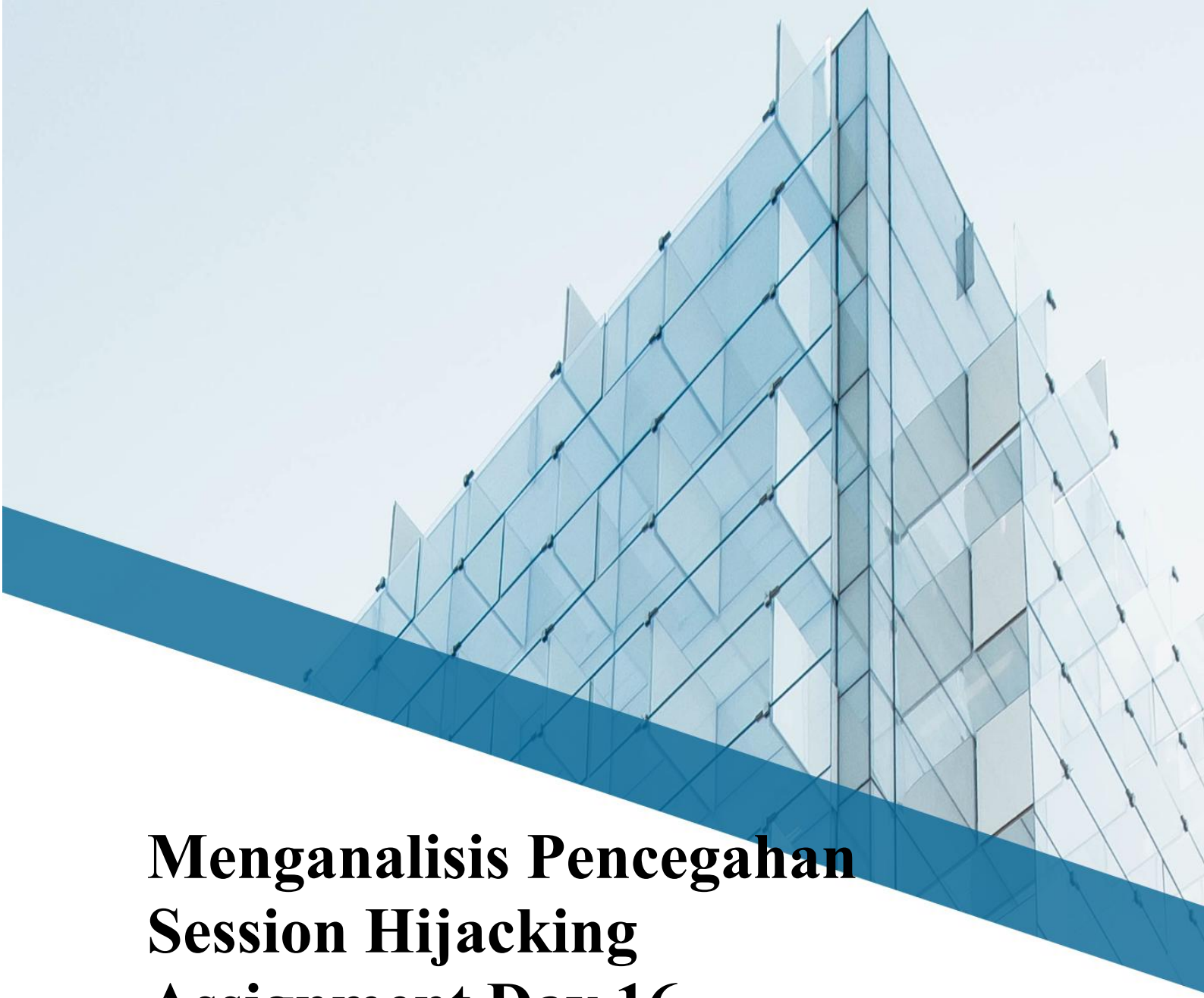


Menganalisis Pencegahan Session Hijacking Assignment Day 16

Bootcamp Cyber Security Batch 4

Disusun oleh: Sabilillah Ramaniya Widodo

!!Catatan: Diharapkan seluruh pengerjaan Assignment tidak sepenuhnya mengandalkan penggunaan AI!!

“Proses belajar ibarat menanam pohon. Jika hanya mengandalkan AI tanpa memahami esensinya, yang berkembang bukan kompetensimu, melainkan ketergantungan yang melemahkan.”
- Learning Design Dibimbing

1. Pendahuluan

Session Hijacking adalah serangan keamanan dimana seorang attacker mengambil alih sesi user yang valid di sebuah aplikasi web. Saat pengguna melakukan login, server akan membuat sebuah Session ID unik yang disimpan di browser pengguna dalam bentuk cookie. Attacker akan berusaha untuk mencuri atau memanipulasi Session ID ini untuk mendapat akses yang tidak sah ke akun pengguna tanpa memerlukan kredensial login.

Risiko dari serangan ini cukup signifikan, mulai dari pencurian data sensitif dan finansial, sampai pengambilalihan akun secara penuh yang memungkinkan penyerang untuk bertindak sebagai pengguna yang sah. Karenanya, teknik pencegahan session hijacking adalah fundamental yang wajib dipahami dan diterapkan untuk menjaga keamanan aplikasi web.

2. Analisis Kerentanan

Situs dvwa.exp-9.com adalah sebuah aplikasi web berbasis PHP/MySQL yang sengaja dirancang dengan banyak celah keamanan (vulnerabilities). Berdasarkan analisis, situs ini punya beberapa kerentanan kritis yang membuka celah untuk serangan session hijacking

A. Transmisi data tanpa enkripsi (clear text HTTP)

Situs tersebut masih menggunakan protokol HTTP standar, yang berarti seluruh komunikasi antara browser dan server, termasuk pengiriman session cookie, dilakukan dalam format teks biasa (clear text).

- Risiko: Attacker yang berada di satu jaringan (misal wifi publik) bisa melakukan teknik sniffing atau Man-in-the-Middle (MITM) menggunakan tools seperti Wireshark atau Bettercap untuk menyadap network traffic dan mengambil Session ID yang lewat.

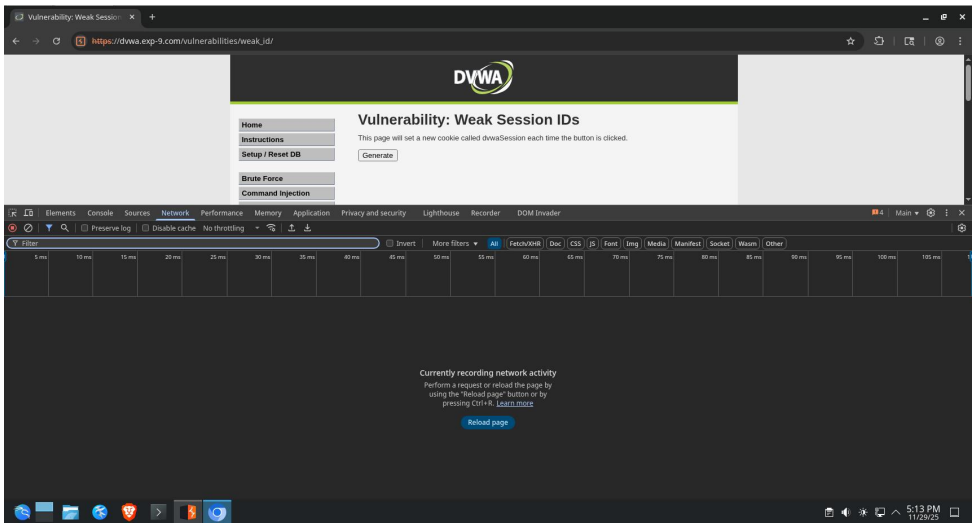
B. Tidak menerapkan Secure Cookie Flags

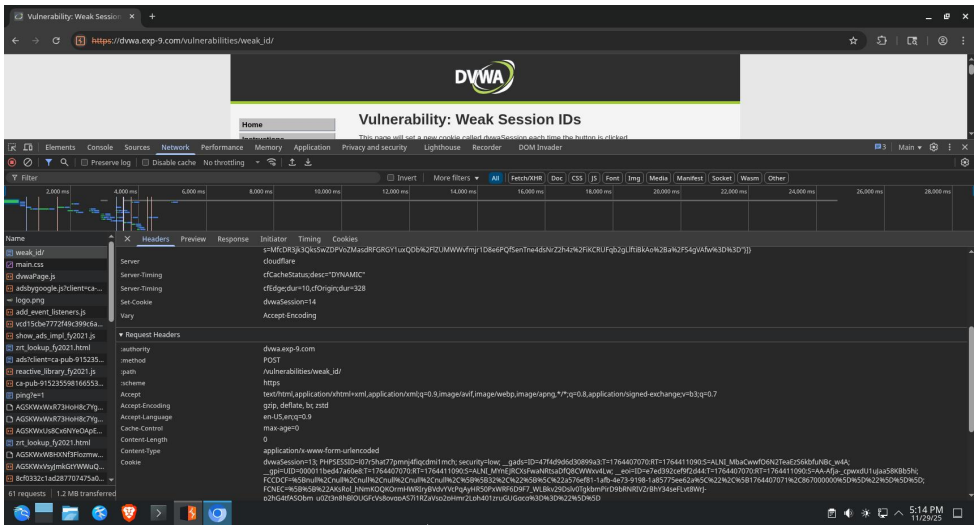
Konfigurasi cookie pada aplikasi target tidak menerapkan atribut keamanan yang memadai:

- Missing HttpOnly Flag: dimana session cookie bisa diakses oleh skrip sisi klien (client-side script) seperti Javascript. Kalau aplikasi memiliki celah Cross-Site Scripting (XSS), attacker bisa meng-inject skrip jahat untuk mencuri cookie pengguna.
- Missing Secure Flag: Cookie diizinkan untuk dikirim melalui koneksi HTTP yang tentunya sangat tidak aman, hal ini membuatnya rentan tersadap saat transit.

C. Prediksi Session ID (Weak Session Management)

Banyak aplikasi web lama atau yang tidak dikonfigurasi dengan baik menghasilkan Session ID yang berpola (misal angka urut atau timestamp), menjadikannya mudah ditebak oleh attacker tanpa perlu meretas jaringan.





3. Teknik Pencegahan

Ada 3 teknik pencegahan yang bisa dan harus diterapkan untuk memitigasi risiko:

A. Implementasi HTTPS

HTTPS mengenkripsi semua saluran komunikasi antara klien dengan server menggunakan SSL/TLS. Hal ini bisa diterapkan dengan memasang sertifikat SSL yang valid pada server web dan mengonfigurasi redirect otomatis dari HTTP ke HTTPS.

Fungsi: Bisa mencegah serangan pada network layer seperti sniffing. Walaupun attacker menyadap jaringan, data yang mereka dapat akan terenkripsi dan tidak bisa dibaca.

B. Menerapkan Secure Cookie Flags

Hal ini bisa dilakukan dengan mengonfigurasi atribut pada header Set-Cookie saat sesi dibuat.

- HttpOnly: mencegah akses ke cookie melalui Javascript, sehingga mematikan vektor serangan XSS untuk pencurian sesi.
- Secure: memastikan cookie hanya dikirim jika koneksi menggunakan HTTPS.
- SameSite: mencegah pengiriman cookie pada permintaan lintas situs (Cross-Site Request) untuk mencegah serang CSRF.

C. Implementasi HSTS (HTTP Strict Transport Security)

Yaitu mekanisme security policy yang memaksa browser untuk selalu berkomunikasi dengan server melalui HTTPS. Hal ini bisa diterapkan dengan menambah header respon Strict-Transport-Security: max-age=…; includeSubDomain
Fungsi: Supaya mencegah serangan downgrade protocol (misal SSL striping) yang mana attacker mencoba memaksa browser korban untuk kembali ke koneksi HTTP yang tidak aman.

4. Demonstrasi dan Pengujian menggunakan Burp Suit

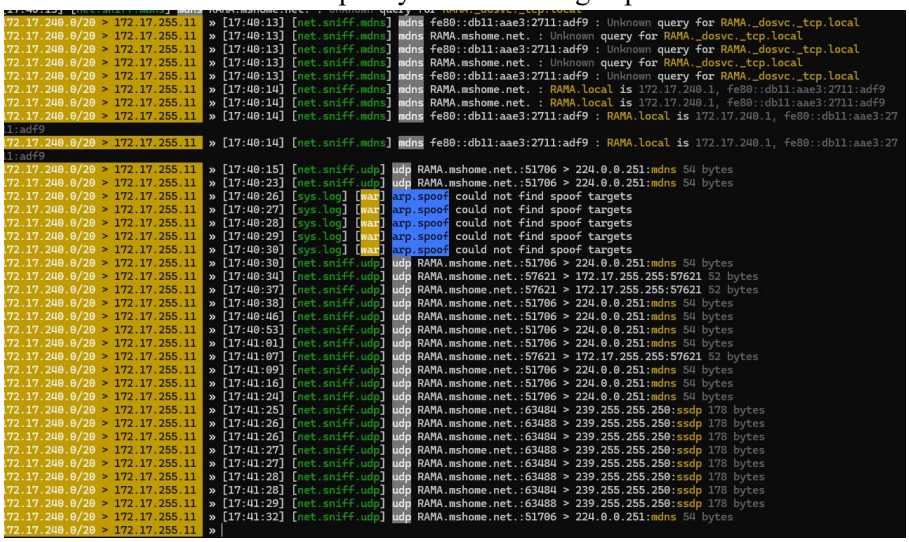
Berikut merupakan simulasi sederhana untuk menganalisis kerentanan dan pengujian pencegahannya menggunakan Burp Suite di lingkungan lab yang terkontrol:

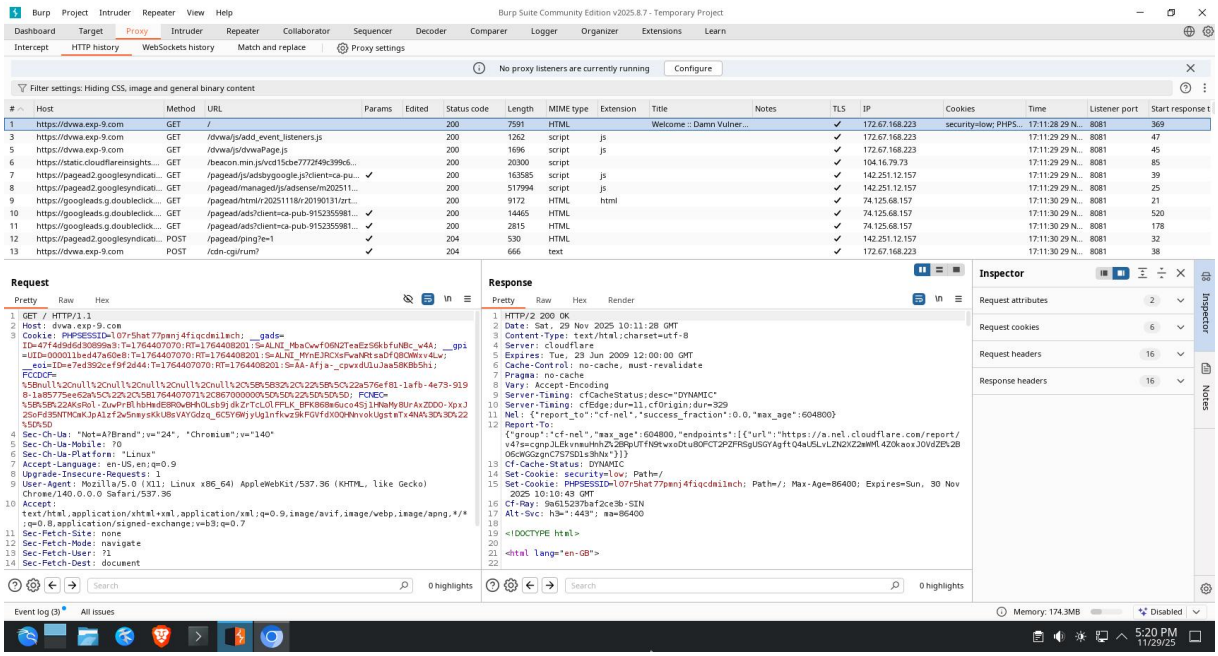
1. Mendemonstrasikan kerentanan

Konfigurasi browser untuk menggunakan Burp Suite sebagai proxy.
Kemudian buka http://dvwa.exp-9.com di browser dan lakukan proses login. Kemudian klik Weak Session ID pada side bar halaman DVWA. Aktifkan mode “Intercept is on” pada tab Proxy di Burp Suite. Lalu klik “Generate” untuk menggenerate sesi di halaman Weak Session ID. Cek di Burp Suite. Lihat pada bagian Cookie di bagian header Request:.
- Akan terlihat dvwaSession=… dan Cookie sesi (contoh, PHPSESSID=qwerty12345) yang dikirim sebagai teks biasa.
- Nilainya akan bertambah seiring kita mengklik tombol Forward di Burp Suite dan mengklik Generate lagi beberapa kali. Nanti pola dvwaSession tersebut akan naik secara berurutan: 1, 2, 3, dst.

2. Menguji metode pencegahan

Kita bisa mengubah konfigurasi server dvwa.exp-9.com dengan men-setting DVWA Security menjadi Impossible, kita bisa mensimulasikan bagaimana pencegahan akan terlihat
- Menguji HTTPS, dimana jika situs telah menerapkan HTTPS, data yang ditangkap di Burp Suite akan terenkripsi dan tidak bisa dibaca secara langsung.
- Menguji HttpOnly: dimana jika situs rentan terhadap XSS, kita bisa coba untuk menyuntikkan payload seperti <script>alert(document.cookie)</script>
- Tanpa HttpOnly: Payload tersebut akan berhasil di eksekusi dan akan menampilkan session cookie dalam sebuah kotak
- Dengan HttpOnly: Payload yang sama akan gagal mengakses cookie, dan kotak akan muncul kosong, hal ini membuktikan kalau efektivitas HttpOnly dalam mencegah pencurian cookie melalui XSS.





5. Evaluasi Efektivitas

Evaluasi pada berbagai teknik pencegahan menunjukkan kalau tidak ada satu solusi yang tunggal yang bisa mengatasi semua ancaman session hijacking. Sebaliknya, kombinasi teknik-teknik tersebut bisa memberikan pertahanan berlapis (Defense in Depth):

Teknik	Strength	Keterbatasan
HTTPS	Sangat efektif dalam mencegah penyadapan data (Sniffing) di jaringan publik.	Tidak mencegah pencurian session via serangan aplikasi seperti XSS (kalau cookie tidak diproteksi).
Secure Cookie Flags	HttpOnly sangat ampuh dalam memitigasi dampak XSS pada keamanan session.	Terlalu bergantug pada dukungan browser dan konfiguraasi server yang benar. Flag Secure butuh HTTPS.
HSTS	Bisa mencegah user error atau serangan SSL Strip dengan memaksa koneksi tetap aman.	Tidak terlalu efektif di kunjungan pertama kali (Trust on first use) sebelum header diterima browser, kecuali situs sudah didaftarkan pada HSTS preload list.

Secara keseluruhan, kombinasi dari ketiga teknik tersebut menciptakan sebuah sistem pertahanan yang cukup solid. HTTPS melindungi data di network, secure cookie flag mengamankan cookie di level browser, dan HSTS memastikan kebijakan koneksi tetap aman selalu ditegakkan setelah kunjungan pertama.

6. Rekomendasi Mitigasi

Berdasarkan analisis yang dilakukan, direkomendasikan langkah-langkah yang konkrit sebagai berikut:

1. Memprioritaskan migrasi HTTPS: dengan mewajibkan enkripsi SSL/TLS di seluruh halaman, tidak hanya pada halaman login, untuk melindungi data saat transit.
2. Mengonfigurasi Cookie dengan ketat: Hal ini bisa dilakukan dengan memastikan server hanya mengirimkan session cookie dengan atribut lengkap, yaitu Secure, HttpOnly, dan SameSite=Strict atau Lax
3. Mengaktifkan HSTS: Dengan menerapkan header HSTS dengan durasi max-age yang penjang (misal 1 tahun), hal ini bisa memastikan keamanan dalam jangka panjang.
4. Session Management: Bisa dengan menerapkan regenerasi Session ID setelah login berhasil dan mengatur session timeout yang wajar (misal 10-15 menit inaktivitas) untuk mempersempit jendela waktu serangan

7. Kesimpulan

Session Hijacking merupakan ancaman yang nyata dimana attacker memanfaatkan kelalaian korban dalam manajemen sesi web, baik di level jaringan (HTTP) maupun di level aplikasi (XSS/Weak Cookies). Analisis pada dvwa.exp-9.com membuktikan bahwa tidak adanya enkripsi dan flag keamanan membuat akun pengguna akan sangat mudah diambil alih.
Dengan demikian, keamanan sesi tidak boleh bergantung hanya pada satu metode saja. Pendekatan keamanan secara berlapus yang mengkombinasikan enkripsi jaringan (HTTPS/VPN) dan konfigurasi aplikasi yang aman (Secure Cookies) bisa dan harus menjadi standar absolut yang harus diterapkan untuk menjaga confidentiality dan integrity dari pengguna.