Evolusi Serangan Session Hijacking dan Inovasi Teknik Pencegahannya

Ahmad Anwary Adzirudin [1)], Trystan Adrian Hanggara Wibawa [2)], Rheva Anindya Wijayanti [3)], Nathanael Berliano Novanka Putra [4)]

1) Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, ahmad.anwary@student.poltekssn.ac.id
2) Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, trystan.adrian @student.poltekssn.ac.id
3) Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara, rheva.anindya @student.poltekssn.ac.id
4) Badan Siber dan Sandi Negara, nathanael.berliano@bssn.go.id

Riwayat Artikel

Dikirim 2 Agu 2025
Diterima 20 Agu 2025
Diterbitkan 31 Agu 2025

Kata kunci:

Session Hijacking
Lapisan Sesi
Keamanan
Serangan

Abstrak

Serangan *session hijacking* merupakan ancaman serius terhadap keamanan komunikasi daring karena memungkinkan penyerang mengambil alih sesi autentikasi pengguna. Penelitian ini bertujuan mengidentifikasi evolusi teknik serangan *session hijacking* sekaligus mengevaluasi efektivitas berbagai metode pencegahannya. Metode yang digunakan adalah studi literatur terhadap publikasi pada basis data IEEE Xplore, Scopus, dan Google Scholar menggunakan kata kunci bertema *session hijacking* dan mitigasi terkait. Pencarian difokuskan pada periode 2012–2024 dengan kriteria inklusi berupa artikel ilmiah berbahasa Inggris atau Indonesia yang membahas serangan dan pencegahannya pada konteks web, IoT, dan jaringan. Sebanyak 32 artikel memenuhi kriteria setelah melalui proses penyaringan. Analisis menunjukkan bahwa serangan dominan adalah *man-in-the-middle* (MITM) dan *sniffing*, sedangkan teknik pencegahan yang paling sering dibahas adalah penggunaan TLS/HTTPS, penguatan atribut cookie (*Secure*, *HttpOnly*, *SameSite*), serta penerapan *one-time cookie* (OTC). Beberapa studi empiris melaporkan bahwa OTC memberikan perlindungan kuat terhadap serangan MITM dengan overhead rata-rata kurang dari 6 milidetik per permintaan. Kontribusi utama penelitian ini adalah penyajian tabel komparatif protokol mitigasi berdasarkan cakupan ancaman, bukti empiris, dan biaya implementasi, serta rekomendasi praktis bagi pengembang sistem untuk memilih kombinasi metode yang sesuai.

Keywords:

Session Hijacking
Session Layer
Security
Attack

Abstract

Session hijacking attacks pose a serious threat to online communication security because they allow attackers to take over user authentication sessions. This study aims to identify the evolution of session hijacking attack techniques and evaluate the effectiveness of various prevention methods. The method used is a literature study of publications in the IEEE Xplore, Scopus, and Google Scholar databases using keywords related to session hijacking and related mitigation. The search focused on the period 2012–2024 with inclusion criteria of scientific articles in English or Indonesian that discussed attacks and their prevention in the context of the web, IoT, and networks. A total of 32 articles met the criteria after screening. The analysis revealed that the dominant attacks were man-in-the-middle (MITM) and sniffing, while the most frequently discussed prevention techniques were the use of TLS/HTTPS, strengthening cookie attributes (Secure, HttpOnly, SameSite), and implementing one-time cookies (OTC). Some empirical studies reported that OTC provides strong protection against MITM attacks with an average overhead of less than 6 milliseconds per request. The main contribution of this research is the presentation of a comparative table of mitigation protocols based on threat coverage, empirical evidence, and implementation costs, as

well as practical recommendations for system developers to select appropriate method combinations.

1. PENDAHULUAN

Perkembangan teknologi saat ini semakin meningkat beriringan dengan kemajuan teknologi informasi dan komunikasi. Aplikasi web dan layanan *online* menjadi salah satu bentuk perkembangan dari teknologi informasi dan komunikasi [1]. Ketika pengguna terhubung ke suatu situs web atau aplikasi, seringkali mereka perlu melakukan login dengan memberikan informasi pengguna seperti *username* dan *password*. Setelah login berhasil, situs web akan menciptakan sesi (*session*) yang menyimpan informasi otentikasi pengguna selama periode waktu tertentu [2]. Lapisan sesi sendiri adalah komponen dari model OSI (*Open Systems Interconnection*) yang bertanggung jawab untuk mengatur, mengontrol, dan menjaga koneksi antara dua perangkat yang berkomunikasi. Tujuan utama dari lapisan sesi adalah untuk memastikan adanya komunikasi yang andal dan efisien antara entitas-entitas yang terlibat dalam proses pertukaran data [3].

Dalam proses pertukaran data, keamanan atau *security* adalah hal yang sangat penting karena terdapat berbagai serangan yang mungkin terjadi. Biasanya, serangan ini terjadi melalui celah keamanan dalam mekanisme autentikasi (*authentication*) yang digunakan dalam sistem. Autentikasi adalah proses untuk memastikan keaslian identitas pengguna atau entitas yang berusaha mengakses sistem. Ini dapat melibatkan penggunaan *username* dan *password*, sertifikat digital, sidik jari, atau metode lainnya untuk memverifikasi bahwa entitas tersebut memang benar-benar yang mereka klaim [4]. Salah satu serangan yang cukup merugikan adalah *session hijacking*.

Serangan *session hijacking* dilakukan dengan maksud untuk mencuri, mengendalikan, atau memanipulasi sesi yang sedang berlangsung antara pengguna yang sah dan sistem yang dilindungi. Serangan *session hijacking* dapat terjadi dalam berbagai bentuk, termasuk serangan *Man-in-the-middle* (MITM) di mana penyerang mencuri atau memanipulasi data yang dikirimkan antara pengguna dan sistem, serangan penggantian sesi (*session replacement*) di mana penyerang mengambil alih identitas sesi yang sah, atau serangan rekayasa sosial (*social engineering*) di mana penyerang memperoleh informasi sensitif dari pengguna untuk memanipulasi sesi yang sedang berlangsung [5].

Dalam menghadapi serangan *session hijacking*, penting untuk mengimplementasikan langkah-langkah keamanan yang tepat. Beberapa tindakan yang dapat dilakukan meliputi penggunaan protokol komunikasi yang aman seperti HTTPS untuk melindungi data selama transmisi, penerapan mekanisme autentikasi yang kuat seperti autentikasi berbasis token atau otentikasi dua faktor (2FA), penggunaan mekanisme enkripsi yang kuat untuk melindungi data sensitif, serta penggunaan tanda waktu (*timestamp*) dan identifikasi unik untuk setiap sesi [6]. Dengan menerapkan langkah-langkah keamanan yang tepat, seperti memperkuat autentikasi, mengenkripsi komunikasi, dan memonitor aktivitas jaringan, sistem dapat dilindungi dengan lebih efektif dari serangan *session hijacking*. Hal ini akan membantu memastikan bahwa sesi yang sedang berlangsung aman dari ancaman dan data sensitif terjaga keutuhannya [7].

Berdasarkan latar belakang tersebut, penelitian ini diarahkan untuk menelusuri evolusi teknik serangan *session hijacking* dalam berbagai konteks aplikasi web dan layanan daring, sekaligus mengevaluasi berbagai metode pencegahan yang ditawarkan dalam literatur terkini. Analisis dilakukan tidak hanya untuk memahami kecenderungan serangan yang muncul, tetapi juga untuk menilai efektivitas, efisiensi, serta potensi implementasi nyata dari protokol atau mekanisme pencegahan yang ada. Dengan pendekatan tersebut, penelitian ini akan berfokus untuk menelusuri evolusi teknik serangan *session hijacking* dalam literatur tahun 2012–2024 serta mengevaluasi efektivitas protokol pencegahan. Kontribusi utama artikel ini adalah memberikan analisis kritis dan rekomendasi terkait penerapan protokol keamanan yang paling sesuai untuk aplikasi web, transaksi finansial, dan jaringan *ad-hoc*.

2. LANDASAN TEORI

2.1. *Open Systems Interconnection* (OSI) *Layer*

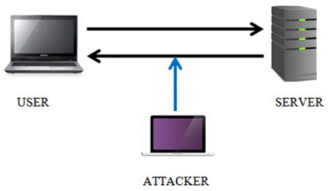
OSI *Layer* adalah sebuah model yang digunakan untuk menggambarkan dan mengorganisir fungsionalitas jaringan komputer yang dikembangkan pada tahun 1984. OSI model dibagi menjadi tujuh

layer sebagai berikut.

1. *Application Layer*
   Application layer bertanggung jawab atas pertukaran informasi antara program komputer dan berfungsi sebagai antarmuka aplikasi dan fungsionalitas jaringan.
2. *Presentation Layer*
   Presentation Layer bertanggung jawab terkait data yang dikonversi dan diformat untuk transfer data.
3. *Session Layer*
   Session Layer adalah layer ke-5 dalam model referensi OSI yang bertanggung jawab untuk mengelola dan mempertahankan sesi komunikasi antara dua entitas yang saling berhubungan dalam jaringan [16].
4. *Transport Layer*
   Transport Layer menyediakan suatu mekanisme pengiriman data antara dua perangkat meliputi pengaturan ukuran segmen, deteksi kesalahan, dan retransmisi data yang hilang.
5. *Network Layer*
   Network Layer bertanggung jawab menentukan alamat jaringan, menentukan rute yang harus diambil.
6. *Data Link Layer*
   Data Link Layer memiliki tugas utama memberikan fasilitas transmisi *raw* data dan mentransmisikan data tersebut kedalam saluran yang bebas dari kesalahan transmisi.
7. *Physical Layer*
   Physical Layer bertanggung jawab atas proses data menjadi bit dan mentransfernya melalui media dan menjaga koneksi fisik antarsistem.

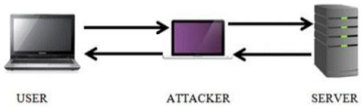
2.2. *Session Hijacking*

Session hijacking merupakan aksi pengambilan kendali *session* milik *user* lain dengan cara memperoleh autentifikasi ID *session* yang biasanya tersimpan dalam *cookies*. *Session hijacking* dapat dibagi menjadi dua, yaitu *session hijacking* aktif dan *session hijacking* pasif. *Session hijacking* aktif mengambil alih kontrol terhadap koneksi jaringan aktif pengguna. Serangan ini dilakukan dengan mengirimkan lalu lintas atau permintaan berlebihan ke *server* sehingga menyebabkan *server* tidak tersedia untuk layanan yang sah karena terfokus untuk menangani lalu lintas data yang palsu. *Session hijacking* aktif dapat diilustrasikan seperti pada Gambar 1 berikut.



Gambar 1. *Session Hijacking* Aktif

Sedangkan *session hijacking* pasif adalah teknik dimana penyerang menyusup diantara *user* dan *server* yang valid tanpa mengubah koneksi antara keduanya. Dalam serangan yang terjadi, penyerang bertindak sebagai perantara yang menangkap seluruh paket data dari jaringan antara *user* dan *server* untuk memahami interaksi yang terjadi. Gambaran *Session Hijacking* pasif dapat dilihat pada Gambar 2 berikut.



Gambar 2. *Session Hijacking Pasif*

Session Hijacking merupakan suatu serangan yang cukup membahayakan pengguna. Oleh karena itu perlu adanya solusi untuk mengatasi hal tersebut, mulai dari menggunakan berbagai *tools* ataupun protokol dan algoritma. Namun, terdapat beberapa hal yang dapat dilakukan dengan mudah sebagai tindakan preventif terkena serangan *session hijacking*, diantaranya sebagai berikut.

1. Menghindari penggunaan wifi public,
2. Menggunakan VPN yang aman,
3. Menambahkan *software* pengaman atau anti virus,
4. Menggunakan website HTTPS,
5. Hindari membuka tautan yang sumbernya tidak jelas,
6. Mengakhiri sesi di setiap akhir pemakaian.

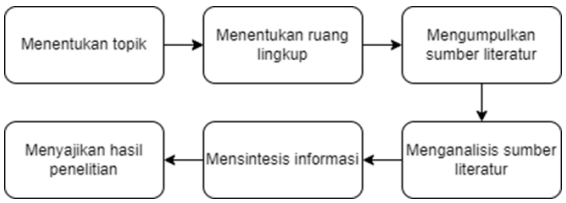
2.3. Metode *Session Hijacking*

Pengambilan sesi pada *session hijacking* dapat dilakukan dengan berbagai cara dan metode. Beberapa metode yang sering digunakan pada serangan *session hijacking* adalah sebagai berikut.

1. *Man–the-middle attack*
   Man-in-the-middle attack (MITM) adalah serangan keamanan dimana penyerang menempatkan diri diantara dua pihak yang sedang berkomunikasi. Dalam hal ini, penyerang menyamar sebagai salah satu atau kedua pihak tersebut dan memantau, memanipulasi, hingga menyadap data dalam proses komunikasi [28].
2. *Cross-Site Scripting* (XSS)
   XSS adalah metode lain dari serangan yang bertujuan mencuri ID *session*. XSS juga dikenal sebagai serangan injeksi kode pada sisi klien, di mana kode berbahaya dapat dieksekusi ke situs web atau aplikasi web. Serangan ini menggunakan kerentanan pada situs web dan apapun yang dimasukkan oleh pengguna ke dalam situs web akan dianggap sebagai teks biasa tanpa dienkripsi atau dikodekan, lalu informasi tersebut dikirimkan ke penyerang [21].
3. *Sniffing*
   Sniffing adalah salah satu cara penyerangan di mana seorang penyerang mencuri data dari jaringan dengan memantau lalu lintas jaringan untuk mencari paket yang tidak terenkripsi. Setelah menemukan paket tersebut, penyerang akan mengambil alih ID *session* yang ada. Dengan ID *session* tersebut penyerang memiliki kemampuan untuk membuat sesi baru dan mendapatkan semua informasi yang dikirimkan antara *user* dan *server* [21].
4. *Spoofing*
   Spoofing adalah suatu bentuk serangan atau teknik dalam keamanan komputer di mana penyerang mencoba untuk menyembunyikan atau memanipulasi identitas, alamat, atau sumber pesan, komunikasi, atau data agar terlihat seolah-olah berasal dari sumber yang sah atau tepercaya. Tujuan utama dari serangan *spoofing* adalah untuk menipu korban atau sistem agar percaya bahwa data yang diterima adalah asli dan berasal dari sumber yang sebenarnya dipercayai, padahal sebenarnya data tersebut dikendalikan atau dimanipulasi oleh penyerang [26].
5. *Brute Force*
   Brute Force adalah salah satu serangan terkenal yang dapat memecahkan kombinasi karakter, angka, simbol, dan karakter khusus apa pun yang digunakan dalam nama pengguna, kata sandi, atau kata apa pun. Serangan *Brute Force* digunakan untuk meretas sesi dengan menguji semua kombinasi yang mungkin hingga menemukan ID *session* yang benar. Setelah berhasil, penyerang dapat mengambil alih sesi yang sudah dibuat dan memanfaatkannya untuk mendapatkan akses yang tidak sah pada jaringan atau sistem target [21].
6. SSL *Stripping*
   SSL *Stripping* adalah serangan keamanan komputer yang dilakukan oleh penyerang dengan tujuan untuk menyingkirkan atau menurunkan koneksi SSL/TLS (*Secure Socket Layer*/*Transport Layer Security*) antara klien dan *server*. SSL/TLS adalah protokol yang mengenkripsi komunikasi antara *user* dan *server*, sehingga data yang dikirimkan melalui jaringan menjadi terenkripsi dan lebih aman dari serangan penyerangan seperti penyadapan [30].
7. *Malware*
   Teknik serangan ini digunakan oleh penyerang untuk mencuri atau mengambil sesi *user* dengan memanfaatkan *malware* yang telah ditanamkan pada perangkat korban. Ketika *malware* berhasil ditanamkan, penyerang dapat menyusup dan bersembunyi di sistem korban, mencuri informasi, mengakses data sensitif, dan menyusup ke sesi yang sedang berjalan [29].

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam artikel ini adalah studi literatur (*literature review*). Pendekatan ini dipilih karena tujuan penelitian bukan untuk melakukan eksperimen langsung, melainkan untuk menelusuri, mengkaji, mengevaluasi, dan mensintesis berbagai hasil penelitian sebelumnya yang relevan dengan topik serangan *session hijacking* dan teknik pencegahannya. Dengan demikian, penelitian ini bersifat analitis dan sintesis dari berbagai sumber, bukan pengusulan konsep atau metode baru. Proses studi literatur dilakukan secara sistematis melalui beberapa tahapan sebagaimana ditunjukkan pada Gambar 3 berikut [31].



Gambar 3. Tahapan Metode Penelitian

1. Menentukan Topik
   Penulis menentukan topik, ruang lingkup, dan batasan penelitian yang difokuskan pada evolusi metode serangan *session hijacking* serta inovasi teknik pencegahan yang dilaporkan dalam literatur.
2. Menentukan Ruang Lingkup
   Penelusuran literatur dilakukan menggunakan basis data akademik seperti IEEE Xplore, Scopus, dan Google Scholar. String pencarian yang digunakan mencakup kata kunci: *session hijacking*, *session security*, *one-time cookie*, dan *session prevention protocol*. Rentang tahun publikasi yang diprioritaskan adalah 2012–2024, dengan bahasa utama Inggris, namun beberapa literatur berbahasa Indonesia yang relevan juga dipertimbangkan.
3. Mengumpulkan Sumber Literatur
   Literatur yang diperoleh kemudian disaring dengan mempertimbangkan kriteria inklusi dan eksklusi. Kriteria inklusi mencakup artikel jurnal, prosiding konferensi, atau buku yang secara teknis membahas metode serangan maupun teknik pencegahan *session hijacking*. Sementara itu, kriteria eksklusi digunakan untuk menyingkirkan artikel yang terlalu umum, duplikat, tidak relevan dengan fokus penelitian, atau tidak memiliki pembahasan teknis yang memadai.
4. Menganalisis Sumber Literatur
   Literatur yang telah terseleksi dibaca dan dipahami secara mendalam untuk mengidentifikasi temuan-temuan penting, data teknis, protokol keamanan yang digunakan, serta argumen-argumen yang relevan. Pada tahap ini dilakukan analisis kritis yang tidak hanya bersifat deskriptif, tetapi juga mengevaluasi kualitas penelitian dengan memperhatikan efektivitas protokol, frekuensi keberhasilan mitigasi, cakupan metode serangan yang dapat dicegah, serta efisiensi implementasi dari masing-masing pendekatan..
5. Mensintesis Informasi
   Hasil dari berbagai literatur disintesis melalui perbandingan dan evaluasi antar penelitian. Untuk memperjelas, disusun tabel komparatif yang menampilkan kelebihan, keterbatasan, dan efektivitas tiap protokol keamanan.
6. Menyajikan Hasil Penelitian
   Sintesis literatur disajikan dalam bentuk laporan yang dijabarkan pada bagian pembahasan, sehingga memberikan gambaran komprehensif mengenai evolusi serangan *session hijacking* dan inovasi teknik pencegahannya.

4. HASIL DAN PEMBAHASAN

4.1. Metode Serangan yang Sering Digunakan pada Serangan *Session Hijacking*

Hasil studi literatur yang pertama diketahui bahwa terdapat 10 jurnal terkait *session hijacking* yang menggunakan metode serangan MITM *attack* [11-14], [20-21], [23-24], [28-29]. MITM *attack* adalah serangan keamanan di mana seorang penyerang menyisipkan diri di antara dua pihak yang

berkomunikasi secara tidak sah, memungkinkan penyerang untuk mengintip, mengubah, atau mencuri informasi yang ditransmisikan di antara keduanya.

Selain metode serangan MITM *attack*, sebanyak enam jurnal terkait *session hijacking* yang menggunakan metode serangan XSS [8], [11-12], [19], [21], [29]. XSS adalah serangan keamanan di mana penyerang menyisipkan kode berbahaya ke dalam situs web yang akan dieksekusi oleh pengguna lain, memungkinkan penyerang untuk mencuri informasi, mengubah tampilan halaman, atau mengalihkan pengguna ke situs berbahaya.

Metode populer ketiga yang digunakan dalam penyerangan sesi adalah *sniffing*. Dimana sebanyak empat jurnal yang menggunakan membahas mengenai serangan *sniffing* [9-11], [21], [28]. *Sniffing* adalah proses memantau dan menyadap lalu lintas data yang berjalan di jaringan, memungkinkan seseorang untuk mengakses informasi yang dikirimkan antara perangkat dan mengumpulkan data sensitif seperti kata sandi, kuki (*cookies*), atau data pribadi tanpa sepengetahuan pengguna yang bersangkutan.

Tabel 1. Jenis Serangan yang Digunakan dalam *Session Hijacking*

No.	Jenis Serangan	Jumlah	Referensi
1.	*Man-in-the-middle Attack*	10	[11], [12], [13], [14], [20], [21], [23], [24], [25], [28], [29]
2.	*Cross-Site Scripting (XSS)*	6	[8], [11], [12], [19], [21], [29]
3.	*Sniffing*	4	[10], [11], [21], [28]
4.	*ARP Spoofing*	2	[26], [28]
5.	*Session Fixation*	2	[9], [29]
6.	*Session Stealing*	2	[25], [29]
7.	*SSL Stripping*	2	[23], [30]
8.	*Brute Force*	1	[21]
9.	*Distributed Denial Of Service (DDOS)*	1	[27]
10.	*Injecting Attack Signals*	1	[15]
11.	*Network-Based Attack*	1	[22]
12.	*Web Session Hijacking*	1	[17]

4.2. Protokol dan Penerapan Keamanan pada *Session Hijacking*

Dalam mengatasi serangan pembajakan sesi atau *session hijacking*, terdapat berbagai langkah dan protokol yang dapat diterapkan. Beberapa protokol yang digunakan ditunjukkan dalam tabel di bawah ini.

Tabel 2. Jenis Protokol untuk Mengatasi *Session Hijacking*

No.	Protokol	Jumlah	Referensi
1.	*Secure Socket Layer (SSL) / Transport Layer Security (TLS)*	7	[8], [9], [11], [13], [21], [23], [27]
2.	*HTTPS*	6	[9], [11], [20], [21], [23], [27]
3.	*Session ID*	3	[10], [19], [29]
4.	*One-Time Cookie*	2	[14], [28]
5.	*Browser Fingerprint*	2	[14], [29]
6.	*Session Packet Inspector*	1	[12]
7.	*Client And Server Side System*	1	[30]
8.	*Secure Shell*	1	[21]
9.	*Physical Layer Key Agreement with User Introduced Randomness (PHY UIR)*	1	[24]
10.	*Supervisory Control and Data Acquisition (SCADA)*	1	[22]

Berdasarkan Tabel 2, terdapat dua protokol yang paling banyak digunakan yaitu *Secure Socket Layer* (SSL) dan *Transport Layer Security* (TLS) sebanyak tujuh artikel dan *Hypertext Transfer Protocol Secure* (HTTPS) sebanyak enam artikel. Protokol SSL/TLS adalah protokol keamanan yang digunakan untuk mengenkripsi komunikasi antara klien dan *server* dengan cara mengikat saluran komunikasi dan menyediakan enkripsi dari ujung ke ujung data sehingga seluruh pesan terenkripsi. Protokol ini cukup efisien untuk mencegah serangan seperti MITM dan dapat melindungi integritas data dengan baik.

Namun, SSL/TLS ini memiliki kinerja yang lambat, biaya yang cukup mahal, dan bergantung pada sertifikat digital.

Pada urutan kedua jenis protokol yang sering digunakan adalah *Hypertext Transfer Protocol Secure* (HTTPS). HTTPS adalah protokol keamanan yang digunakan untuk mengenkripsi data yang dikirimkan antara klien dan *server* sehingga menghalangi peretas untuk mencuri informasi sensitif seperti *session* ID dan kredensial pengguna selama proses transmisi data.

Pada Tabel 2 juga ditampilkan berbagai protokol lain yang digunakan dalam mengatasi serangan *session hijacking*. Mulai dari *Session* ID*, one-time cookie* hingga *Supervisory Control and Data Acquisition* (SCADA). Setiap protokol tersebut memiliki kelebihan dan kekurangan masing-masing yang menjadi pertimbangan bagi pengguna dalam penerapan protokol tersebut. Kelebihan dan kekurangan setiap protokol juga menentukan protokol mana yang paling efektif dan efisien untuk digunakan.

4.3. Protokol yang dinilai Efektif dan Efisien

Berdasarkan analisis yang dilakukan, protokol yang paling ampuh dalam mengatasi serangan *session hijacking* adalah *one-time cookie* [14], [28]. Bentuk dari metode penyerangan yang dapat diatasi *one-time cookie* yaitu MITM *attack, sniffing,* dan ARP *spoofing*. *One-time cookie* adalah metode autentikasi sesi HTTP yang menggunakan kuki unik yang hanya dapat digunakan satu kali dalam sesi tertentu.

Protokol *one-time cookie* menggantikan HTTP dengan HTTPS untuk meningkatkan keamanan sesi HTTP dengan menyediakan kuki unik yang hanya dapat digunakan sekali, sehingga mengurangi risiko serangan perekaman sesi dan melindungi integritas komunikasi antara klien dan *server* [28]. Hasil studi sebelumnya juga menunjukkan bahwa penggunaan *one-time cookie* mampu meningkatkan keamanan sesi web secara signifikan dengan dampak minimal terhadap performa sistem [32], [33]. Dacosta et al.. (2012) melaporkan bahwa *overhead* tambahan yang dihasilkan sangat kecil, sekitar <6 ms per permintaan, sehingga nyaris tidak memengaruhi pengalaman pengguna [32]. Studi lain oleh Modi (2020) memperkuat temuan ini dengan menyatakan bahwa *one-time cookie* dapat menjaga integritas sesi dan memberikan *throughput* yang lebih tinggi dibanding HTTPS penuh, sekaligus mempertahankan kinerja hampir setara dengan *cookie* tradisional [33]. Dengan demikian, klaim efektivitas protokol ini didukung tidak hanya secara konseptual, tetapi juga melalui pengujian empiris pada implementasi nyata.

Penerapan protokol *one-time cookie* juga dapat digabungkan dengan *browser fingerprint* melibatkan mengikat lapisan jaringan dan lapisan aplikasi melalui *server proxy* terbalik. Melalui mekanisme ini, sistem dapat mendeteksi perubahan pada *browser* pengguna, sehingga penyerang tidak mendapatkan akses ilegal ke sesi yang sedang berlangsung. Dengan menggunakan kuki unik yang hanya dapat digunakan sekali dan menganalisis sidik jari *browser* untuk memverifikasi identitas pengguna, protokol ini meningkatkan keamanan dan melindungi integritas komunikasi antara klien dan *server* [14].

Selain protokol keamanan *one-time cookie*, protokol yang dapat mengatasi penyerangan sesi adalah *session-packet inspector*, yaitu perangkat lunak yang memantau, menganalisis, dan menyaring paket data dalam sesi jaringan untuk mengidentifikasi potensi ancaman, masalah jaringan, atau aktivitas tidak sah. Metode serangan *session hijacking* yang dapat diamankan oleh *session-packet inspector* adalah MITM *attack* dan XSS, dengan penerapan melalui mekanisme yang mencegah intrusi pada jaringan *ad-hoc* nirkabel menggunakan agen bergerak (*mobile agent*) [12]. Protokol ini mampu mencegah peretasan HTTP *post* maupun kuki terenkripsi dalam jaringan nirkabel terbuka, meningkatkan keamanan data dan privasi pengguna, serta menawarkan teknik deteksi intrusi yang lebih efektif dan responsif dibanding model lainnya.

5. KESIMPULAN

Berdasarkan studi literatur, serangan *session hijacking* umumnya dilakukan melalui metode MITM, XSS, dan *sniffing*. Untuk mengatasi ancaman ini, diperlukan penerapan protokol keamanan yang tepat. SSL/TLS dan HTTPS menjadi protokol yang paling banyak digunakan karena mampu mengenkripsi komunikasi antara klien dan *server*, sehingga meminimalkan risiko MITM sekaligus menjaga integritas data. HTTPS juga memastikan kuki yang dikirimkan tetap terenkripsi sehingga dapat mencegah pencurian sesi.

Selain itu, protokol *one-time cookie* dinilai sangat efektif karena menggunakan kuki unik yang hanya dapat digunakan sekali dalam satu sesi, sehingga memperkecil peluang pembajakan sesi.

Penerapan *session-packet inspector* juga terbukti membantu karena mampu memantau dan menganalisis paket data yang melintas di jaringan, sehingga potensi ancaman dapat diidentifikasi lebih cepat dan tepat. Dengan dukungan teknik deteksi intrusi yang memadai, kombinasi protokol dan mekanisme ini mampu meningkatkan perlindungan data dan privasi pengguna, terutama pada jaringan nirkabel terbuka.

REFERENSI

[1] O. A. Кравчук, "Information technologies in the development of web resources," *Vìsnik Hmel'nic'kogo nacìonal'nogo unìversitetu*, vol. 341, no. 5, pp. 334–337, Oct. 2024, doi: 10.31891/2307-5732-2024-341-5-49.

[2] K. Ranjan and C. Sreenivasa, "Securing *user* sessions," Apr. 23, 2020.

[3] D. A. Abdulmonim and Z. H. Muhamad, "Comparative Study Between the OSI Model and the TCP/IP Model: Architecture and Protocols in Computer Networking Systems," *Int. J. Eng. Comput. Sci.*, vol. 13, no. 08, pp. 26358–26372, Aug. 2024, doi: 10.18535/ijecs/v13i08.4880.

[4] J. M. Kizza, "Authentication," in *Guide to Computer Network Security*, Springer Int. Publishing, 2024, pp. 215–238, doi: 10.1007/978-3-031-47549-8_10.

[5] H. Chavoshi, A. Salasi, O. Payam, and H. Khaloozadeh, "Man-in-the-Middle Attack Against a Network Control System: Practical Implementation and Detection," pp. 1–6, Oct. 2023, doi: 10.1109/itms59786.2023.10317671.

[6] L. V. Cherckesova, E. Revyakina, E. Roshchina, and V. Porksheyan, "The development of countermeasures against session hijacking," *E3S Web Conf.*, vol. 531, p. 03019, Jan. 2024, doi: 10.1051/e3sconf/202453103019.

[7] D. Tank and A. Dalvi, "A Novel Approach to Prevent Session Hijacking Attack," *Int. J. Comput. Appl.*, vol. 181, no. 14, pp. 28–30, Sep. 2018, doi: 10.5120/IJCA2018917798.

[8] W. Burgers, R. Verdult, and M. Van Eekelen, "Prevent Session Hijacking by Binding the Session to the Cryptographic Network Credentials," *LNCS 8208*, [n.d.].

[9] P. De Ryck, L. Desmet, F. Piessens, and W. Joosen, "SecSess: Keeping your session tucked away in your browser," in *Proc. ACM Symp. Appl. Comput.*, Apr. 2015, pp. 2171–2176, doi: 10.1145/2695664.2695764.

[10] S. S. Manivannan and E. Sathiyamoorthy, "A prevention model for session hijack attacks in wireless networks using strong and encrypted session ID," *Cybern. Inf. Technol.*, vol. 14, no. 3, pp. 46–60, 2015, doi: 10.2478/cait-2014-0032.

[11] G. Singh et al., "A-BLAZE 2015: International Conference on Futuristic Trends on Computational Analysis and Knowledge Management," *Amity Univ.*, [n.d.].

[12] S. K. L., "Session-packet inspector mobile agent to prevent encrypted cookies and HTTP POST hijacking in MANET," *J. Eng. Sci. Technol.*, vol. 11, no. 12, 2016.

[13] P. Kamal, "State of the Art Survey on Session Hijacking," 2016, doi: 10.17406.

[14] M. Doshi, N. Patel, N. Patel, and Y. Shah, "A review on prevention for session hijacking using one-time cookies," *Res. Gate*, 2017. [Online]. Available: https://www.researchgate.net/publication/342391782

[15] Q. Hu and G. P. Hancke, "A Session Hijacking Attack on Physical Layer Key Generation Agreement," 2017.

[16] O. Koksal and B. Tekinerdogan, "Feature-driven domain analysis of session layer protocols of internet of things," in *Proc. IEEE 2nd Int. Congr. Internet Things (ICIOT)*, 2017, pp. 105–112, doi: 10.1109/IEEE.ICIOT.2017.19.

[17] N. N. S. Ismail, M. N. M. Warip, S. J. Elias, and R. B. Ahmad, "A preliminary review on web session hijacking," *Int. J. Eng. Technol. (UAE)*, vol. 7, no. 3, pp. 124–129, 2018, doi: 10.14419/ijet.v7i3.15.17515.

[18] M. Bilal, M. Asif, and A. Bashir, "Assessment of secure OpenID-based DAAA protocol for avoiding session hijacking in web applications," *Secur. Commun. Netw.*, 2018, doi: 10.1155/2018/6315039.

[19] A. K. Sinha and S. Tripathy, "CookieArmor: Safeguarding against cross-site request forgery and session hijacking," *Secur. Privacy*, vol. 2, no. 2, e60, 2019, doi: 10.1002/spy2.60.

[20] IEEE, "2018 IEEE Int. Conf. Electro/Information Technology (EIT)," *Oakland Univ.*, May 2018.

[21] . Kumar Baitha and S. Vinod, "Session hijacking and prevention technique," *Int. J. Eng. Technol.*, vol. 7, no. 2.6, pp. 193, 2018, doi: 10.14419/ijet.v7i2.6.10566.

[22] A. Kleinmann et al., "Stealthy deception attacks against SCADA systems," *Lect. Notes Comput. Sci.*, vol. 10683, pp. 93–109, 2018, doi: 10.1007/978-3-319-72817-9_7.

[23] Md. S. Hossain, A. Paul, Md. H. Islam, and M. Atiquzzaman, "Survey of the protection mechanisms to the SSL-based session hijacking attacks," *Netw. Protoc. Algorithms*, vol. 10, no. 1, pp. 83, 2018, doi: 10.5296/npa.v10i1.12478.

[24] Q. Hu, B. Du, K. Markantonakis, and G. P. Hancke, "A session hijacking attack against a device-assisted physical layer key agreement," [n.d.].

[25] R. R. Katta and B. P. Valluri, "United States Patent Topic-of-the-week: Detecting Browser Fingerprint Changes During," 2019.

[26] Y. B. Choi, Y. L. Loo, and K. Lacroix, "Cookies and sessions: A study of what they are, how they can be stolen and a discussion on security," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, 2019. [Online]. Available: www.ijacsa.thesai.org

[27] M. M. Naeem, I. Hussain, and M. M. S. Missen, "A survey on registration hijacking attack consequences and protection for

session initiation protocol (SIP)," *Comput. Netw.*, vol. 175, 2020, doi: 10.1016/j.comnet.2020.107250.

[28] N. Modi, "Comparative analysis of session features in session hijacking and performance improvement using OTC," 2020.

[29] T. Singh and Meenakshi, "Prevention of session hijacking using token and session ID reset approach," *Int. J. Inf. Technol. (Singapore)*, vol. 12, no. 3, pp. 781–788, 2020, doi: 10.1007/s41870-020-00486-w.

[30] M. Ahmad Jonas, M. Shohrab Hossain, R. Islam, H. S. Narman, and M. Atiquzzaman, "An intelligent system for preventing SSL stripping-based session hijacking attacks," [n.d.].

[31] Moh. Nazir and R. Sikmumbang, *Metode Penelitian*, Ghalia Indonesia, 2009.

[32] I. Dacosta, S. Chakradeo, M. Ahamad, and P. Traynor, "One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens," *ACM Transactions on Internet Technology*, vol. 12, no. 1, pp. 1–24, 2012, doi: 10.1145/2220352.2220353.

[33] N. Modi, "Comparative analysis of session features in session hijacking and performance improvement using OTC," *International Journal of Scientific Research & Engineering Trends*, vol. 6, no. 2, pp. 972–979, 2020.