Vol. 1 No. 1 Januari 2026
E-ISSN :-

SNIFFING JARINGAN PADA WEBSITE BERPROTOCOL HTTP (STUDI KASUS PADA WEBSITE SISTEM AKADEMIK SMA WAHIDIYAH TRENGGALEK)

Misbachuddin Zuhdi
Program Studi Informatika, Fakultas Teknik, Universitas Wahidiyah
Arekmbatu@gmail.com

ABSTRAK

Dalam era digital, keamanan informasi di situs web sangat penting karena banyaknya data sensitif yang dipertukarkan online. Ancaman terhadap keamanan semakin kompleks dengan berbagai metode serangan, termasuk sniffing jaringan, yang memantau dan merekam paket data di jaringan. Penelitian ini mengeksplorasi teknik sniffing jaringan menggunakan Wireshark untuk menemukan username dan password yang dikirim melalui protokol HTTP. Wireshark digunakan untuk menangkap dan menganalisis data jaringan, fokus pada bagaimana sniffing dapat mengakses informasi sensitif seperti username dan password. Metodologi mencakup pemilihan situs web target, pengaturan lingkungan pengujian, dan penerapan teknik sniffing. Hasil penelitian menunjukkan sniffing jaringan efektif untuk menemukan data sensitif melalui HTTP dan mengungkapkan kerentanannya terhadap serangan sniffing. Penelitian ini menegaskan pentingnya menggunakan protokol yang aman seperti HTTPS dan merekomendasikan penerapannya untuk melindungi data sensitif. Wireshark terbukti sebagai alat yang berguna dalam analisis keamanan jaringan.

Kata Kunci: Sniffing Jaringan, Wireshark, Keamanan Web, Username dan Password, Protokol HTTP, HTTPS, Analisis Keamanan

ABSTRACT

In the digital age, information security on websites is critical due to the large amount of sensitive data exchanged online. Threats to security are increasingly complex with various attack methods, including network sniffing, which monitors and records data packets on the network. This research explores network sniffing techniques using Wireshark to discover usernames and passwords sent over the HTTP protocol. Wireshark is used to capture and analyze network data, focusing on how sniffing can access sensitive information such as usernames and passwords. The methodology includes the selection of target websites, setting up the test environment, and applying sniffing techniques. The results show network sniffing is effective for discovering sensitive data over HTTP and reveals its vulnerability to sniffing attacks. This research emphasizes the importance of using secure protocols such as HTTPS and recommends their implementation to protect sensitive data. Wireshark proved to be a useful tool in network security analysis.

Keywords: Network Sniffing, Wireshark, Web Security, Username and Password, HTTP Protocol, HTTPS, Security Analysis Translated with DeepL.com (free version)

PENDAHULUAN

Dalam ekosistem yang terus berkembang, keamanan informasi di situs web sangatlah penting. Ancaman terhadap keamanan tersebut menjadi semakin kompleks seiring dengan meningkatnya jumlah metode pembobolan yang tidak sah. Salah satu teknik yang menjadi perhatian utama dalam konteks sniffing jaringan. Sniffing jaringan memungkinkan pengamatan dan perekaman paket data yang mengalir melalui jaringan, memungkinkan akses terhadap informasi sensitif yang dikirim dan diterima oleh website. Dalam konteks ini, Wireshark, sebagai alat analisis jaringan yang mapan, memiliki peran krusial dalam mendeteksi dan menganalisis potensi kebocoran data serta kerentanan yang mungkin dimanfaatkan oleh serangan sniffing. Dalam lingkup ini, Wireshark muncul sebagai alat analisis jaringan yang penting, menyediakan kemampuan untuk menangkap dan menganalisis paket data yang berpotensi mengungkapkan informasi sensitif seperti username dan password. Namun, upaya untuk mengidentifikasi, menganalisis, dan mengatasi kelemahan keamanan dalam protocol autentikasi, khususnya dalam protocol HTTP, merupakan tantangan yang signifikan. Menurut (Cantika et al., 2019), Wireshark memiliki beberapa fitur termasuk display filter language. Display filter language sendiri digunakan untuk memfilter dan menganalisis data dalam paket jaringan yang ditangkap. TCP (Transmission Control Protocol) adalah salah satu protocol utama dalam suite protocol

internet (TCP/IP) yang digunakan untuk mengatur bagaimana data dikirimkan dan diterima antara perangkat-perangkat yang terhubung dalam jaringan komputer, yang banyak dan kemampuan me-reka ulang sebuah aliran pada sesi TCP. Paket sniffier diartikan sebuah tool yang berkemampuan menahan dan melakukan pencatatan terhadap traffic data dalam jaringan. Tujuan sniffer sendiri digunakan untuk berbagai tujuan, termasuk pemecahan masalah jaringan, analisis keamanan, debugging aplikasi jaringan dan pemantauan kinerja jaringan. Dengan menganalisis packet sniffer, administrator jaringan dapat memahami bagaimana data bergerak di jaringan, mengidentifikasi masalah koneksi, mendeteksi serangan keamanan, atau mengevaluasii performa aplikasi. Selama terjadi aliran data dalam jaringan packet sniffer dapat menangkap protocol data unit (PDU), melakukan decoding serta analisis terhadap isi paket. Decoding sendiri merupakan proses untuk mengubah informasi yang terenskripsi, tersembunyi, atau terkompresi ke dalam format yang dapat dibaca atau dimengerti. Proses ini umumnya terjadi dalam berbagai konteks, termasuk dalam bidang komunikasi, komputasi dan teknologi informasi. PDU (Protocol Data Unit) adalah istilah yang digunakan dalam dunia jaringan computer untuk menyebut unit data yang dikirimkan atau diterima melalui jaringan, Dimana setiap lapisan protocol memiliki PDU tersendiri. PDU adalah cara standar untuk mengatur dan menggambarkan bagaimana data diatur dan ditransmisikan dalam sebuah jaringan.

METODE PENELITIAN

Penelitian ini menggunakan metode deskriptif, yang sesuai untuk memberikan penjelasan terperinci terhadap suatu peristiwa. Metode deskriptif diaplikasikan untuk menjelaskan pengujian protocol HTTP, yang dilakukan berulang kali untuk memastikan keberhasilan serangan menggunakan teknik Man in the Middle dalam memperoleh informasi login seperti username dan password dari website sistem akademik. Selama simulasi penyerangan, penulis akan menganalisis setiap paket data dan mengamati perubahan pada protocol HTTP yang melewati jaringan server lokal. Dalam pengamatan terhadap paket data, teridentifikasi beberapa komponen yang menjadi fokus penelitian. Berikut 6 komponen dari paket data:

1.Time Menjelaskan format waktu packet yang tertangkap.
2. Source Merupakan IP sumber dari suatu packet data.
3. Destination Merupakan IP tujuan kemana suatu packet data akan diteruskan.
4. Protocol Merupakan jenis protocol apa yang digunakan. 5. Packet Length Merupakan Panjang dari suatu packet data yang digunakan.
6. Info Merupakan Info lebih lanjut mengenai suatu packet.

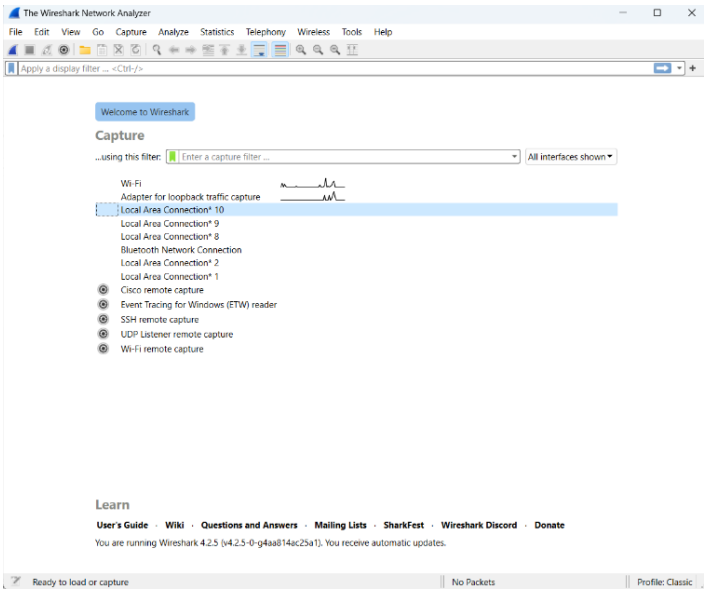
Pelaksanaan penelitian Sniffing jaringan pada website dilaksanakan pada tanggal 4 November 2023 sampai dengan 5 Juli 2024 dan untuk lokasi penelitian bertempat di Fakultas Teknik Universitas Wahidiyah Kota Kediri

No	Kegiatan	Bulan																													
		November				Desember				Januari				Februari				Maret				April				Mei				Juni				Juli			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Analisa Masalah																																				
2	Analisa Kebutuhan																																				
3	Pengumpulan Data																																				
4	Penulisan																																				
5	Desain Sistem																																				
6	Implementasi																																				
7	Pengujian Sistem																																				

Tabel 1. 1 Waktu penelitian

HASIL DAN PEMBAHASAN

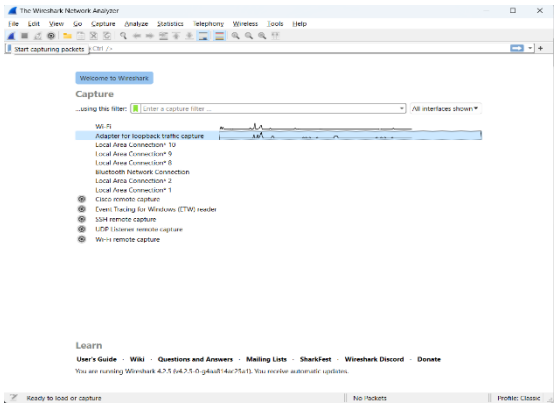
1. Memilih jalur internet mana yang akan di



Gambar 1. 1 Memilih jalur sniffing

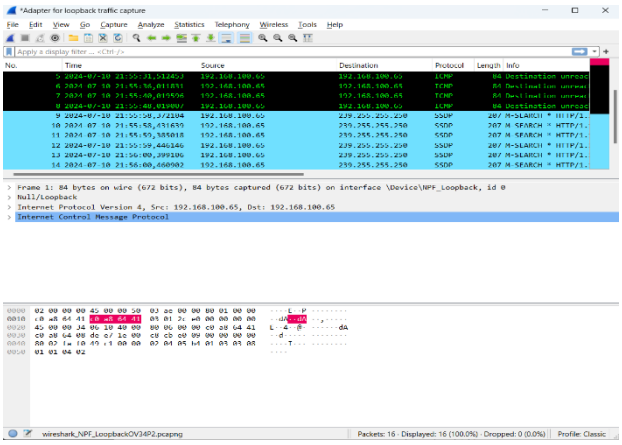
sniffing

Pada gambar 1.1 dapat dilihat Perangkat lunak Wireshark menampilkan beberapa antarmuka, yang merupakan daftar jalur yang tersedia pada perangkat untuk terhubung ke jaringan internet. Ini dapat diamati pada Gambar 4. 2, di mana ada garis yang mirip dengan yang ditemukan dalam diagram listrik. Jika ada gelombang pada garis-garis ini, dapat dipastikan bahwa ada aktivitas di sana. Tujuan dari aktivitas ini adalah bahwa komunikasi data sedang dilakukan pada antarmuka yang terhubung ke jaringan internet. Oleh karena itu, penting untuk memastikan penggunaan antarmuka yang benar sebelum memulai proses pemantauan menggunakan perangkat lunak wireshark.



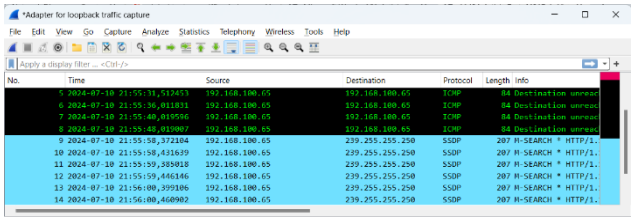
Setelah antarmuka yang sesuai telah ditentukan, lanjutkan dengan mengklik tombol dengan ikon yang menyerupai sirip hiu dan diberi label "Start Capturing Packet". Setelah mengklik, proses sniffing pada antarmuka yang dipilih akan segera dimulai. Di karenakan yang di teliti adalah jaringan localhost maka saya memakai interface “Adapter for loopback traffic capture”.

2. Memulai untuk menjalankan sniffing
   Setelah memulai proses sniffing, tiga jendela ditampilkan di layar awal. Jendela-jendela ini termasuk jendela daftar paket, jendela detail paket, dan jendela byte paket. Masing-masing jendela ini menyajikan serangkaian informasi yang berbeda. Tampilan awal Wireshark ketika memulai proses sniffing adalah sebagai berikut:



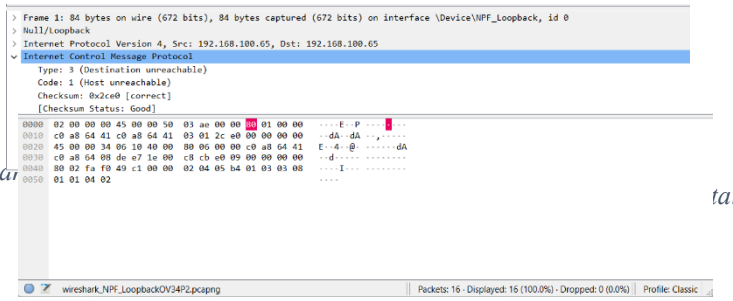
Pada gambar 1. 4 di atas adalah tampilan dari packet list, packet list sediri merupakan tampilan yang menunjukan hasil data tangkapan pada jaringan internet dalam format

Gambar 1. 4 Packet list



tabel yang disajikan dalam baris. Setiap baris akan berisi informasi tentang data paket seperti somber paket (source), tujuan (destination), protokol (protokol), panjang paket (packet lengt), dan data informasi paket (info).

Pada gambar 1. 5 memiliki fungsi yaitu menampilkan



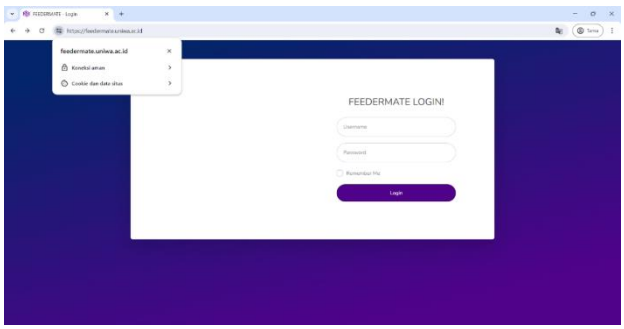
Ga *tail*

beberapa informasi mengenai protokol-protokol dari baris paket data yang dipilih pada tampilan packet list. Data tersebut disajikan secara horizontal dan juga berhirarki.

Padagambar 4. 6 di atas memiliki Fungsi yaitu untuk menampilkan data mentah (Raw) dari paket data yang dipilih di tampilan packet list. Data mentah (Raw) ini ditampilkan dalam format heksadesimal dan mencakup 16 byte heksadesimal dan 16 byte ASCII.

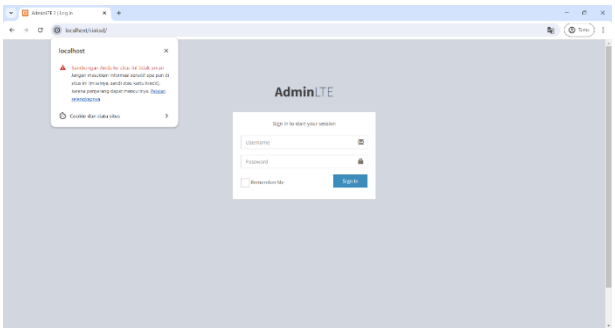
Gambar 1. 6 Packet bytes

3. Membuka website berprotocol HTTP
   Dikarenakan website sistem akademik Universitas Wahidiyah (Feedermate) sudah menggunakan protocol HTTPS, maka pada penelitian ini menggunakan website sistem akademik yang di buat sendiri dan masih menggunakan protocol HTTP.



Gambar 1. 7 Website feedermate system akademik universitas wahidiyah

Pada gambar 1. 7 diatas yaitu menampilkan website feedermate Sistem akademik Universitas Wahidiyah yang

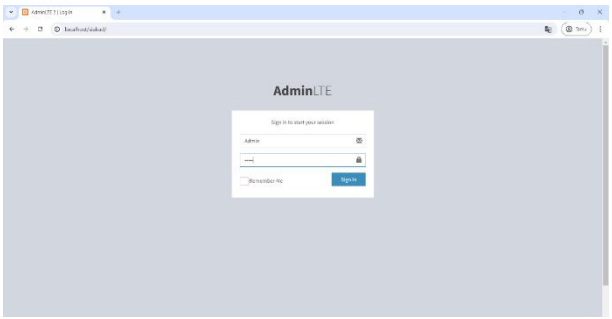


Gambar 1. 8 Website localhost siakad SMA Wahidiyah Trenggalek

sudah menggunakan HTTPS dan sudah memiliki sebuah "*Secure Sockets Layer*" (SSL) dan website tersebut di nyatakan aman dari serangan *sniffing*.

Pada penelitian ini penulis menggunakan sebuah website sistem akademik sekolah SMA Wahidiyah Trenggalek yang memiliki protocol HTTP buatan penulis sendiri menggunakan server localhost. Pada gambar 1. 8 diatas dapat dilihat bahwasannya website tidak aman atau not secure yang terdapat disamping alamat website, setiap komunikasi data yang terjadi pada website ini tidak dienkripsi.

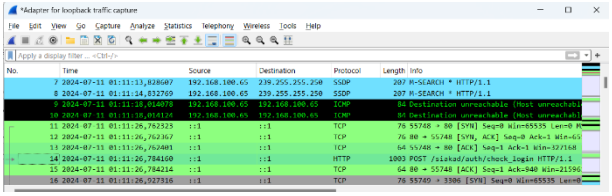
4. Melakukan login dan filter
   Setelah mengetahui website siakad tersebut tidak aman, maka akan dilakukan login dengan memasukkan username dan password. Berikut akun yang akan di gunakan untuk login:
   Username : Admin
   Password : Admin_717



Gambar 1. 9 Login website

Pada gambar 4. 9 di atas terlihat bahwasanya website berhasil login. Kembali lagi ke perangkat Wireshark, saat melakukan tombol klik Saat menekan tombol login, Wireshark telah merekam komunikasi yang terjadi.

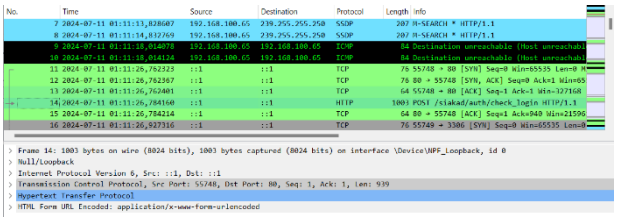
Pada jaringan internet. Sekarang, cukup saring kolom filter untuk mendapatkan protokol HTTP dan akan



Gambar 1. 10 Filter protocol HTTP

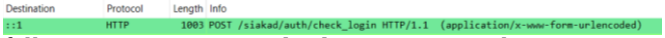
ditampilkan pada tampilan packet list seperti yang terlihat pada Gambar 1. 10

Kemudian pada gambar 4. 11 menampilkan beberapa IP di *Destination* dan ada sebuah aksi yang terjadi pada kolom *Info* seperti "POST", dikarenakan menggunakan interface localhost maka IP untuk localhost sendiri ditampilkan pada jendela packet detail. Kemudian pada data POST sendiri memiliki fungsi yaitu untuk mengirimkan data dari sumber ke halaman lain untuk diproses. Disini penulis hanya focus pada data POST yang dilakukan saat login pada website.



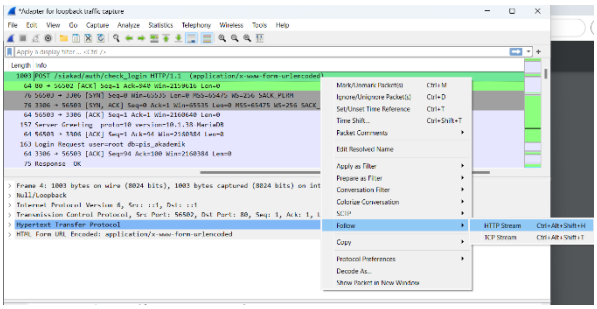
Gambar 1. 11 Info dan destination

5. Hasil dari sniffing
   Setelah memilih *interface,* menjalankan *website localhost,* dan melakukan display filter untuk memfilter data yang ingin ditampilkan. Maka selanjutnya yaitu mencari informasi yang tepat pada kolom *Info.* Disini penulis focus ke protocol HTTP dengan data POST dan HTTP/1.1. HTTP/1.1 ini artinya webserver hanya menerima 1 koneksi kepada tiap objek yang ditampilkan pada website



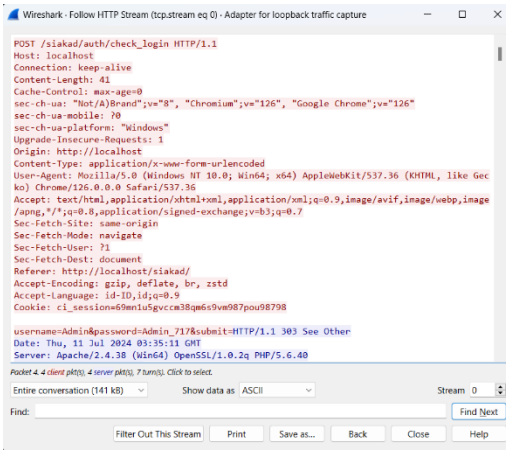
Gambar 1. 12 Info

Setelah itu lakukan *follow* HTTP *Stream* pada data POST tersebut seperti pada gambar 1. 13. Fungsi dari HTTP scream sendiri



adalah untuk melihat langsung apa yang terjadi pada protocol HTTP saat melakukan pengiriman data saat login.

Gambar 1. 13 Tata letak HTTP stream



Gambar 1. 14 Hasil dari HTTP stream

Dapat terlihat pada gambar 4. 14, ada blok tulisan berwarna merah dan biru. Ini bukan hanya sekadar gaya, tetapi memiliki makna. Tulisan dengan blok merah adalah data yang dikirimkan dari website ke webserver, sedangkan tulisan dengan blok biru adalah respon dari webserver yang akan ditampilkan di website. Dalam aliran HTTP ini, dapat dilihat host dari website, tipe konten, User-Agent, dan lain-lain. Data-data ini cukup penting karena dari situ peneliti dapat mengetahui sistem operasi dan arsitektur yang digunakan saat mengakses website.

Yang terpenting adalah jika username dan password yang dimasukkan saat login tadi terbuka untuk dilihat. Username dan password yang dimasukkan pada saat login tadi bisa di lihat seperti pada gambar 1. 15.

```
username=Admin&password=Admin_717&submit=HTTP/1.1 303 See Other
```

Gambar 1. 15 Terbukanya username dan password

Bisa terlihat dengan jelas di sana bahwa username dan password yang dimasukkan saat login tadi terpampang. Inilah sebabnya mengapa sniffing pada komunikasi data dalam jaringan internet berprotokol HTTP sangat berbahaya, karena selain mudah dilakukan, hasil yang diperoleh berupa data yang sangat sensitif. Sniffing dalam penelitian ini hanya mencakup koneksi antara perangkat peneliti dan webserver.

6. Solusi untuk mencegah serangan packet sniffing
   Setelah melakukan studi, penulis telah merumuskan beberapa saran solusi untuk meningkatkan keamanan dari ancaman serupa yang diteliti.

- Menggunakan HTTPS Bahkan di Localhost: Meskipun localhost umumnya dianggap aman karena hanya diakses secara lokal, menerapkan HTTPS tetap merupakan praktik yang baik. HTTPS akan mengenkripsi lalu lintas data antara server dan browser, bahkan dalam lingkungan localhost XAMPP. Ini akan melindungi data dari potensi serangan sniffing yang mungkin terjadi secara tidak sengaja di dalam jaringan local.
- Enkripsi Data Login: Selain menggunakan HTTPS, enkripsi data login secara khusus dapat memberikan lapisan keamanan tambahan. Data login seperti username dan password harus dienkripsi sebelum disimpan di database dan juga saat ditransmisikan melalui jaringan. Hal ini akan memastikan bahwa meskipun penyerang berhasil mencegat data, mereka tidak akan dapat membacanya tanpa kunci dekripsi yang sesuai.
- Penggunaan Password yang Kuat: Kebijakan password yang kuat harus diterapkan bahkan di lingkungan localhost XAMPP. Pengguna harus didorong untuk menggunakan password yang panjang, kompleks, dan unik, yang terdiri dari kombinasi huruf besar dan kecil, angka, dan simbol.
- Two-Factor Authentication (2FA): Menerapkan 2FA di localhost XAMPP mungkin tidak umum, tetapi dapat menjadi langkah tambahan untuk meningkatkan keamanan. 2FA akan mengharuskan pengguna untuk memberikan informasi tambahan selain password, seperti kode yang dikirimkan melalui email atau aplikasi otentikasi, untuk mengakses sistem.
- Pembatasan Akses: Batasi akses ke localhost XAMPP hanya untuk pengguna yang berwenang. Gunakan fitur keamanan yang disediakan oleh XAMPP, seperti otentikasi dasar atau .htaccess, untuk membatasi akses ke direktori atau file tertentu.

PENUTUP

Kesimpulan

Setelah melakukan penelitian dan analisis terkait sniffing jaringan pada website untuk menemukan username dan password menggunakan Wireshark, dapat ditarik beberapa kesimpulan sebagai berikut:

- Efektivitas Wireshark dalam Sniffing Jaringan : Wireshark terbukti menjadi alat yang efektif dalam melakukan sniffing jaringan untuk menganalisis lalu lintas data. Dengan menggunakan Wireshark, penulis dapat memonitor dan menangkap paket data yang melintasi jaringan, serta menganalisis informasi yang terkait dengan username dan password yang dikirimkan melalui protokol HTTP.

- Kerentanannya Protokol HTTP : Hasil penelitian menunjukkan bahwa protokol HTTP yang tidak terenkripsi rawan terhadap serangan sniffing. Data yang dikirimkan melalui HTTP dapat dengan mudah ditangkap dan dibaca oleh pihak ketiga yang memiliki akses ke jaringan yang sama termasuk informasi sensitif seperti username dan password.
- Pentingnya Penggunaan Protokol yang Aman : Penelitian ini menggarisbawahi pentingnya penggunaan protokol yang aman seperti HTTPS untuk melindungi data sensitif dalam transmisi data antara client dan server. HTTPS menggunakan enkripsi TLS/SSL yang mengamankan komunikasi data sehingga mengurangi risiko penangkapan data oleh pihak yang tidak berwenang.
- Peran Wireshark dalam Keamanan Jaringan : Wireshark bukan hanya alat untuk penyerangan tetapi juga dapat digunakan untuk tujuan keamanan jaringan yang positif, seperti pemantauan dan analisis lalu lintas untuk mengidentifikasi potensi kerentanan dan memperbaiki sistem keamanan jaringan.

Saran

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, penulis memberikan beberapa saran sebagai berikut:

- Mengimplementasikan HTTPS : Website sebaiknya selalu menggunakan HTTPS untuk mengenkripsi data yang ditransmisikan antara pengguna dan server. Penggunaan HTTPS sangat penting untuk melindungi informasi sensitif seperti username dan password dari kemungkinan penangkapan data oleh pihak ketiga.
- Meningkatkan Kesadaran Keamanan : Pengguna dan pengembang website perlu meningkatkan kesadaran tentang risiko keamanan yang terkait dengan transmisi data melalui jaringan. Edukasi mengenai pentingnya enkripsi data dan penerapan praktik keamanan yang baik dapat membantu mencegah serangan sniffing.
- Menggunakan Alat Keamanan Jaringan Lainnya : Selain Wireshark, pengguna disarankan untuk menggunakan berbagai alat keamanan jaringan lainnya untuk melakukan pemantauan dan pengujian keamanan secara menyeluruh. Alat-alat seperti Snort, Nmap, dan Metasploit dapat digunakan untuk melengkapi analisis dan mendeteksi potensi kerentanan jaringan.
- Memperbarui dan Memelihara Sistem : Pastikan bahwa semua perangkat dan aplikasi yang digunakan dalam jaringan diperbarui secara berkala untuk mengatasi kerentanan keamanan. Pembaruan perangkat lunak dan sistem operasi adalah langkah penting untuk melindungi jaringan dari ancaman yang berkembang. 5) Melakukan Penetrasi Testing Secara Berkala : Organisasi atau pengembang website sebaiknya melakukan penetrasi testing secara berkala untuk mengevaluasi keamanan sistem mereka. Penetrasi testing dapat membantu mengidentifikasi potensi celah keamanan dan menerapkan langkah-langkah perbaikan sebelum potensi penyerang dapat mengeksploitasi kelemahan tersebut.

DAFTAR PUSTAKA

1) Abdillah, M. A., Yudhana, A., & Fadil, A. (2020). Sniffing Pada Jaringan WiFi Berbasis Protokol 802.1x Menggunakan Aplikasi Wireshark. J-SAKTI (Jurnal Sains Komputer Dan Informatika), 4(1), 1. https://doi.org/10.30645/j-sakti.v4i1.181
2) Cantika, I., Huda, F. N., & Saputra, D. M. (2019). SNIFFING JARINGAN MENGGUNAKAN WIRESHARK. 5–8.
3) Desi, A. (2018). Analisis Keamanan Website Terhadap Sniffing Procces Pada Jaringan Nirkabel Menggunakan Aplikasi Wireshark (STUDI KASUS : SIMAK UNISMUH).
4) Fitri, M. O. (2021). Awebserver Sebagai Alternatif Pengganti Xampp Pada Platform Android. Teknosains: Media Informasi Sains Dan Teknologi, 15(2), 245. https://doi.org/10.24252/teknosains.v15i2.2002
5) Hanipah, R., & Dhika, H. (2020). Analisa Pencegahan Aktivitas Ilegal Didalam Jaringan Dengan Wireshark. DoubleClick: Journal of Computer and Information Technology, 4(1), 11. https://doi.org/10.25273/doubleclick.v4i1.5668
6) Huzaeni, F., Gunawan, I., Cahya, D., Yanti, M., & Krisdayanti, N. (2021). Analisis Keamanan

Data Pada Website Dengan Wireshark. JES (Jurnal Elektro Smart), 1(1), 13–17.

7) Ibrahim, M. M. (2020). Analisis Keamanan Jaringan pada Fasilitas Internet (Wifi) Kantor Pemerintahan Kota Batam terhadap Serangan Packet Sniffing.
8) Luthfansa, Z. M., & Rosiani, U. D. (2021). Pemanfaatan Wireshark untuk Sniffing Komunikasi Data Berprotokol HTTP pada Jaringan Internet. Journal of Information Engineering and Educational Technology, 5(1), 34–39. https://doi.org/10.26740/jieet.v5n1.p34-39
9) Majid, A., & Purwanto, T. D. (2021). Analisis Dan Monitoring Sniffing Paket Data Jaringan Lokal Bps Sumseldengan Network Analyzer Wireshark. Seminar Hasil Penelitian Vokasi (SEMHAVOK), 03(1), 102–109.
10) Nurdiana, F. R., Gunawan, I., Viollita, R. C., Faizal, M. A., & Nurcahyadi, D. (2021). Analisis Keamanan Jaringan Wifi Menggunakan Wireshark. JES (Jurnal Elektro Smart), 1(1), 10–13.
11) Putu, I., Pratama, A. E., Adhika Dharmesta, P., & Informasi, T. (2019). IMPLEMENTASI WIRESHARK DALAM MELAKUKAN PEMANTAUAN PROTOCOL JARINGAN (Studi Kasus : Intranet Jurusan Teknologi Informasi Universitas Udayana). Jurnal Mantik Penusa, 3(1), 94.
12) Sugiyatno. (2019). Perancangan Clustering Database Server Untuk. XVIII, 281–289.
13) Tamsir, T. A., Eggy Saputra, Kundari, & Muhammad Tio Farizky. (2023). Analisis Paket Icmp Website Universitas Binadarma Menggunakan Wireshark. STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer, 2(2), 55–60. https://doi.org/10.55123/storage.v2i2.1956
14) Ubaedila, I., Nurdiawan, O., Wijaya, Y. A., & Sidik, J. (2022). Layanan Jaringan Menggunakan Metode Sniffing Berbasis Wireshark. INFORMATICS FOR EDUCATORS AND PROFESSIONAL : Journal of Informatics, 6(1), 95. https://doi.org/10.51211/itbi.v6i1.1697