LOGIQUE

Ancaman Serangan pada Web Application

PT. LOGIQUE DIGITAL INDONESIA © 2020

INTRO

Aplikasi web adalah hal yang banyak dikembangkan oleh suatu organisasi, teknologinya yang mampu mempermudah pekerjaan tentu sangat banyak digunakan oleh banyak pihak, baik itu perorangan, institusi atau organisasi. Applikasi web bisa menjadi pintu masuk ke suatu perushaan tentunya banyak informasi atau data penting yang bisa didapatkan. Hal ini tentu bisa menjadi sasaran “empuk” bagi pihak-pihak yang tidak bertanggung jawab untuk melakukan eksploitasi ke dalamnya. Banyak risiko ancaman serangan yang mungkin bisa terjadi pada aplikasi yang digunakan. OWASP (Open Web Securitiy Project) adalah sebuah organisasi internasional yang fokus terhadap pengembangan keamanan sistem informasi telah merilis dan mengklasifikasikan mengenai celah dan jenis ancaman serangan terhadap web application ke dalam OWASP TOP 10 Application Security Risk. *(sumber: https://owasp.org/www-project-top-ten/)*

A1 Injection

Injeksi adalah sebuah jenis serangan di mana *attacker* memanfaatkan celah pada aplikasi web dengan cara memasukkan perintah atau query berbahaya. Kerentanan injeksi sering terjadi terhadap web aplikasi yang menggunakan query SQL, LDAP, XPath, atau NoSQL, perintah OS, SMTP headers dan lainnya

Teknik serangan injeksi terkadang bisa menyebabkan risiko yang tinggi bagi sebuah sistem, karena dengan memanfaatkan kerentanan tersebut attacker dapat menyebabkan sebuah sistem pada suatu organisasi kehilangan data, data corrupt, denial of access atau bahkan attacker bisa melakukan *take over* suatu host.

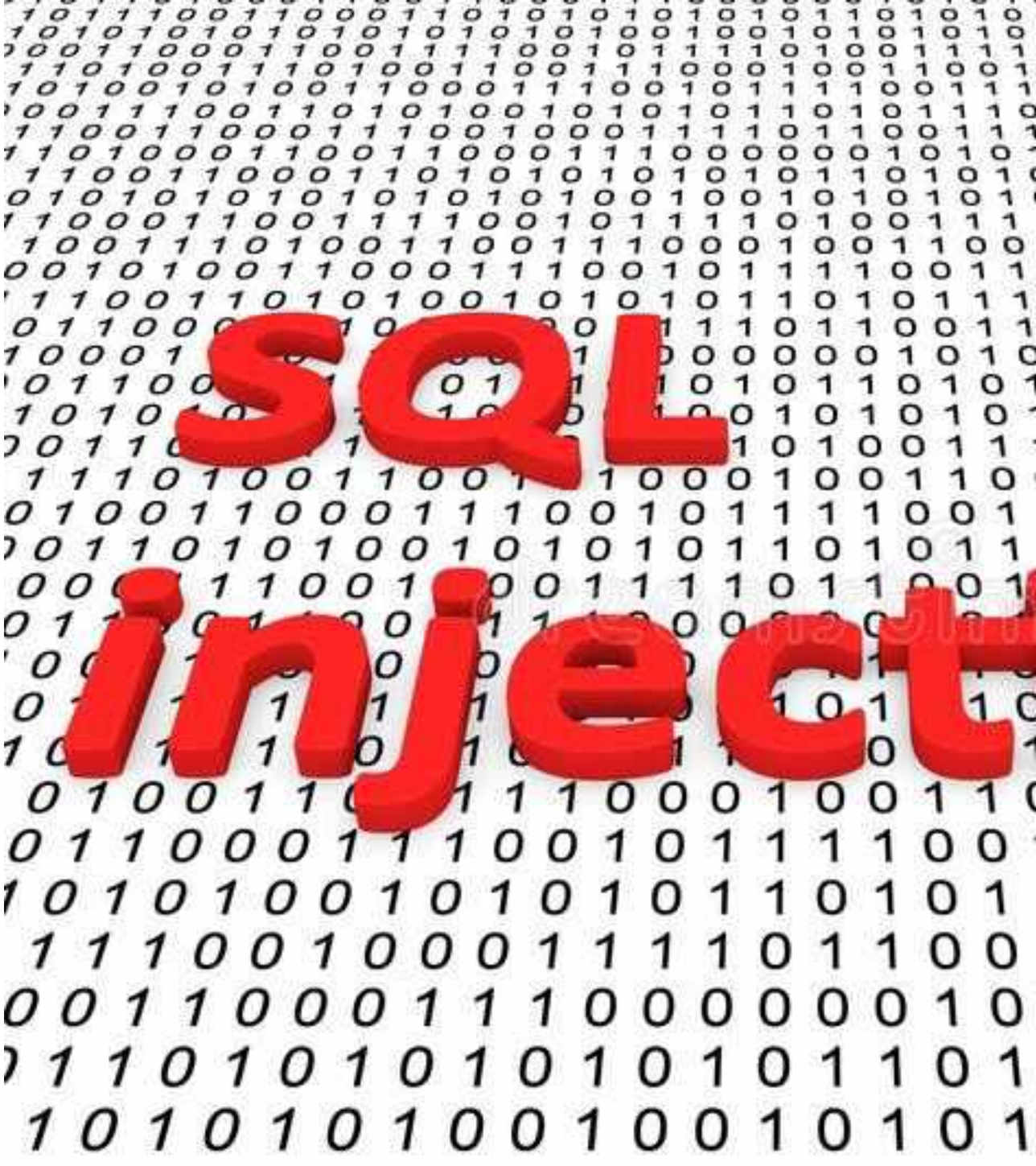
1 SQL injection

Teknik SQL injection adalah teknik yang cukup sering dipakai para attacker untuk masuk ke dalam sebuah web application, karena celah ini juga cukup banyak terdapat pada web application di internet. SQL injection digunakan ketika web application tidak melakukan validasi terhadap perintah-perintah SQL, attacker biasanya menginjeksikan *"malicious"* SQL query kedalam web browser address bar, form field, query, kolom pencarian dan lainnya.

Contoh Dampak yang mungkin terjadi

SQL injection sebagai salah satu ancaman yang cukup banyak terjadi dan menjadi *high risk threat* pada web application mengapa demikian? Karena dampak dari celah ini bisa dibilang cukup fatal, beberapa diantaranya ialah:

a. Login kedalam sistem tanpa valid credential
b. Ilegal query database sehingga attacker dapat melihat isi database
c. Memodifikasi isi database bahkan dapat melakukan drop database



2 Command Injection Attack

Contoh serangan yang menggunakan teknik injeksi berikutnya adalah Coomand Injection Attack, sebuah serangan yang memanfaatkan sebuah kecacatan pada sistem web application dengan cara melakukan injeksi perintah OS melalui HTTP *request* ke dalam aplikasi. OS command injection adalah teknik yang menggunakan antarmuka web untuk melakukan OS command di dalam web server.

Contoh Dampak Yang mungkin Terjadi

Selain SQL injection yang mempunyai dampak fatal pada sebuah web application, Command Injection Attack juga tak kalah fatal karena dapat berdampak pada web application beberapa contoh diantaranya ialah:

a. Attacker dapat mengakses direktori sensitif pada web server
b. Attacker dapat mengupload file berbahaya seperti virus, trojan dan lainnya ke dalam server
c. Attacker dapat melakukan enumerasi username dan password

3 LDAP Injection

Lightweight Directory Access Protocol (LDAP) adalah protokol internet yang digunakan untuk mengakses direktory misalnya saja digunakan untuk akses direktori alamat perusahaan, alamat email atau telepon. Kesalahan konfirgurasi terhadap LDAP dapat menimbulkan risiko celah keamanan. LDAP injection mirip dengan SQL injection di mana risiko ini akan terjadi apabila aplikasi gagal melakukan sanitasi terhadap inputan user.

Contoh Dampak Yang mungkin Terjadi
LDAP injection mempunyai dampak yang mungkin ditimbulkan terhadap sistem aplikasi web apps di antarnya ialah:
a. Bypass login
b. Information Disclosure
c. Privilege escalation

A2 Broken Authentication

web apps merupakan pintu masuk dan biasanya akan dilengkapi sebauh autentikasi dan manajemen sesi tentang siapa saja yang boleh memasukinya, *authentication* dan *session management* sering kali tidak diterapkan sebagaimana mestinya hal ini memungkinkan untuk attacker dapat menyesupi sandi, *keys, session tokens* atau mengeksploitasi kelemahan lainnya.

Attackers dapat melakukan eksplotasi terhadap kerentanan ini dengan berbagai cara misalnya menggunakan deteksi *broken authentication* dengan cara manal dan mengeksploitasinya mengganakan *automation tool* dengan *password list* dan d*ictionary attack*. Attacker juga bisa melancarkan aksinya jika terdapat celah-celah pada fitur *authentication* dan *session management* , web application seperti *session ID, logout, remember me, time outs, secret question* atau apapun itu yang membuat seolah-olah attacker ada pengguna yang mempunyai hak askses yang legal atau diijinkan.

Dampak yang mungkin terjadi

Attacker akan mencoba segala cara untuk mencoba merusak dan menerobos autentikasi pada sistem web apps, dampak yang bisa terjadi ketika attacker dapat mengkompromikan ini ialah:

a. Penyerang bisa mendapatkan akses ke suatu sistem;
entah itu user akun atau bahkan admin akun
b. Pencurian Identitas
c. Mendapatkan informasi sensitif

A3 Sensitive Data Exposure

Tak jarang sebuah web application akan menyimpan data-data penting yang sifatnya sensitif atau bisa dikatakan rahasia seperi data kartu kredit, KTP, dan info lainnya, banyak web appication yang tidak terlindungi secara penuh karena terkadang data-data atau informasi yang tersimpan di dalam database tidak dilengkapi dengan enkripsi yang kuat sehingga ketika terdapat attacker yang berhasil mencuri data sensitif attacker akan dengan mudah membca data tersebut, sehingga sebaiknya sebuah web application yang memiliki data- data sensitif harus melindungi datanya baik itu saat disimpan atau bahkan saay ditranmisikan.

Data enkripsi juga harus dipastikan menggunakan standar yang kuat karena meskipun demikian ketika data enkripsi tidak kuat attacker bisa saja masih bisa mengeksploitasinya, selain menggunakan enkripsi yang kuat developer juga harus memastikan meletakan *keys* enkripsi di tempat yang aman.

Dampak yang mungkin terjadi

Perlindungan terdapat data yang di dalam sebuah web application adalah hal yang harus dilakukan untuk itu deteksi dan pengujian untuk memastikan data yang ada di dalam website sudah dilindungi harus dilakuan, berikut adalah dampak yang mungkin terjadi apabila data tidak terlindungi dengan enkripsi:

a. Pencurian Identitas
b. Pencurian data sensitif
c. Bocornya credential login

A4 XML External Entities (XXE)

Beberapa web application menggunakan format XML, untuk menjembatani komunikasi antara server dan browser. Kerentanan XXE muncul karena struktur XML mengandung berbagai macam potensi berbahaya untuk bisa dieksploitasi. XXE adalah sebuah teknik di mana attacker bisa melakukan eksplotasi sebuah celah processor XML ketika sebuah website bisa mengupload file XML dan memodifikasinya sesuai keinginan attacker untuk mendapatkan informasi sensitif sebuah server. Hal ini terjadi karena sistem akan menerima secara langsung XML file tanpa divalidasi terlebih dahulu.

XXE attack bisa terjadi dengan banyak cara contoh diantaranya ialah sebagai berikut:

a. XXE attack dengan File Upload
beberapa web application terdapat sebuah fitur di mana user diperbolehkan untuk mengupload file ke dalam web server. Meskipun file upload terkadang adalah file seperti file DOCX dan format SVG, sebagai contoh ketika aplikasi mengijinkan upload file image berformat PNG atau JPEG namun terkadang prsosesing image librarynya support terhadap format SVG yang menggunakan XML, sehingga attacker bisa saja mengupload malicious SVG yang sudah dimodifikasi.

Contoh Dampak yang mungkin terjadi

Kerentanan XXE biasanya akan menimbulkan dampak yang cukup berbahaya diantaranya adalah:

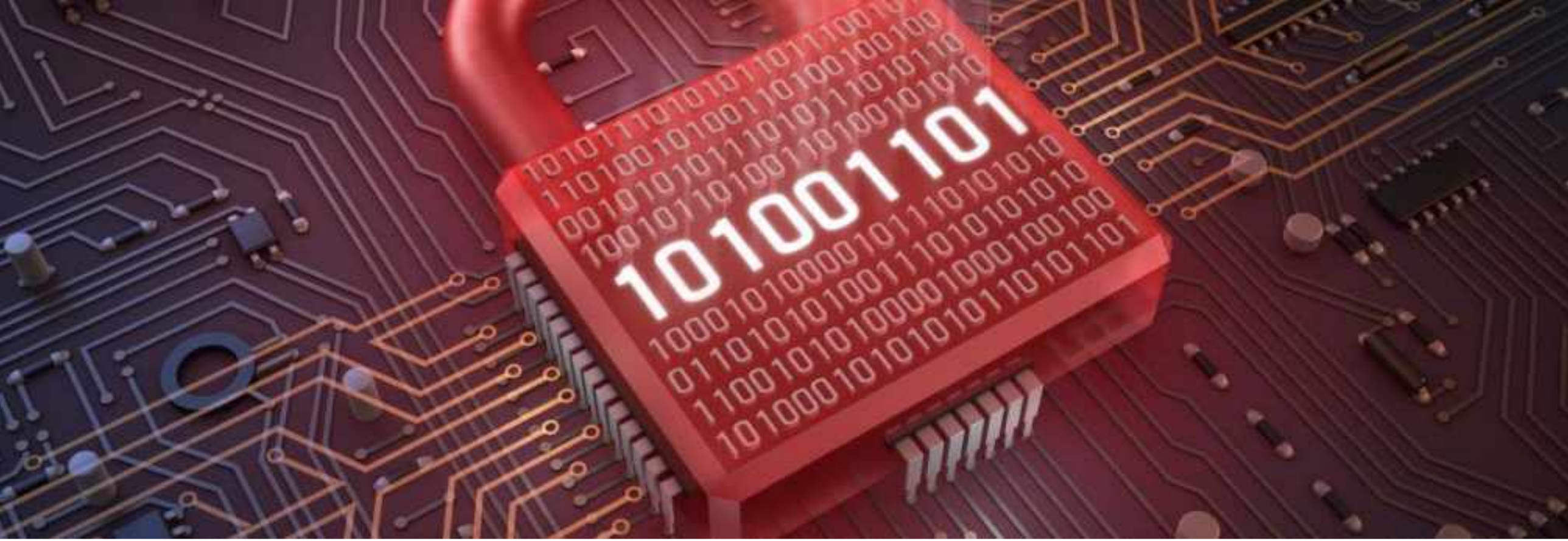
a. Attacker bisa meng-ekstrak data sensitif dari server
b. Scan internal system
c. Melakukan DoS
d. Remote request dari server

A5 Broken Access Control

Access Control adalah hal yang sering diterapkan pada sebuah web application, misalnya di dalam web application terdapat user role sebagai Admin di mana user role tersebut dapat melakuakn Create, Read, Update ataupun Delete terhadap suatu data pada web.
Broken Access Control adalah sebuah upaya yang dilakukan attacker untuk melewati atau mengeksploitasi Access Control tersebut.
Dengan adanya kerentanan terhadap Access Control pada sebuah application maka attacker bisa saja bertindak sebagai admin yang dapat melakukan perubahan data pada web application.
Broken Access control biasanya bisa dideteksi dengan manual testing, seperti bisa melakukan manual testing untuk melakukan request terhadap HTTP method seperti GET, PUT dan lainnya, bisa juga memanfaatkan manual testing terhadap controller, direct object reference dan lainnya.

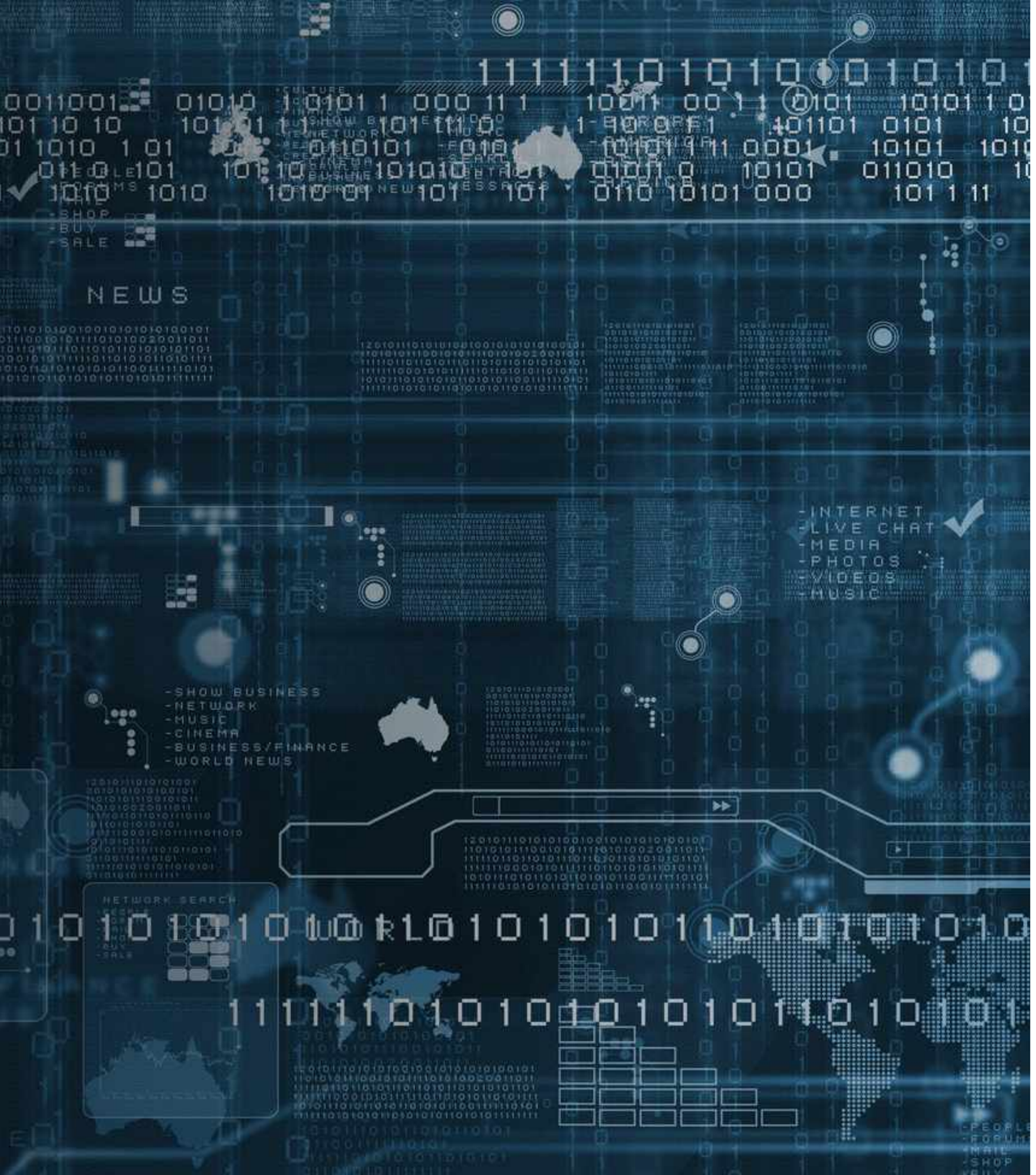
Contoh Dampak yang mungkin terjadi

Web application memiliki access control tertentu terhadap setiap user rolenya, namun ketika sebuah wep application memiliki kerentanan terhadap access controlnya akan memunculkan beberapa dampak yang mungkin terjadi diantaranya ialah attacker bisa melakukan perubahan data yang ada pada sistem dan berperilaku layaknya *legitimate user*.



A6 Security Misconfiguration

Attackers akan selalu berusaha untuk mengeksploitasi sebuah sistem untuk dapat mendapatkan akses ke dalamnya, sebuah web application system mempunyai banyak konfigurasi baik itu konfigurasi yang bersifat fungsional atau teknis. Kesalahan konfigurasi dapat berdampak dalam aspek security, sehingga memungkinkan attacker dapat mendapatkan akses yang seharusnya tidak diperbolehkan.
Security Misconfiguration pada web application bisa datang dari berbagai faktor, entah itu dari layanan jaringan, web server, application server, database, framework, custom code dan lainnya.



Berikut adalah contoh teknik serangan saat web application tidak menerapkan konfigurasi yang baik diantaranya ialah:

1 Unvalidated Input

Dalam sebuah web application terkadang terdapat sebuah fitur di mana user bisa melakukan input data melalui form atau field tertentu. Dalam beberapa kasus ternyata web application tidak melakukan konfigurasi terhadap sistemnya untuk memvalidasi inputan tersebut sehingga user dapat memasukkan input apapun ke dalam sistem tanpa validasi terlebih dahulu. Sehingga memungkinkan untuk attacker dapat melakukan aksi berbahaya dan dapat berdampak pada sebuah sistem web.
Unvalidated Input pada sebuah form bisa menyebabkan serangan yang serius misalnya serangan injection, XSS, buffer overflow dan lainnya.

2 Data Tampering

Web application tentunya terdapat komunikasi antara client dan server, dalam prakteknya dari sisi client akan mengirim data ke dalam server agar data tersebut diproses ke dalam server. Attacker akan memanfaatkan kondisi ini untuk melakukan serangan data tampering sebagai contoh menggunakan teknik serangan man in the midlle (MITM) dengan serangan ini attacker akan mencoba memodifikasi pertukaran data antara client dan server. Sehingga integritas data bisa saja berpengaruh.

Contoh Dampak Yang Mungkin Terjadi

Security Misconfiguration bisa berdampak terhadap banyak hal baik itu secara intgritas data bahkan sampai ke business logic suatu web application. Berikut beberapa contoh yang mungkin terjadi apabila web application tidak menerapkan configuration yang baik:

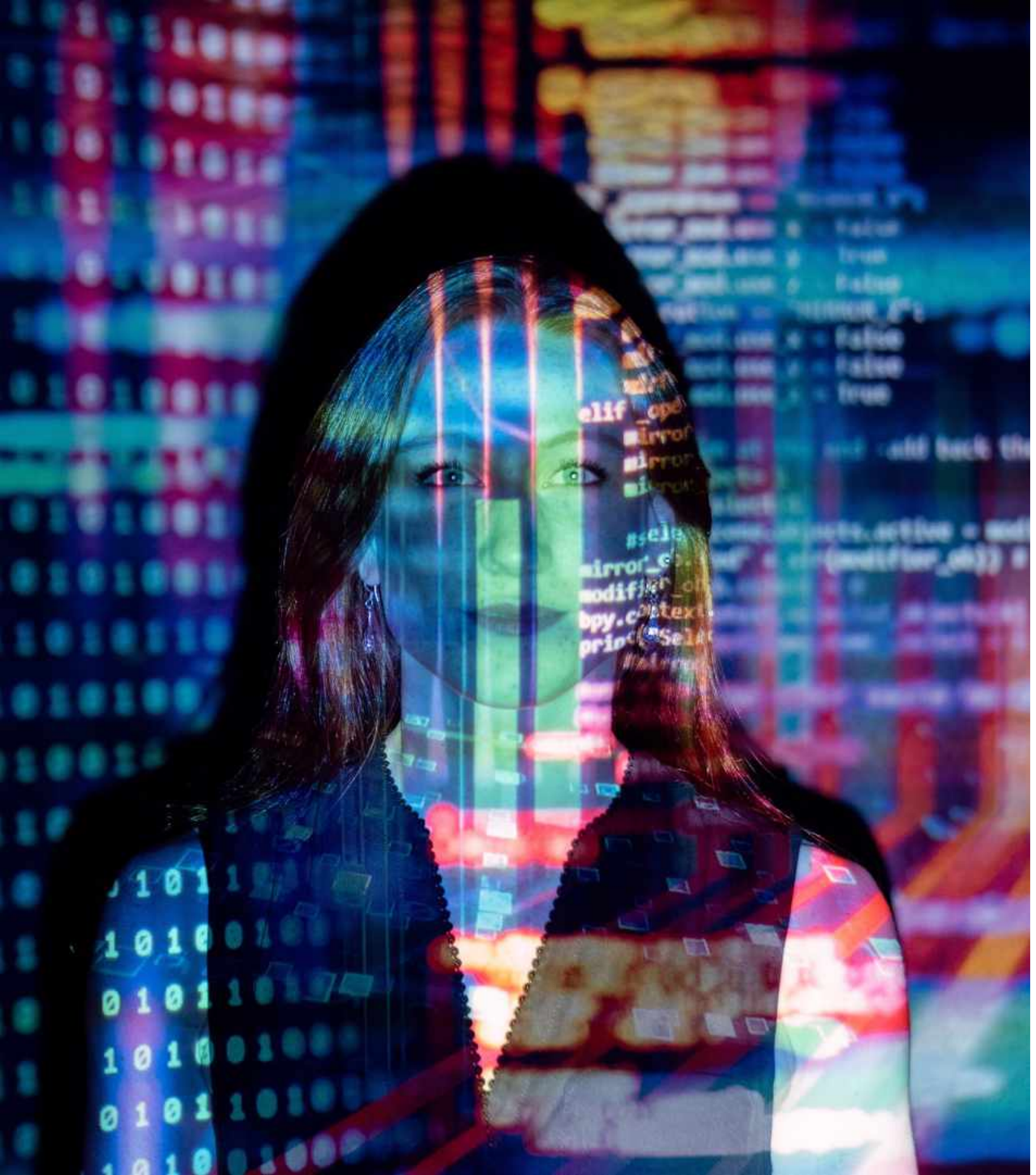
a. Pengaksesan Data sensitif
b. Pengubahan Data harga, ID user atau user credential lain
c. Attacker bisa mengubah data hanya melalui URL yang terdapat parameter tertentu

A7 Cross-Site Scripting

Cross Site Scripting (XSS) adalah sebuah *issue* yang cukup banyak terjadi dalam sebuah wen application, XSS adalah sebuah upaya dari attacker untuk mengeksploitasi kerentanan pada halaman web di mana halaman tersebut dapat tergenerate di sisi user lain. XSS attack dilakukan dengan cara melakukan injeksi script berbahaya yang ditargetkan kepada user lain.
Halaman web terdiri dari text dan HTML script, attacker bisa menyisipkan script yang tidak terpercaya ke dalamnya dan akan tergenerate kedalam halaman web yang terdampak dari script tersebut, ketika server maupun client tidak menyadari adanya script yang berbahaya tersebut maka risiko keamanan bisa terjadi dan biasaya client atau userlah menjadi korbanya.

XSS attack bisa dilakukan dengan banyak cara contohnya adalah melalui comment field pada blog posting yang rentan terhadap serangan tersebut, attacker dengan script berbaya bisa saja mengelabuhi pengunjung blog post lain untuk di arahkan ke situs berbahaya atau lainnya.

XSS biasanya bisa terdeksi menggunakan *automated test* memanfaatkan tool scanner atau sejenisnya, namun beberapa kasus manual testing juga perlu dilakukan untuk menemukan celah dari XSS tersebut.



Contoh Dampak yang mungkin terjadi

Sebagai issue yang sering terjadi pada web application, XSS tentu menjadi ancaman yang bisa dibilang cukup tinggi meski kadang sebenarnya XSS tidak berpengaruh secara langsung terhadap server, karena memang target XSS biasanya akan menargetkan ke pada user dari web application tersebut. Beberapa dampak yang mungkin terjadi ialah sebagai berikut:

a. Attacker dapat mencuri cookies user lain, di beberapa kasus attacker bisa juga mencuri cookies pada user role level admin.
b. Mengirim malware
c. Merusak UI karena script yang tak seharusnya bisa saja terender di halaman web

A8 Insecure Deserialization

Data dalam komputer biasanya akan tersimpan secara terstruktur, data terstruktur biasanya akan terepresentasi kedalam array, graph atau lainnya. Untuk mempermudah mentransmisikan data ke dalam jaringan maka biasanya data akan dilakukan serialization dan deserelization.

Insecure Deserialization adalah upaya dari attacker untuk menginject malicious code terhadap proses deserialization tersebut. Eksploit Insecure Deserialization adalah sesuatu yang cukup sulit, dalam praktiknya beberapa tools dapat melakukan deteksi terhadap kerentanan ini, namun bantuan manusia untuk menguji kerentanan ini akan lebih baik untuk memvalidasi hasil dari tool yang digunakan.

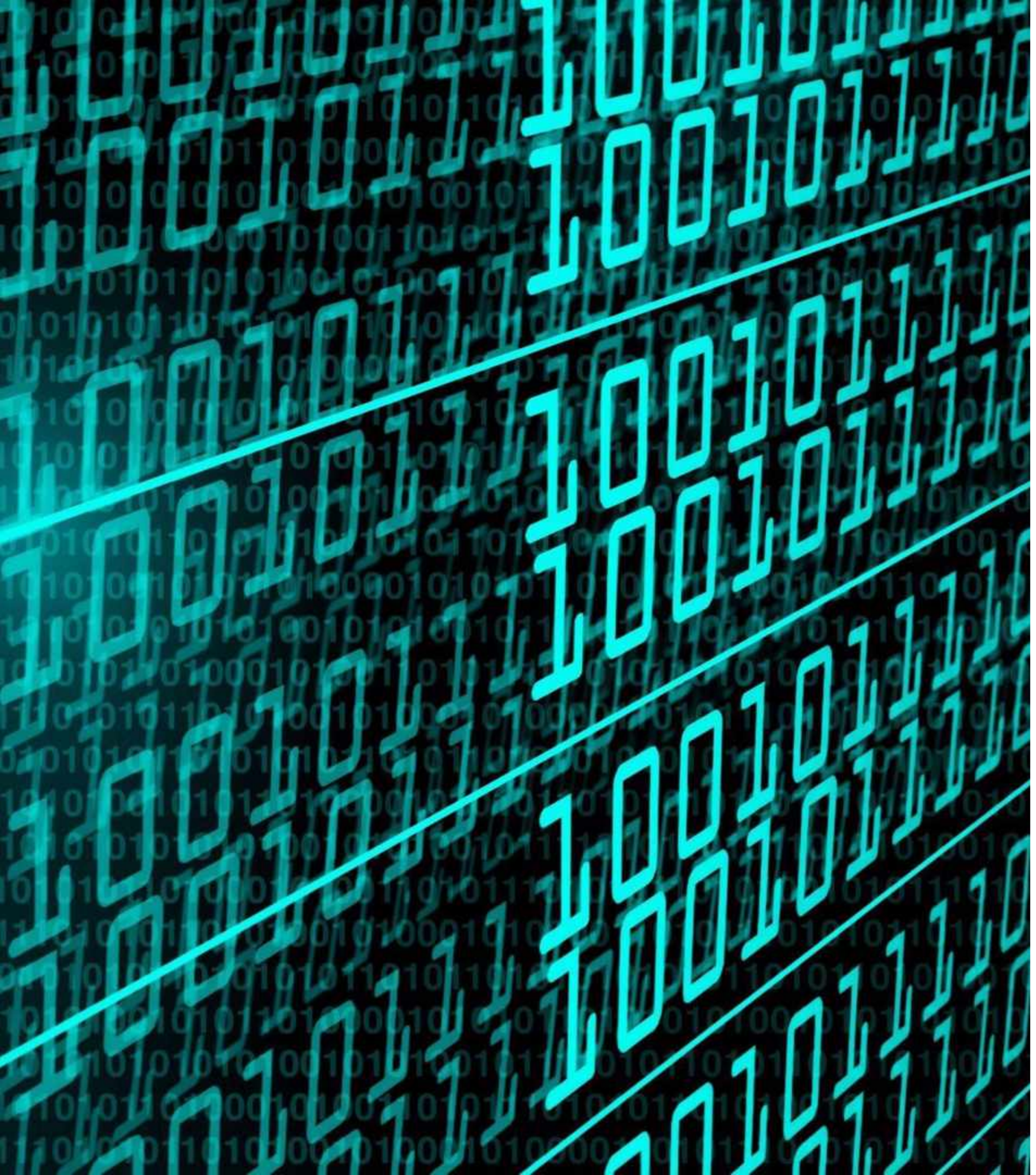
Contoh Dampak yang mungkin terjadi

Insecure Deserialization adalah hal yang mempunyai dampak fatal diantaranaya adalah RCE atau Remode Code Execution

A9 Using Components with Known Vulnerabilities

Web application bisa saja terdiri dari banyak library atau biasanya web application juga dikembangkan menggunakan suatu framework tertentu. Attacker akan melakukan identifikasi terhadap librabry atau framework yang dipakai dalam web application tersebut dan attaker akan mencoba mencari celah pada komponen tersebut berdasarkan celah yang telah diketahui sebelumnya. Sebagai contoh sebuah web application menggunakan framework versi tertentu kebetulan framework tersebut memiliki kerentanan dan sudah dituliskan oleh security analisis lain, maka attacker akan bisa memanfaatkan celah tersebut untuk mengeksploitasi sebuag web apps.

Untuk itu memonitoring versi komponen dan selalu update mengenai security informasi perlu terus dilakukan, agar issue-issue pada komponen web application (baik library, framework dll) yang
digunakan dapat selalu diupdate dan dipatch ketika diketahui adanya sebuah kerentanang



Contoh Dampak yang mungkin terjadi

Jika kita tidak aware untuk melakukan monitoring terhadap komponen-komponen yang ada dalam web application kita atau kita tidak melakukan update mengenai issue-issue yang berkaitan tentang bug yang pernah ditemukan dalam suatu kompenen tersebut maka bisa saja attacker akan memanfaatkan bug tersebut untuk melakukan eksploitasi terhadap web application tersebut.

A10 Insufficient Logging & Monitoring

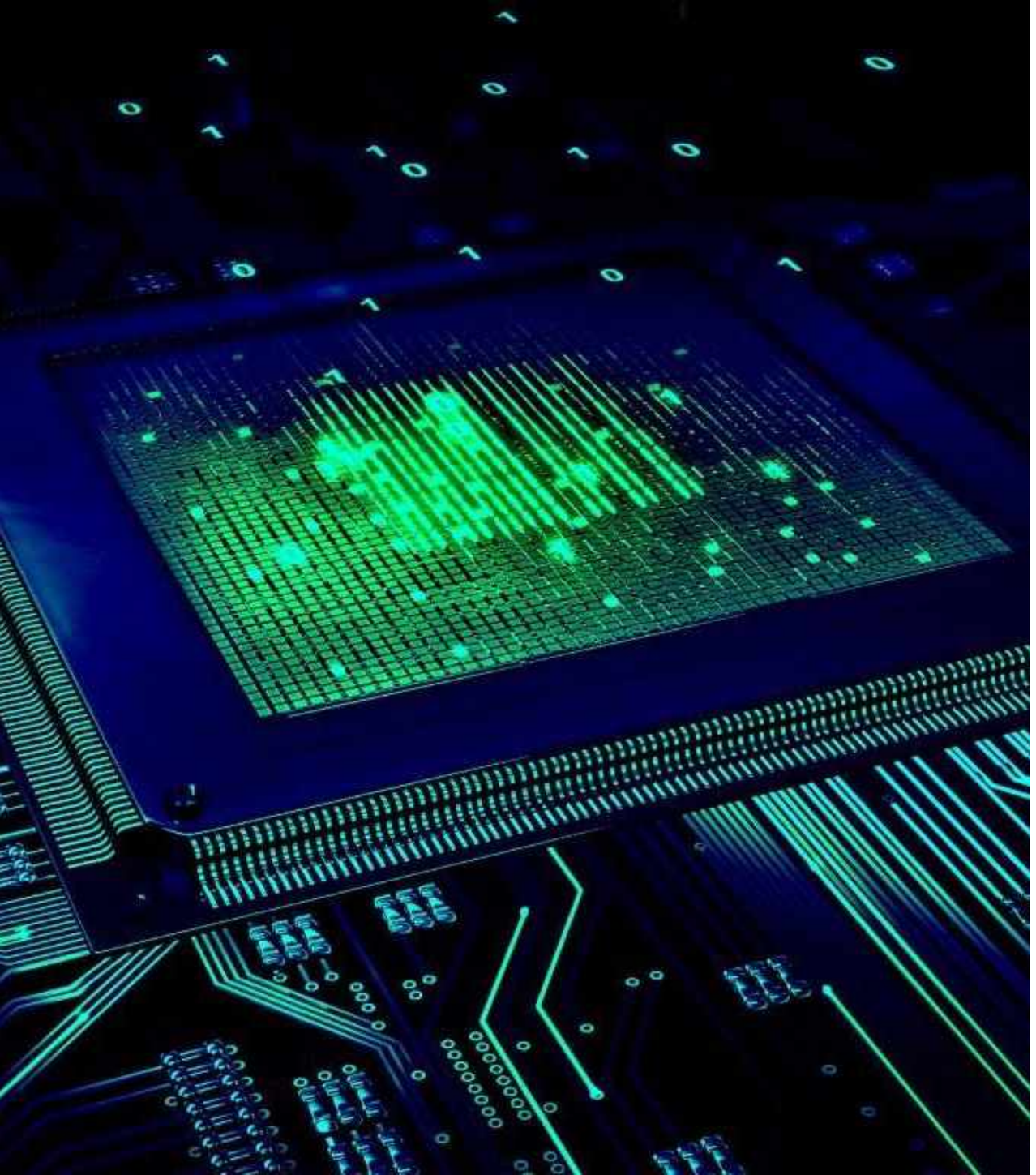
Dalam teknologi web apps biasanaya akan melakukan pencatatan mengenai user login, hal ini berguna untuk melihat pola penggunaan kredensial login pengguna atau bahkan kredensial login admin. Namun terkadang attacker akan memanfaatkan web application yang lemah dalam memonitoring logging tersebut.

Contoh Dampak yang mungkin terjadi

Ketika sebuah web apps rentan dalam pencatatan dan manajemen loggingnya maka hal ini akan mengakibatkan sulitnya deteksi sebuah serangan, karena pencatatan dan manajemen log yang tidak memadai hal ini mengakibatkan attakcer akan melakukan serangan seperti bruteforce dan lainnya untuk mencuri credential pada web apps tersebut tidak tercatat sehingga proses untuk incident handlingnya akan lebih sulit dilakukan.

Summary

Penetration testing pada Web application adalah hal yang perlu dilaksanakan untuk mengamankan web application pada suatu organisasi, upaya pengujian penetration testing berguna untuk mengukur keamanan web sebuah organisasi atau perusahaan dari serangan yang diketahui, penetration testing adalah sebuah metode dengan mensimulasi serangan “real hacker” pada web target. Web Penetration testing digunakan untuk identifikasi, analyze dan report kerentanan atau celah seperti SQL Injection, XSS, bypass autentikasi dan lainnya.



Mengapa Penetration Test pada Web perlu dilakukan? Pada umumnya Penetration testing akan melakukan banyak metodologi secara garis besar beberapa contoh diantaranya ialah:

1 Port Scanning

dilakukan untuk identifikasi layanan yang digunakan pada web application, hal ini bisa dilakukan dengan cara manual atau automated test, kemudian setelahnya akan dianalisa untuk menemukan celah atau kerentanannya

2 Vulnerability Scanning

dilakukan untuk menemukan vulnerability dan misconfiguration pada web application

3 Vulnerability Exploitation

dilakukan explotasi pada vulnerability yang ditemukan mencoba untuk mengujinya dan memberikan rekomendasi perbaikan dari celah tersebut

4 Remediation

Setalah vulnerability diperbaiki penetration tester akan melakukan pengujian ulang (re-test) untuk memastikan bahwa celah yang ada sudah benar-benar fix dan aman.

KONTAK KAMI

LOGIQUE DIGITAL INDONESIA

Ad Premier Building 19th Floor.
Jalan Tb. Simatupang No. 5
Ragunan, Ps. Minggu, Jakarta Selatan,
Indonesia 12550

info@logique.co.id

+62 21 22708935 / 36

+62 811 870 321

**nomor selular dan whatsapp*

logiquedigital