UNIVERSITAS AMIKOM YOGYAKARTA
Creative Economy Park

MATA KULIAH CYBER SECURITY

Program Studi Magister Informatika

Universitas AMIKOM Yogyakarta
Tahun 2025

Creative Economy Park

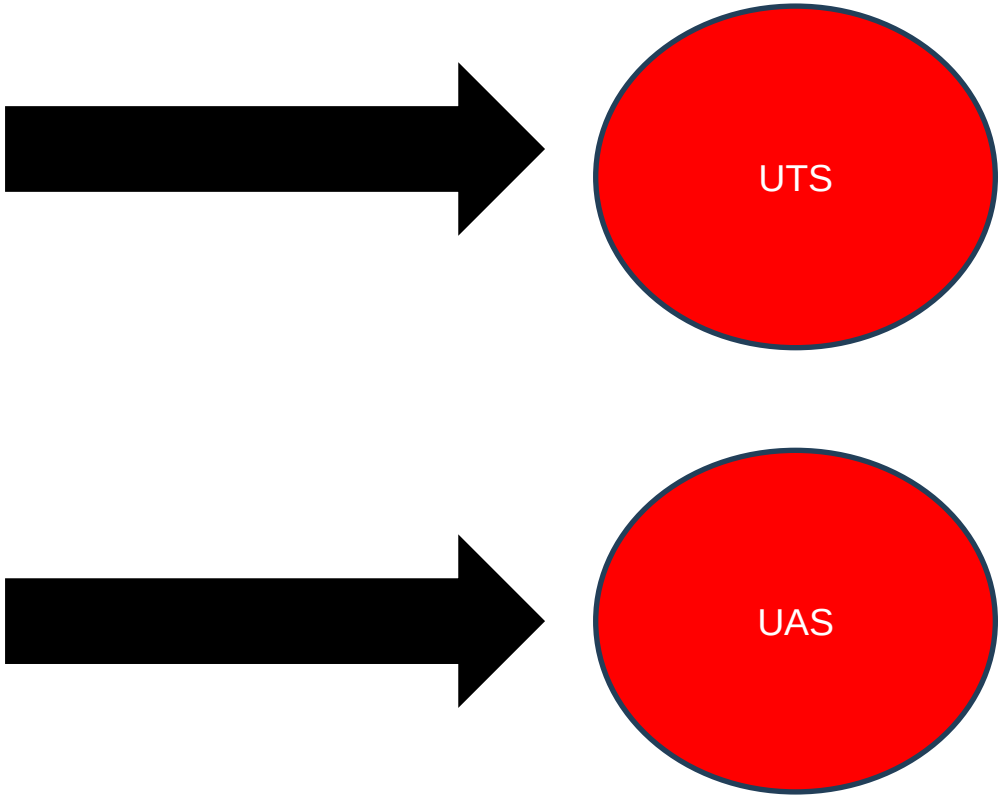
Dr. Ir. Yoga Sahria, S.Kom., M.Kom.

Fakultas Ilmu Komputer

Universitas AMIKOM Yogyakarta
Jln RingRoad Utara, Depok, Condongcatur
Yogyakarta

Outline Mata Kuliah Selama 1 Semester

1. Ruang lingkup keamanan cyber menurut framework Cyber Security Body of Knowledge (CyBoK)
2. Serangan infrastruktur jaringan
3. Malware
4. Serangan aplikasi desktop dan web
5. Social engineering
6. Solusi pengamanan data dan sistem modern
7. Regulasi dan kebijakan cyber law di Indonesia
8. Implementasi kebijakan keamanan TI
9. Penetration testing
10. Digital Forensic



UNIVERSITAS AMIKOM YOGYAKARTA
Creative Economy Park

Serangan aplikasi desktop dan web

PERTEMUAN 5

Universitas AMIKOM Yogyakarta
Tahun 2025

Creative Economy Park

Dr. Ir. Yoga Sahria, S.Kom., M.Kom.
Fakultas Ilmu Komputer
Universitas AMIKOM Yogyakarta
Jln RingRoad Utara, Depok, Condongcatur
Yogyakarta

Tujuan Pembelajaran

- Mahasiswa dapat menganalisa jenis dan bahaya malicious software
- Mahasiswa dapat menganalisa potensi bahaya dari aplikasi di internet
- Mahasiswa dapat melakukan analisa serangan malware

Injection Attack, Cross-site Scripting Attack Countermeasure serangan network, virus, dan web

Minggu 5

Dr. Ir. Yoga Sahria, S.Kom., M.Kom.

Pendahuluan

- Ancaman siber terhadap sistem verifikasi dan otentikasi semakin canggih seiring dengan kemajuan teknologi. Dua jenis serangan yang baru-baru ini semakin sering dibicarakan adalah **Presentation Attack dan Injection Attack.**

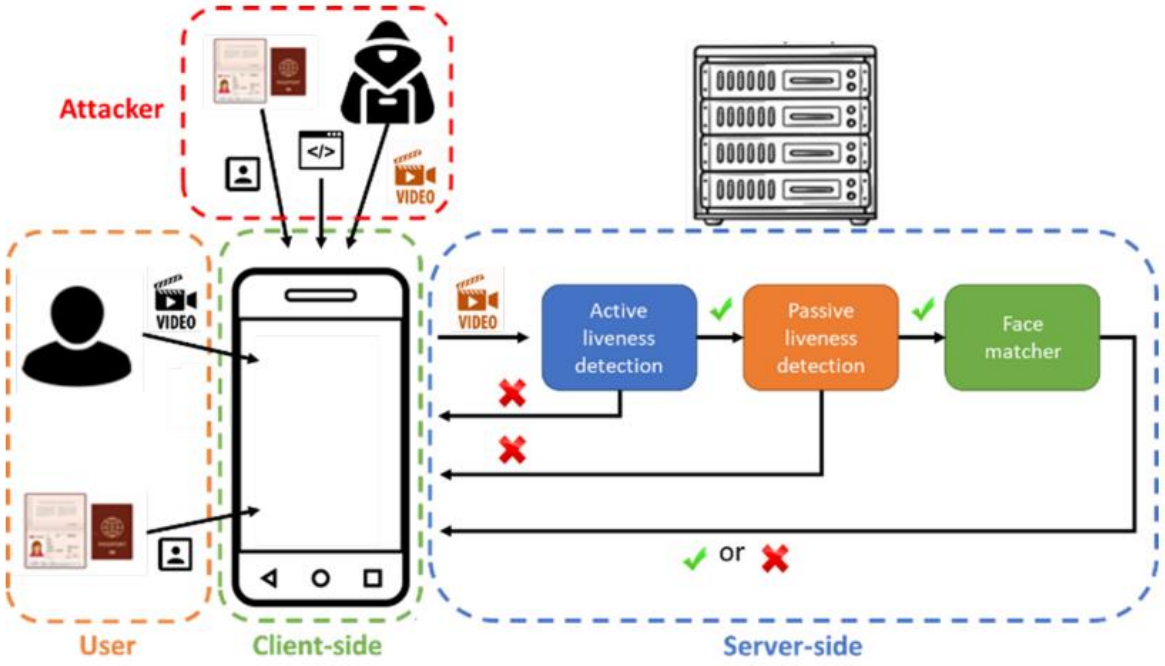
Injection Attack atau Code Injection

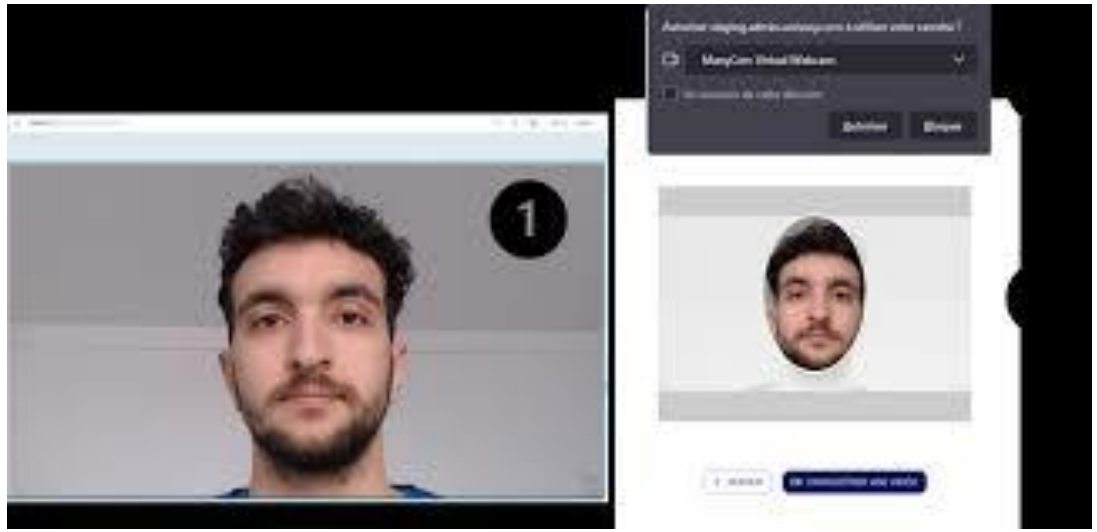
- **Injection Attack atau Code Injection** adalah upaya hacker menyuntikkan kode atau perintah yang telah dimanipulasi ke dalam sistem untuk mengecoh sistem. Ada dua jenis injection attack, yakni SQL Injection dan Deepfake Injection.
- Pada SQL Injection, penyerang menyisipkan kode berbahaya ke dalam kolom login pengguna untuk mendapatkan akses ke database. Sementara **Deepfake Injection menggunakan teknologi deepfake berupa penyuntikan data biometrik palsu** langsung ke dalam aliran data (data stream) yang diterima oleh sistem verifikasi atau otentikasi.

4 Metode serangan Code Injection

1. Kamera Virtual
2. Rooting Perangkat dan Mengaitkan API Kamera
3. Serangan Man-in-the-Middle
4. Menggunakan Emulator Perangkat

- Penyerang yang telah mendapatkan informasi pribadi korban kemudian menginstall aplikasi kamera palsu (kamera virtual) pada perangkatnya sendiri. Kamera **palsu** tersebut telah diinjeksikan dengan teknologi **deepfake.**
- Penyerang lalu mendaftar bank, online lending, dan aplikasi keuangan lainnya menggunakan informasi korban. Peran kamera palsu yang telah disuntik **deepfake adalah mengizinkan penyerang menggunakan foto korban untuk proses verifikasi biometrik.**
- **Contoh:** Seorang penyerang menginstal kamera palsu pada perangkatnya dan menggunakannya saat verifikasi biometrik dalam proses membuat akun. Penyerang tersebut menggunakan data KTP dan data biometrik korban yang didapatkannya.





Rooting Perangkat dan Mengaitkan API Kamera

- Pada perangkat yang telah di-root, penyerang dapat memperoleh akses untuk memodifikasi sistem operasi perangkat, memungkinkan mereka untuk memanipulasi API kamera. Setelah itu, mereka memodifikasi data input/output, menggantikan kamera langsung dengan video yang dimanipulasi atau pra-rekaman.
- Seorang penyerang me-root smartphone dan mengaitkan API kamera untuk memanipulasi sebuah video selama proses verifikasi identitas jarak jauh untuk aplikasi pinjaman online. Penyerang menggunakan **video pra-rekaman dari orang lain untuk melewati pemeriksaan deteksi liveness dan mendapatkan pinjaman menggunakan informasi identitas yang dicuri.**
- Metode ini juga melibatkan Deepfake Injection berupa penyuntikan data biometrik palsu langsung ke dalam aliran data (data stream) yang diterima oleh sistem verifikasi atau otentikasi. Alhasil, server menggunakan biometrik palsu tersebut dan menganggapnya sebagai data sah dari pengguna.

Serangan Man-in-the-Middle

- Penyerang mencegat komunikasi antara aplikasi dan server, seperti transmisi **gambar selfie.** Kemudian, peretas memodifikasi sistem, **memanipulasi gambar selfie,** atau mengubah **hasil liveness** untuk melewati pemeriksaan keamanan. Teknik ini membuat penyerang dapat membuat akun fiktif atau melakukan transaksi yang tidak sah tanpa terdeteksi oleh sistem keamanan aplikasi.
- Seorang penyerang bisa mencegat penerimaan data antara aplikasi dan server lewat WiFi publik. Misalnya, kamu menggunakan WiFi gratis di bandara untuk buka aplikasi penting. Hacker juga pakai WiFi bandara yang sama. Karena kalian ada di jaringan yang sama, hacker bisa langsung mengintervensi traffic. **Misalnya** waktu kamu mau transaksi ke aplikasi layanan keuangan mobile, hacker bisa tahu password kamu dan mencuri data-data kamu karena kalian berbagi jaringan WiFi.

LIFESTYLE

Bahaya! Jangan Selfie dengan Pose Dua Jari

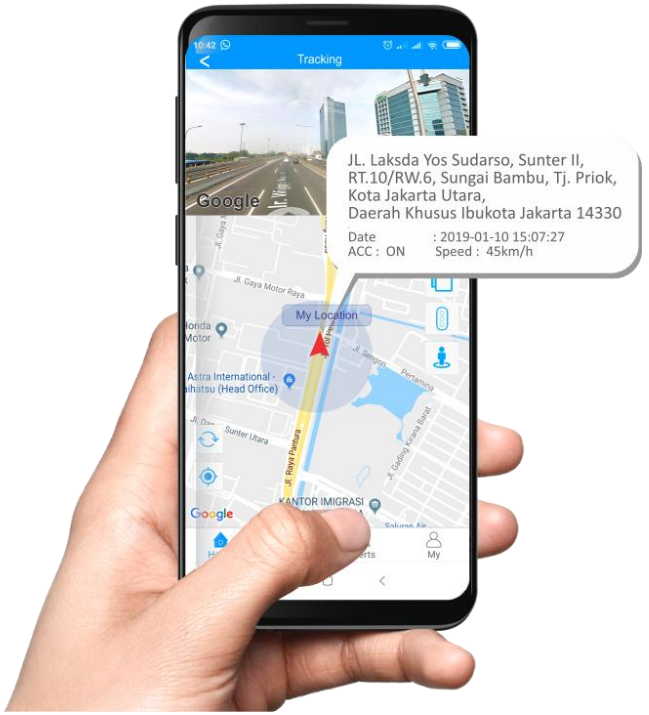
Menggunakan Emulator Perangkat

- Penyerang menggunakan emulator perangkat, yakni software program yang meniru fungsi perangkat fisik. Tujuannya adalah untuk melakukan serangan Code Injection. Dengan menjalankan aplikasi yang ditargetkan **pada emulator**, penyerang dapat dengan mudah menyuntikkan kode berbahaya, memanipulasi **feedback kamera**, atau **memodifikasi data aplikasi tanpa perlu akses** fisik ke perangkat.
- Seorang penyerang menggunakan emulator untuk menjalankan sebuah **aplikasi ride-hailing dan memanipulasi data GPS** untuk mensimulasikan **perjalanan palsu.** Penyerang menggunakan video pra-rekaman dari orang lain untuk melewati pemeriksaan deteksi liveness selama proses pendaftaran pengemudi dan membuat akun pengemudi palsu.

Pengamat: 'Tuyul' Bajak Aplikasi Gojek dan Grab Singapura

CNN Indonesia

Jumat, 26 Jul 2019 20:40 WIB



Mengatasi Ancaman Injection Attack

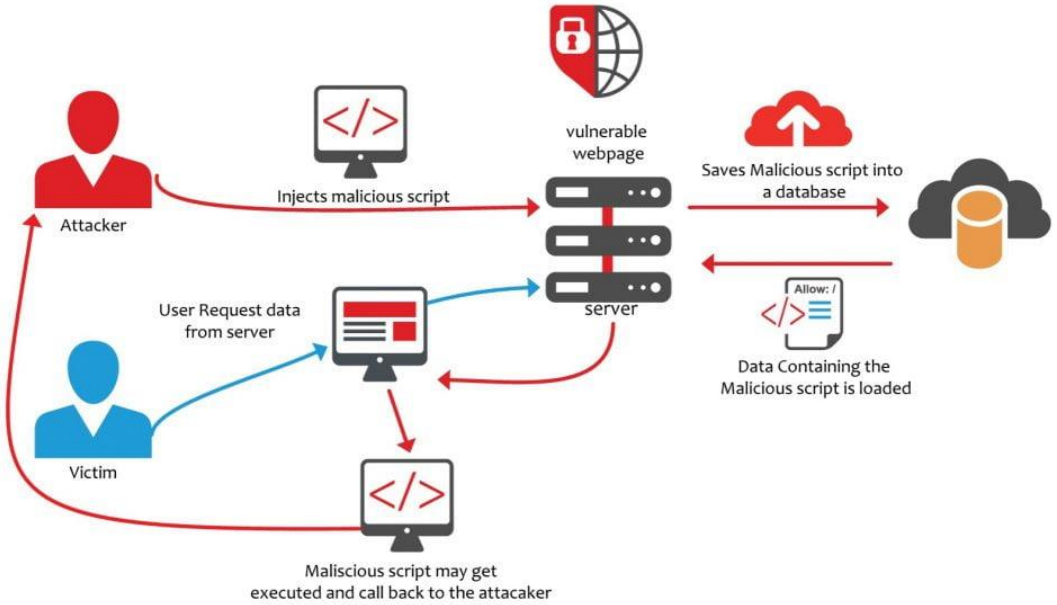
- **Presentation Attack Detection (PAD):** Yakni fitur yang mendeteksi adanya Presentation Attack dalam sistem verifikasi dengan Passive Liveness dan Morphing Detection.
- **Injection Attack Security**: Sistem untuk memastikan tidak ada injeksi kode atau perintah berbahaya ke dalam sistem verifikasi.
- **Umpan Balik Kualitas Gambar:** Pengguna mendapatkan umpan balik real time mengenai kualitas gambar ketika pengguna melakukan verifikasi biometrik.

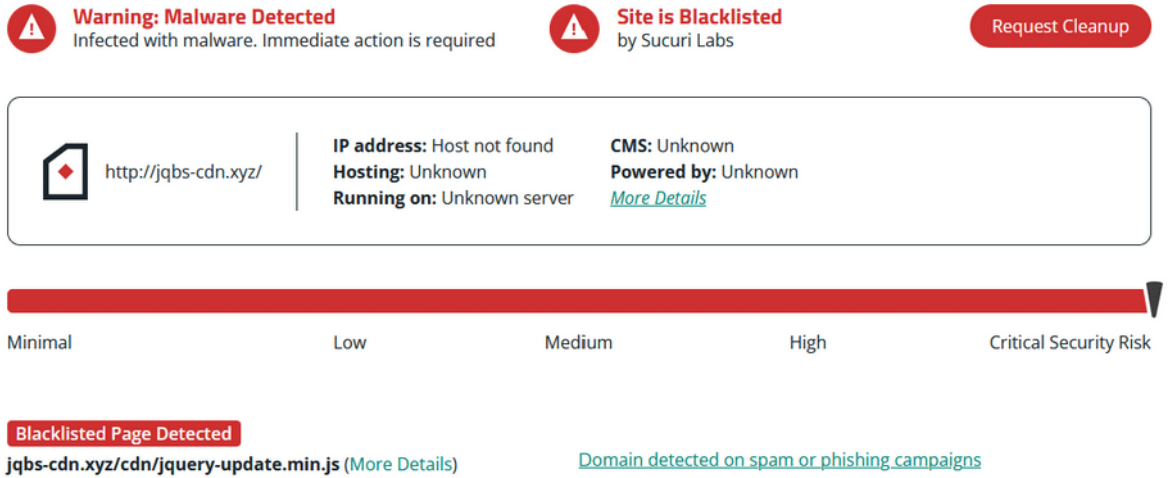
Cross-Site Scripting (XSS)

- ***Cross-Site Scripting (XSS)*** adalah salah satu jenis serangan cyber yang cukup diperhitungkan. Bahkan, serangan ini pernah menyerang platform yang ternama sekalipun seperti **Facebook, Google, dan PayPal.** Beberapa orang yang berpengalaman di bidang IT Security mungkin sudah tidak asing dengan beberapa hal yang disebut sebagai resiko keamanan pada aplikasi.

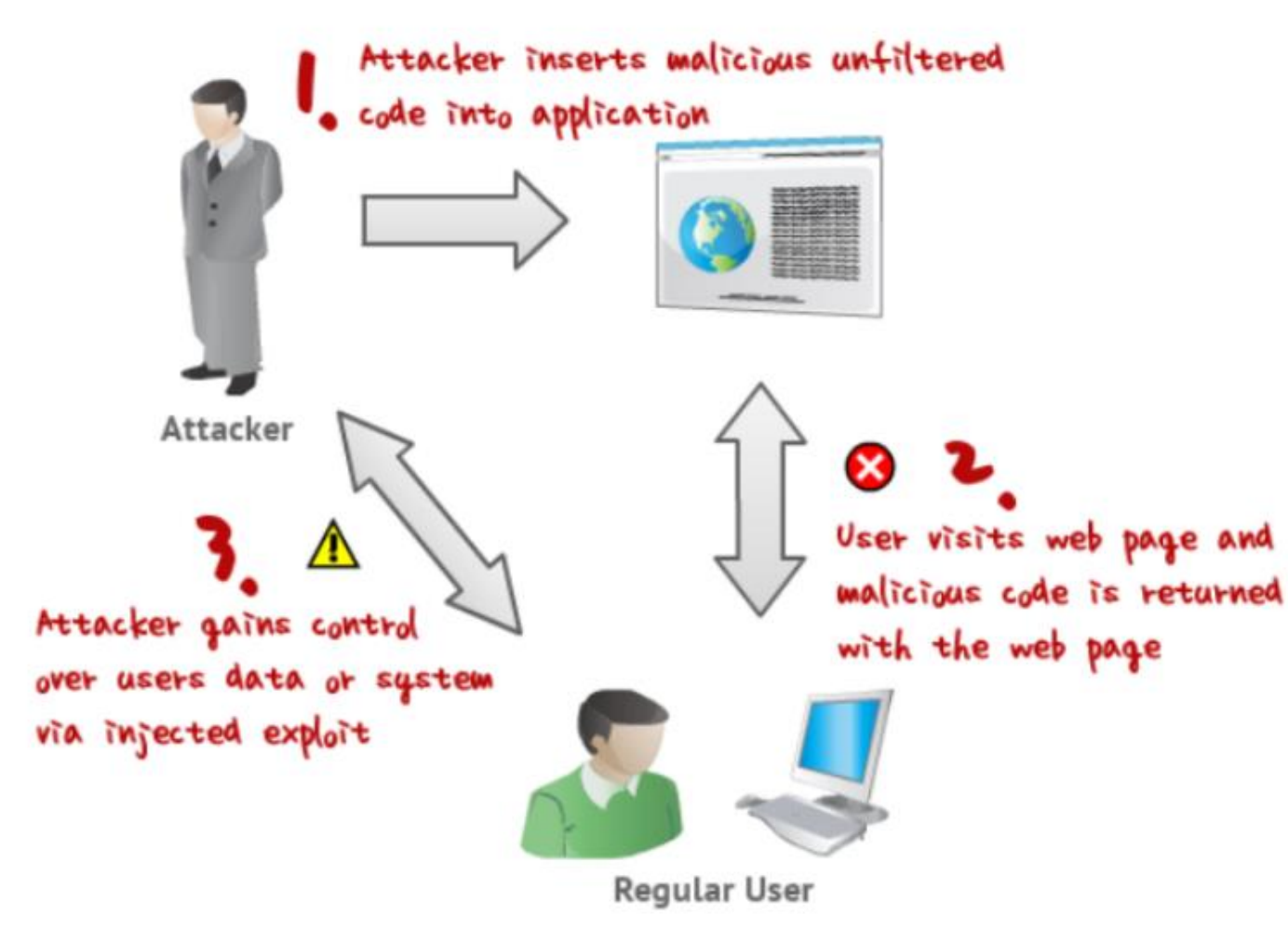

1. Hacker memasukkan kode berbahaya ke dalam website
2. Korban mengunjungi situs web tersebut dan memicu skrip berbahaya
TRUSTED WEBSITE
3. Browser korban mengeksekusi kode berbahaya tersebut dan tanpa sadar memberikan informasi milik korban
HACKER
VICTIM

- **Cross-Site Scripting,** atau biasa disebut XSS, merupakan serangan berupa code injection yang menanamkan kode/skrip berbahaya di website. Cross-site scripting memanfaatkan kerentanan pada aplikasi web berupa input dan output yang tidak divalidasi atau dikodekan.
- Dalam kata lain, Cross-site scripting **mengunakan aplikasi web** untuk mengantarkan script berbahaya ke browser dari korban karena penyerang tidak bisa langsung menjalankan script berbahaya di browser korban.
- Korban secara tidak sadar akan **mengunduh script berbahaya dan browser akan menjalankan code tersebut, script berbahaya bisa terunduh dikarenakan aplikasi web menampilkan user input tanpa di validasi terlebih dahulu.** Penyerang akan memasukkan string berupa kode kedalam input web app seperti search, form, command dan bahkan postingan. contoh serangan yang biasanya dilakukan adalah dengan menggunakan javascript karena bisa melakukan beberapa tipe serangan seperti: **cookie stealing, keylogging, dan phising.**





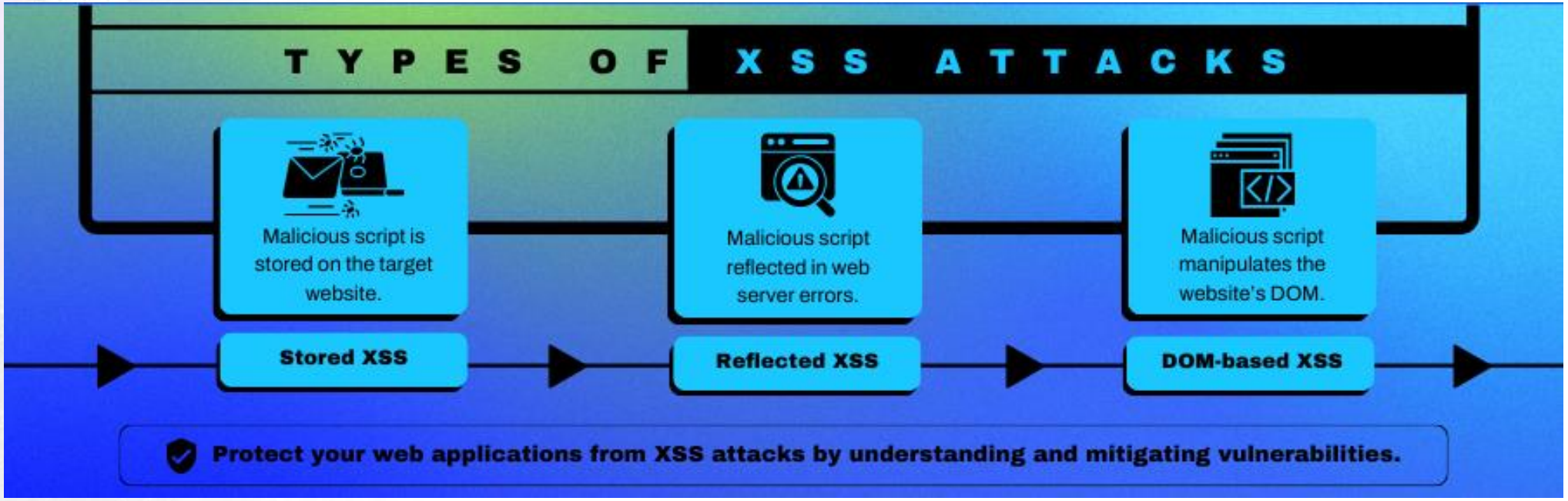
Cara Kerja XSS



- XSS sangat berbahaya bagi para developer website. Oleh karena itu, penting untuk mengetahui bagaimana cara kerja XSS. Dengan begitu, para developer website dapat mendeteksinya dari awal dan melakukan berbagai macam pencegahan dan mitigasi.
- Sederhananya, XSS bekerja dengan melakukan **eksekusi script berbahaya di browser korban dengan cara memasukkan kode berbahaya ke halaman web atau web aplikasi yang sah.** Umumnya serangan ini dilakukan menggunakan **Javascript, VBScript, ActiveX, Flash, dan bahasa sisi klien lainnya.** Nantinya, penyerang akan menghubungi para **korban melalui form kolom komentar, hingga message boards dengan mengunggah link untuk membuat script yang berbahaya.** Ketika korban melakukan klik terhadap link tersebut, script mulai menyerang dan menyamar sebagai si korban. Melalui cara inilah, para hacker dapat mengetahui data-data milik korban.

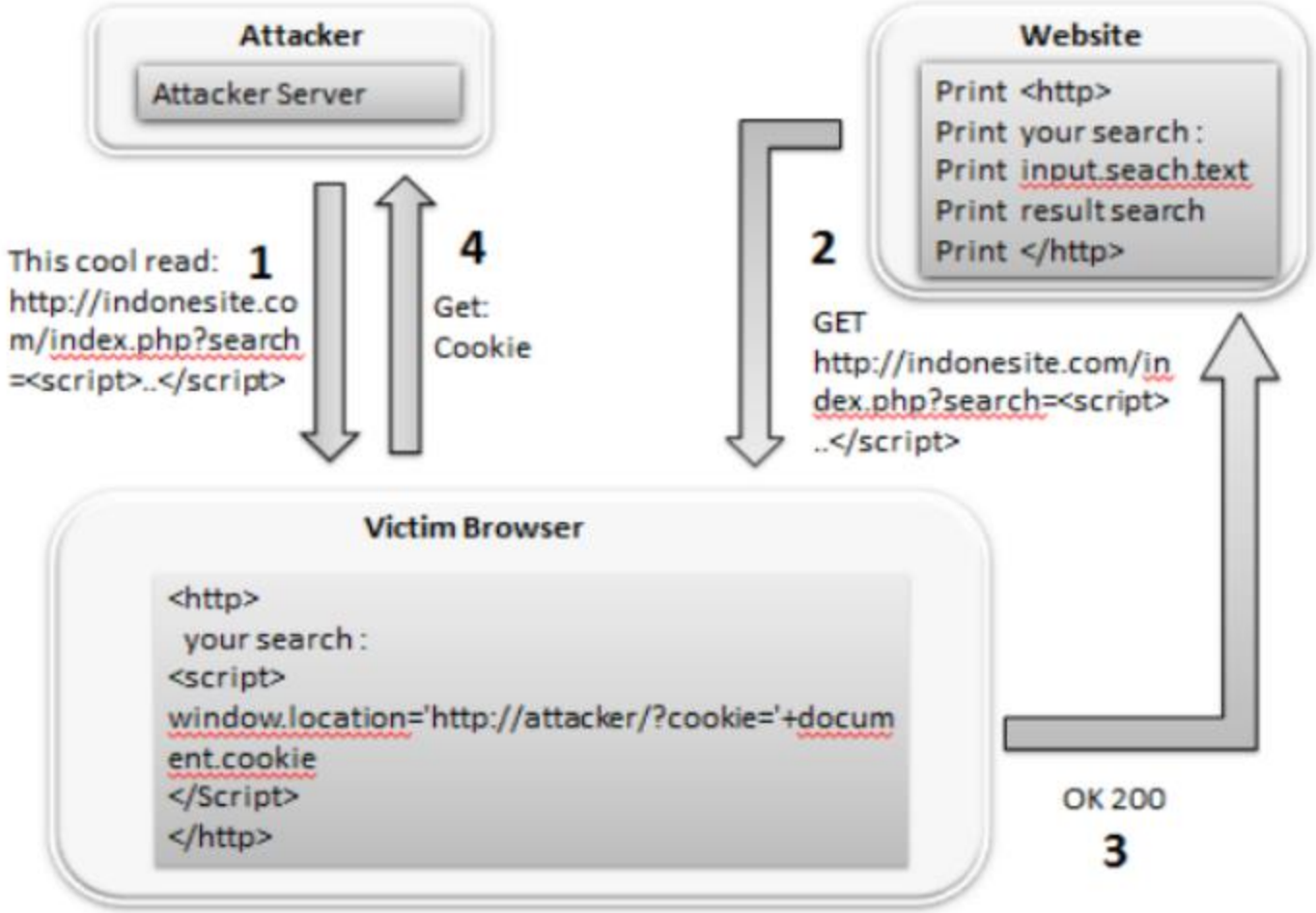
Jenis-Jenis Cross-Site Scripting (XSS)

1. Reflected Cross Site Scripting (non-persistent)
2. Stored Cross-Site Scripting (Persistent)
3. DOM-Based Cross-Site Scripting



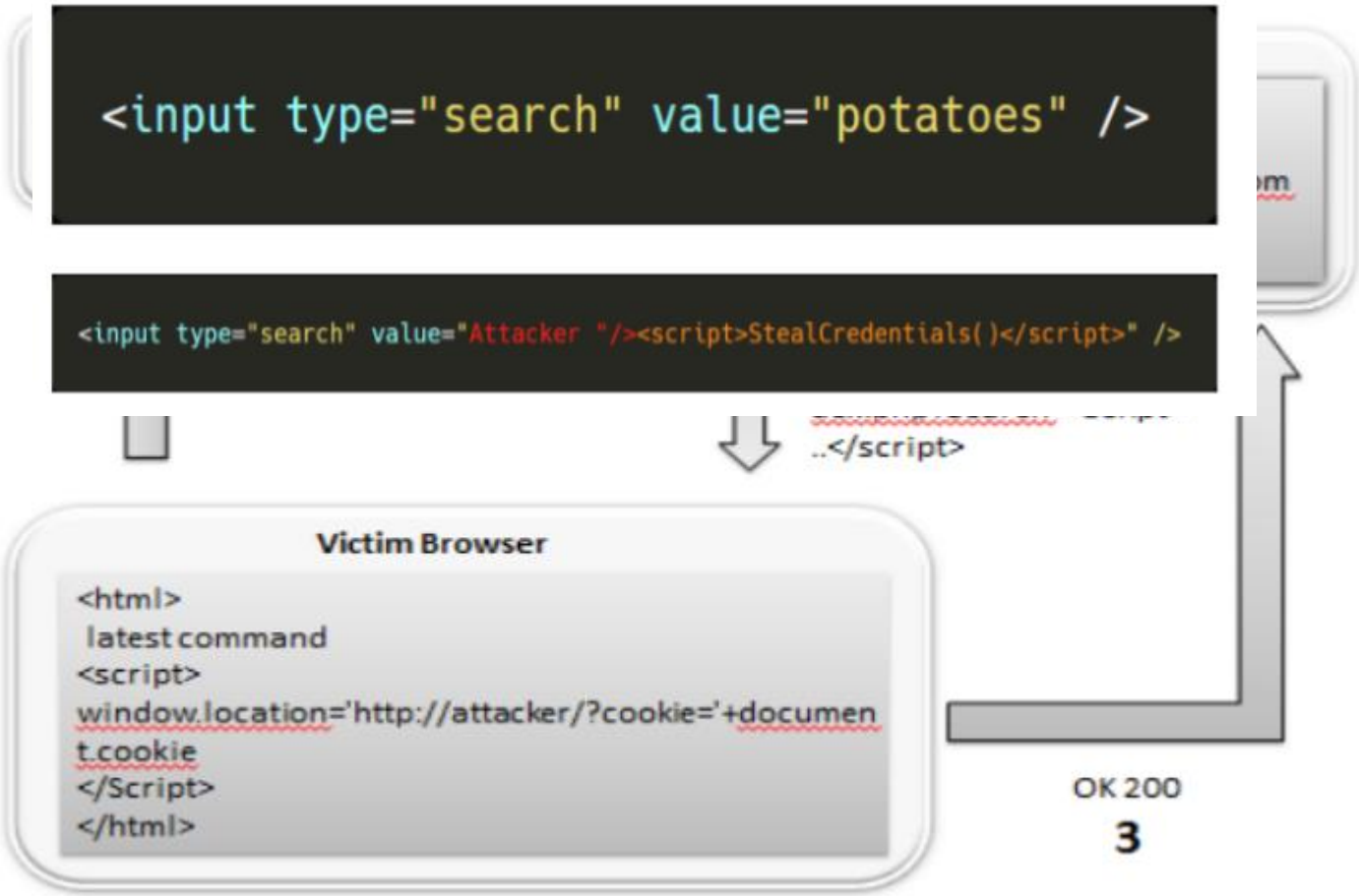
Reflected Cross Site Scripting (non-persistent)

- Reflected XSS juga dikenal sebagai serangan XSS yang tidak persisten atau menetap. Dalam kasus serangan reflected XSS, script **berbahaya dipantulkan ke situs web lain di browser pengguna**. Ini terjadi ketika input pengguna dari URL atau data POST tercermin pada halaman tanpa disimpan, sehingga memungkinkan penyerang untuk menyuntikkan konten berbahaya.
- Ini berarti bahwa seorang penyerang harus **mengirim URL jahat atau formulir posting kepada korban untuk memasukkan payload, dan korban harus mengklik tautan.** Payload semacam ini juga umumnya ditangkap oleh filter XSS bawaan di browser pengguna, seperti Chrome, Internet Explorer atau Edge.



Stored Cross-Site Scripting (Persistent)

- Stored XSS, juga dikenal sebagai XSS persisten, termasuk **memasukkan kode berbahaya langsung ke aplikasi web.** Ini terjadi ketika payload disimpan, misalnya dalam database dan kemudian dieksekusi ketika pengguna membuka halaman pada aplikasi web.
- Contoh lain dari Cross-Site **Scripting ini adalah formulir pencarian (Search form),** di mana pengunjung mengirim **Query pencarian mereka ke server**, dan hanya mereka yang melihat hasilnya..



DOM-Based Cross-Site Scripting

- DOM XSS adalah bentuk serangan XSS di mana script berbahaya hadir dalam **Document Object Model alih-alih HTML.**
- Dalam serangan Cross-Site Scripting yang reflected dan stored, kalian dapat melihat payload kerentanan di halaman respons tetapi dalam XSS berbasis DOM, kode sumber HTML dan respons serangan akan persis sama, yaitu payload tidak dapat ditemukan diresponnya. Itu hanya dapat diamati pada saat runtime atau dengan menyelidiki DOM halaman.

aksi pamer teknologi KPU itu menarik banyak perhatian dan ternyata terdapat banyak lubang dalam sistem teknologi tersebut. Xnuxer, yang bernama asli **Dani Firmansyah, sebagai pelaku yang melakukan peretasan tersebut. Xnuxer melakukan pembobolan situs KPU untuk melakukan beberapa tes sistem keamanannya saja. Ternyata, lulusan sarjana Ilmu Polituk Internasional Universitas Muhammadiyah Yogyakarta itu bisa melakukannya denga mudah bahkan tanpa *tools* khusus.**

KISAH XNUXER

LEGENDA PEMBOBOL SITUS KPU

2004

KPU mengekspos sistem ITnya yang bernilai Rp152 miliar ke media sehingga menarik perhatian

"Ini bukan sesuatu yang luar biasa"

16 April 2004

Xnuxer melihat ada lubang pada sistem IT KPU

Lalu mengetes untuk bobol situs KPU dan berhasil tanpa menggunakan tools khusus

Setelah itu, Xnuxer mengganti nama partai menjadi nama-nama nyeleneh dari Partai Jambu sampai Partai Kolor Ijo

Xnuxer
Dani Firmansyah

Xnuxer resmi ditahan setelah KPU melaporkan aksi pembobolan sistem IT ke Polda Metro Jaya

Xnuxer dituntut 1 tahun penjara dan denda Rp10 juta subsider 3 bulan kurungan tapi karena kooperatif, dia hanya dihukum 6 bulan 21 hari penjara

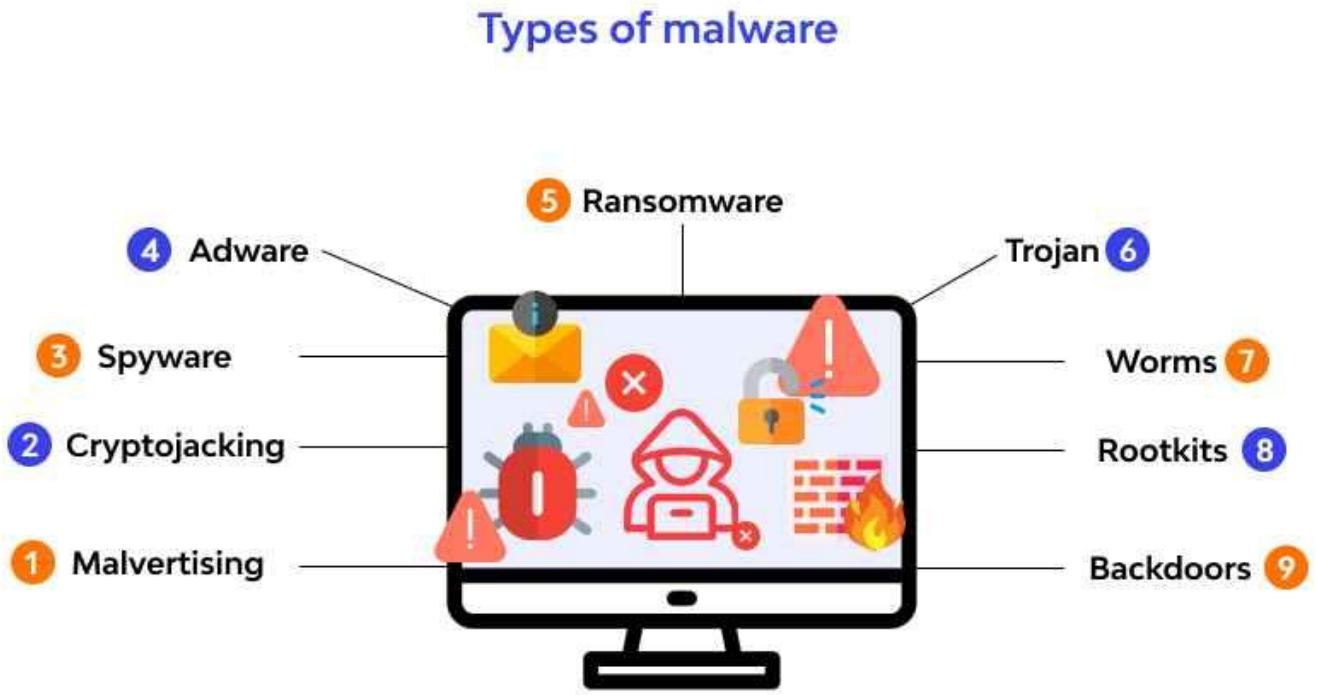
Bagaimana Mencegah Serangan Cross-Site Scripting (XSS)?

- **Periksa Keamanan Situs.** Demi menjaga keamanan aplikasi website yang dimiliki, anda perlu memastikan bahwa halaman yang membangkitkan konten secara dinamis tidak mendukung tag yang tidak diinginkan, seperti filtering, validasi, hingga encoding. Pemilik situs websites dapat menggunakan website vulnerability scanner, seperti Sucuri atau VirusTotal untuk menganalis keamanan situs. Dengan melakukan cara ini, diharapkan pemilik situs dapat mengetahui informasi lengkap tentang kelemahan dan kerentanan keamanan yang ada di dalam situs tersebut.
- **Mengadopsi Cross Boundaries Policy.** Ini memungkinkan pengguna untuk memasukkan informasi login sebagai bentuk otentikasi. Pemilik situs website juga dapat mengatur ulang dan meminta pengguna untuk memasukkan kredensial mereka pada halamn website tertentu.
- **Menambahkan SDL (Security Development Lifecycle).** Dengan menambahkan SDL, aplikasi web dapat membatasi jumlah kesalahan coding dan pelanggaran keamanan. SDL juga bisa membantu pengembangan untuk membangun perangkat lunak yang aman dan terhindar dari serangan XSS.

Virus dan Malware

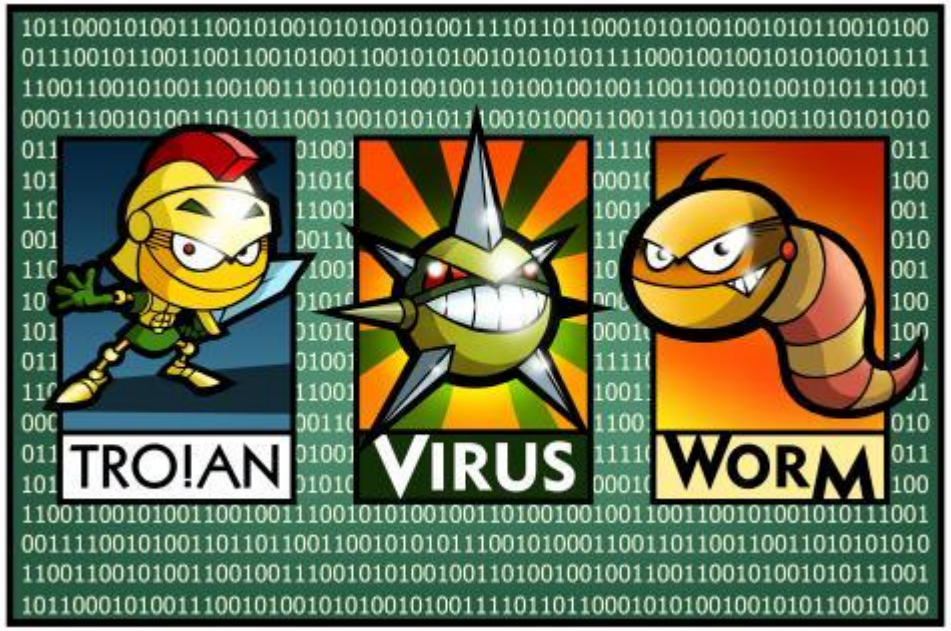
- **Virus dan malware** adalah perangkat lunak berbahaya yang dirancang untuk menginfeksi, mencuri, atau merusak data di sistem pengguna. Jenis umum dari malware termasuk:

1. **Virus**: Malware yang mereplikasi dirinya sendiri dengan menginfeksi file atau program lain.
2. **Trojan**: Malware yang menyamar sebagai program yang sah.
3. **Ransomware**: Malware yang mengenkripsi data korban dan meminta tebusan.



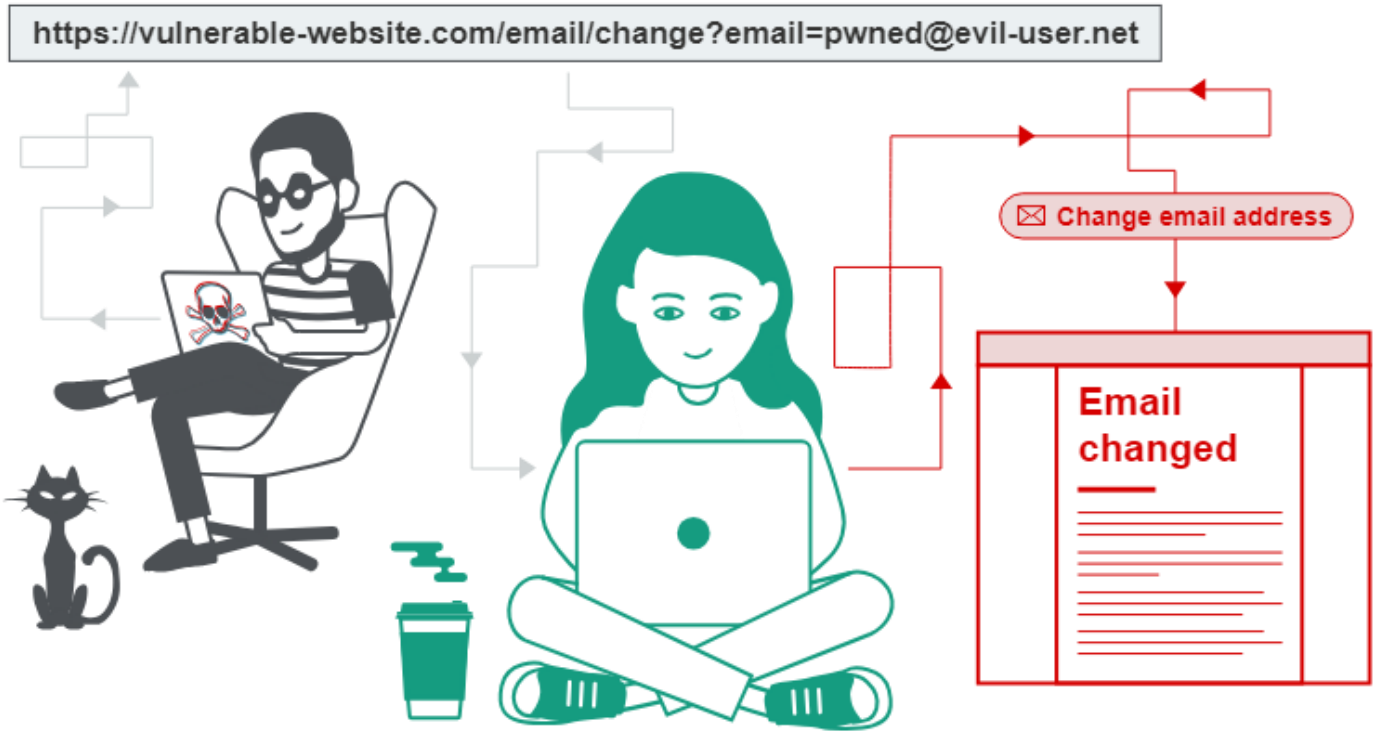
Countermeasures

- **Antivirus** Gunakan perangkat lunak antivirus untuk mendeteksi dan menghapus malware.
- **Patching dan Pembaruan** Pastikan perangkat lunak selalu diperbarui dengan patch keamanan terbaru.
- **Backup Berkala** Melakukan backup data secara rutin untuk mengurangi dampak ransomware.
- **User Awareness** Mendidik pengguna agar berhati-hati terhadap email phishing dan lampiran yang mencurigakan.



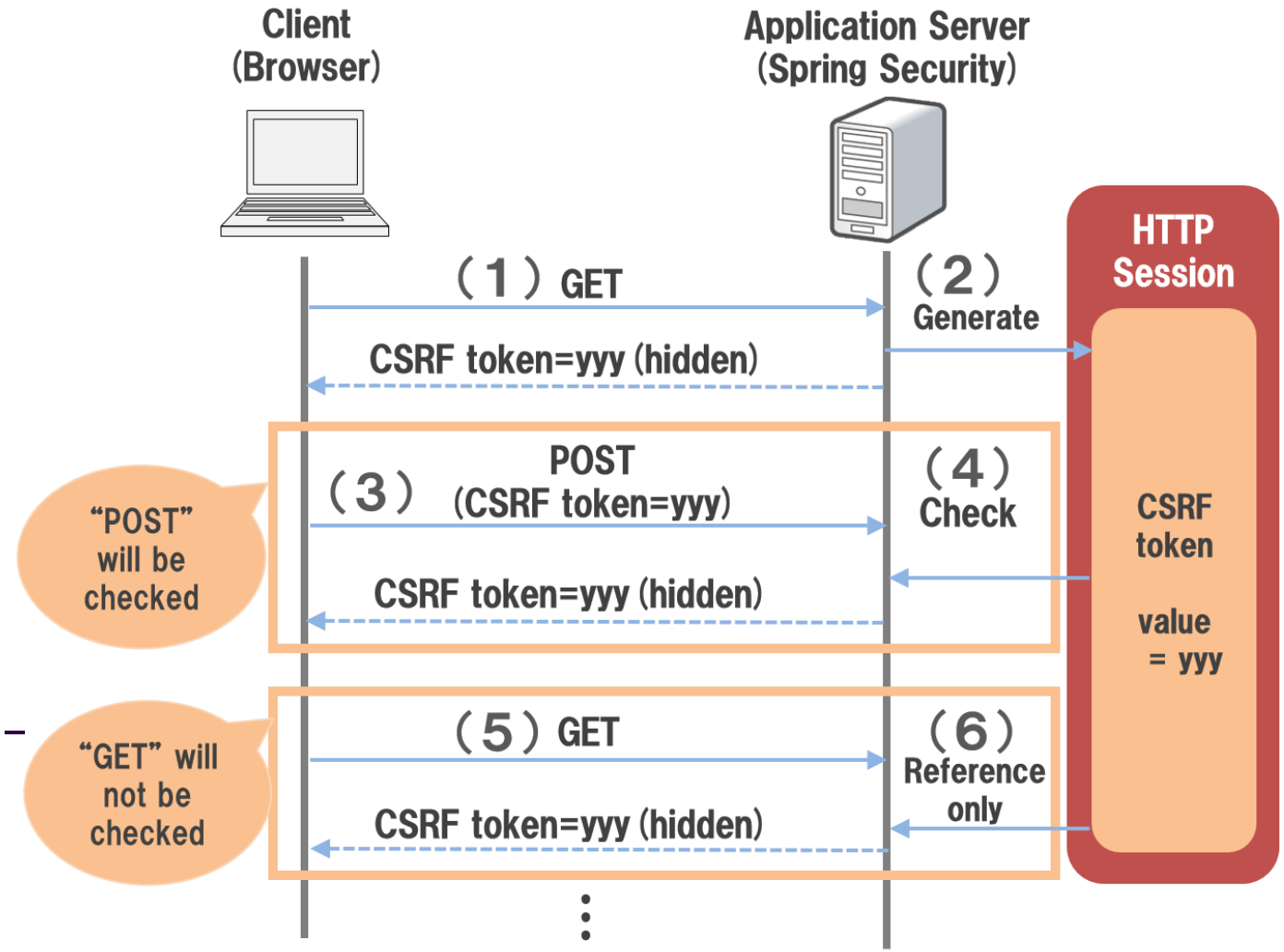
Countermeasure Serangan Web

- **Cross-Site Request Forgery (CSRF)** Serangan yang memanfaatkan sesi aktif pengguna untuk mengirimkan permintaan tanpa sepengetahuan mereka.
- **Directory Traversal** Penyerang mengakses file di luar direktori yang diizinkan dengan memanipulasi URL atau input.



Countermeasures

- **Token CSRF** Menambahkan token rahasia dalam setiap permintaan yang sah dari pengguna.
- **File Permissions** Batasi hak akses file pada server untuk mencegah akses yang tidak sah.
- **Security Headers** Gunakan header keamanan HTTP seperti X-Frame-Options, X-XSS-Protection, dan Strict-Transport-Security untuk melindungi aplikasi web.




Cara Kerja Enkripsi
Plain Text
Proses Enkripsi Data
Plain Text Terenkripsi
Special Key untuk Membuka Enkripsi
Plain Text
dewaweb

Praktek

- https://colab.research.google.com/drive/14dX9fSRqJbR1ZWIFriRFhYyMY_YeWDwC?authuser=1#scrollTo=JKrGoEbqDNPk
- https://colab.research.google.com/github/pinecone-io/examples/blob/master/learn/security/it-threat-detection.ipynb#scrollTo=fgIWJfAq_kw3

Kesimpulan

- Memahami berbagai jenis serangan dan langkah-langkah pencegahan adalah bagian krusial dari pengembangan aplikasi dan sistem yang aman. Penerapan kontrol keamanan yang tepat, seperti validasi input, penggunaan protokol keamanan, dan mitigasi serangan berbasis jaringan, dapat mencegah banyak ancaman dan melindungi data serta sistem dari kompromi.

TUGAS

	Software and Platform Security
Software Security	Known categories of programming errors resulting in security bugs, & techniques for avoiding these errors—both through coding practice and improved language design—and tools, techniques, and methods for detection of such errors in existing systems.
Web & Mobile Security	Issues related to web applications and services distributed across devices and frameworks, including the diverse programming paradigms and protection models.
Secure Software Lifecycle	The application of security software engineering techniques in the whole systems development lifecycle resulting in software that is secure by default.
	Infrastructure Security
Applied Cryptography	The application of cryptographic algorithms, schemes, and protocols, including issues around implementation, key management, and their use within protocols and systems.
Network Security	Security aspects of networking & telecommunication protocols, including the security of routing, network security elements, and specific cryptographic protocols used for network security.
Hardware Security	Security in the design, implementation, & deployment of general-purpose and specialist hardware, including trusted computing technologies and sources of randomness.
Cyber-Physical Systems Security	Security challenges in cyber-physical systems, such as the Internet of Things & industrial control systems, attacker models, safe-secure designs, and security of large-scale infrastructures.
Physical Layer & Telecommunications Security	Security concerns and limitations of the physical layer including aspects of radio frequency encodings and transmission techniques, unintended radiation, and interference.

UNIVERSITAS
AMIKOM
YOGYAKARTA