seperti netcat untuk menyuntikkan input. - Gunakan klien yang dimodifikasi (dimodifikasi oleh rekayasa balik) untuk menyuntikkan input.	[Eksperimen dengan kerentanan SQL Injection] Setelah menentukan bahwa input yang diberikan rentan terhadap SQL Injection, buat hipotesis seperti apa kueri yang mendasarinya. Coba tambahkan logika ke kueri secara berulang untuk mengekstrak informasi dari database, atau untuk mengubah atau menghapus informasi dalam database.. - Gunakan sumber daya publik seperti Lembar Cheat Injeksi SQL di http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/, dan coba pendekatan berbeda untuk menambahkan logika ke kueri SQL.. - Tambahkan logika ke kueri, dan gunakan pesan kesalahan terperinci dari server untuk men-debug kueri. Misalnya, jika menambahkan satu kutipan ke kueri menyebabkan pesan kesalahan, coba : ' OR 1=1; --, atau hal lain yang secara sintaksis akan menyelesaikan kueri yang dihipotesiskan. Perbaiki kueri secara berulang.. - Gunakan teknik Blind SQL Injection untuk mengekstrak informasi tentang skema basis data.. - Jika serangan penolakan layanan adalah tujuannya, coba susun kueri. Ini tidak bekerja pada semua platform (terutama, tidak bekerja pada Oracle atau MySQL). Contoh masukan untuk dicoba meliputi: '; DROP TABLE SYSOBJECT; -- dan '); DROP TABLE SYSOBJECT; --. Kueri khusus ini kemungkinan besar tidak akan berfungsi karena tabel SYSOBJECTS umumnya dilindungi.
344	Blind SQL Injection	Musuh mengidentifikasi sistem manajemen basis data yang berjalan pada mesin yang ingin mereka kendalikan, atau pada jaringan yang ingin mereka lewati secara lateral.	Musuh melakukan langkah-langkah khas injeksi SQL dan menentukan apakah injeksi mungkin dilakukan.	Setelah Musuh menentukan bahwa injeksi SQL dimungkinkan, mereka harus memastikan bahwa persyaratan untuk serangan terpenuhi. Ini adalah pengguna sesi dengan hak istimewa tinggi dan dukungan kueri batch. Ini dilakukan dengan cara yang mirip untuk mengetahui apakah injeksi SQL dimungkinkan.

344	Object Relational Mapping Injection	Musuh mengidentifikasi sistem manajemen basis data yang berjalan pada mesin yang ingin mereka kendalikan, atau pada jaringan yang ingin mereka lewati secara lateral.	Musuh melakukan langkah-langkah khas injeksi SQL dan menentukan apakah injeksi mungkin dilakukan.	Setelah Musuh menentukan bahwa injeksi SQL dimungkinkan, mereka harus memastikan bahwa persyaratan untuk serangan terpenuhi. Ini adalah pengguna sesi dengan hak istimewa tinggi dan dukungan kueri batch. Ini dilakukan dengan cara yang mirip untuk mengetahui apakah injeksi SQL dimungkinkan.
344	SQL Injection through SOAP Parameter Tampering	Musuh mengidentifikasi sistem manajemen basis data yang berjalan pada mesin yang ingin mereka kendalikan, atau pada jaringan yang ingin mereka lewati secara lateral.	Musuh melakukan langkah-langkah khas injeksi SQL dan menentukan apakah injeksi mungkin dilakukan.	Setelah Musuh menentukan bahwa injeksi SQL dimungkinkan, mereka harus memastikan bahwa persyaratan untuk serangan terpenuhi. Ini adalah pengguna sesi dengan hak istimewa tinggi dan dukungan kueri batch. Ini dilakukan dengan cara yang mirip untuk mengetahui apakah injeksi SQL dimungkinkan.
344	SQL Injection	Musuh mengidentifikasi sistem manajemen basis data yang berjalan pada mesin yang ingin mereka kendalikan, atau pada jaringan yang ingin mereka lewati secara lateral.	Musuh melakukan langkah-langkah khas injeksi SQL dan menentukan apakah injeksi mungkin dilakukan.	Setelah Musuh menentukan bahwa injeksi SQL dimungkinkan, mereka harus memastikan bahwa persyaratan untuk serangan terpenuhi. Ini adalah pengguna sesi dengan hak istimewa tinggi dan dukungan kueri batch. Ini dilakukan dengan cara yang mirip untuk mengetahui apakah injeksi SQL dimungkinkan.
344	Expanding Control over the Operating System from the Database	Musuh mengidentifikasi sistem manajemen basis data yang berjalan pada mesin yang ingin mereka kendalikan, atau pada jaringan yang ingin mereka lewati secara lateral.	Musuh melakukan langkah-langkah khas injeksi SQL dan menentukan apakah injeksi mungkin dilakukan.	Setelah Musuh menentukan bahwa injeksi SQL dimungkinkan, mereka harus memastikan bahwa persyaratan untuk serangan terpenuhi. Ini adalah pengguna sesi dengan hak istimewa tinggi dan dukungan kueri batch. Ini dilakukan dengan cara yang mirip untuk mengetahui apakah injeksi SQL dimungkinkan.
344	Command Line Execution through SQL Injection	Musuh mengidentifikasi sistem manajemen basis data yang berjalan pada mesin yang ingin mereka kendalikan, atau pada jaringan yang ingin mereka lewati secara lateral.	Musuh melakukan langkah-langkah khas injeksi SQL dan menentukan apakah injeksi mungkin dilakukan.	Setelah Musuh menentukan bahwa injeksi SQL dimungkinkan, mereka harus memastikan bahwa persyaratan untuk serangan terpenuhi. Ini adalah pengguna sesi dengan hak istimewa tinggi dan dukungan kueri batch. Ini dilakukan dengan cara yang mirip untuk mengetahui apakah injeksi SQL dimungkinkan.
11	Blind SQL Injection	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke input data yang dapat dikontrol pengguna untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Mencapai eksekusi perintah sewenang-wenang melalui SQL Injection dengan direktif MSSQL_xp_cmdshell] Penyerang memanfaatkan serangan SQL Injection untuk menyuntikkan kode shell yang akan dieksekusi dengan memanfaatkan direktif xp_cmdshell.	[Menyuntikkan data berbahaya dalam database] Memanfaatkan injeksi SQL untuk menyuntikkan data dalam database yang nantinya dapat digunakan untuk mencapai injeksi perintah jika pernah digunakan sebagai argumen baris perintah

11	Object Relational Mapping Injection	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke input data yang dapat dikontrol pengguna untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Mencapai eksekusi perintah sewenang-wenang melalui SQL Injection dengan direktif MSSQL_xp_cmdshell] Penyerang memanfaatkan serangan SQL Injection untuk menyuntikkan kode shell yang akan dieksekusi dengan memanfaatkan direktif xp_cmdshell.	[Menyuntikkan data berbahaya dalam database] Memanfaatkan injeksi SQL untuk menyuntikkan data dalam database yang nantinya dapat digunakan untuk mencapai injeksi perintah jika pernah digunakan sebagai argumen baris perintah
11	SQL Injection through SOAP Parameter Tampering	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke input data yang dapat dikontrol pengguna untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Mencapai eksekusi perintah sewenang-wenang melalui SQL Injection dengan direktif MSSQL_xp_cmdshell] Penyerang memanfaatkan serangan SQL Injection untuk menyuntikkan kode shell yang akan dieksekusi dengan memanfaatkan direktif xp_cmdshell.	[Menyuntikkan data berbahaya dalam database] Memanfaatkan injeksi SQL untuk menyuntikkan data dalam database yang nantinya dapat digunakan untuk mencapai injeksi perintah jika pernah digunakan sebagai argumen baris perintah
11	SQL Injection	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke input data yang dapat dikontrol pengguna untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Mencapai eksekusi perintah sewenang-wenang melalui SQL Injection dengan direktif MSSQL_xp_cmdshell] Penyerang memanfaatkan serangan SQL Injection untuk menyuntikkan kode shell yang akan dieksekusi dengan memanfaatkan direktif xp_cmdshell.	[Menyuntikkan data berbahaya dalam database] Memanfaatkan injeksi SQL untuk menyuntikkan data dalam database yang nantinya dapat digunakan untuk mencapai injeksi perintah jika pernah digunakan sebagai argumen baris perintah
11	Expanding Control over the Operating System from the Database	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke input data yang dapat dikontrol pengguna untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Mencapai eksekusi perintah sewenang-wenang melalui SQL Injection dengan direktif MSSQL_xp_cmdshell] Penyerang memanfaatkan serangan SQL Injection untuk menyuntikkan kode shell yang akan dieksekusi dengan memanfaatkan direktif xp_cmdshell.	[Menyuntikkan data berbahaya dalam database] Memanfaatkan injeksi SQL untuk menyuntikkan data dalam database yang nantinya dapat digunakan untuk mencapai injeksi perintah jika pernah digunakan sebagai argumen baris perintah
11	Command Line Execution through SQL Injection	[Penyelidikan untuk kerentanan Injeksi SQL] Penyerang menyuntikkan sintaks SQL ke input data yang dapat dikontrol pengguna untuk mencari eksekusi sintaks SQL tanpa filter dalam kueri.	[Mencapai eksekusi perintah sewenang-wenang melalui SQL Injection dengan direktif MSSQL_xp_cmdshell] Penyerang memanfaatkan serangan SQL Injection untuk menyuntikkan kode shell yang akan dieksekusi dengan memanfaatkan direktif xp_cmdshell.	[Menyuntikkan data berbahaya dalam database] Memanfaatkan injeksi SQL untuk menyuntikkan data dalam database yang nantinya dapat digunakan untuk mencapai injeksi perintah jika pernah digunakan sebagai argumen baris perintah