Keamanan Basis Data

Dosen Pengampu: Bapak Antoni Haikal, S.T., M.T

IMPLEMENTED AND CONFIGURING HTTP - HTTPS CONNECTION

Submitted by:
Nisrina Amelia Putri
4332101006

Politeknik Negeri Batam
Teknik Informatika
Rekayasa Keamanan Siber

DAFTAR ISI

PENDAHULUAN

A. LATAR BELAKANG

Dengan perkembangan Internet yang sangat pesat, keamanan menjadi semakin penting bagi seluruh pengguna. Untuk beberapa situs seperti keuangan dan sistem pembayaran online, ini lebih penting daripada kinerja. Banyak penelitian telah dilakukan untuk meningkatkan keamanan data. Beberapa mekanisme diusulkan untuk meningkatkan keamanan sistem penyimpanan yang terhubung ke jaringan. Freeman dan Miller memberikan sistem file arsitektur untuk menjamin keamanan data pengguna menggunakan enkripsi end-to-end. Skema aman untuk melindungi sistem penyimpanan yang terhubung ke jaringan terhadap berbagai jenis serangan menggunakan kriptografi yang kuat diusulkan di HTTP adalah protokol paling populer untuk mentransfer dokumen melalui Internet dan Secure HTTP (HTTPS) adalah protokol untuk mentransfer data HTTP sensitif melalui SSL (Secure Socket Layer).

Protokol HTTP adalah blok bangunan utama dari Web, namun tidak memberikan jaminan kerahasiaan atau integritas apa pun. HTTPS melindungi komunikasi jaringan terhadap penyadapan dan perusakan dengan menjalankan HTTP di atas protokol kriptografi seperti Secure Socket Layer (SSL) dan penerusnya Transport Layer Security (TLS), yang memungkinkan pembentukan saluran komunikasi dua arah terenkripsi. Selain kerahasiaan dan integritas, HTTPS juga memastikan autentikasi, karena klien dan server dapat membuktikan identitasnya dengan menghadirkan sertifikat yang ditandatangani oleh otoritas sertifikasi tepercaya. HTTPS semakin diakui sebagai landasan keamanan aplikasi web dari waktu ke waktu dan secara rutin digunakan oleh semakin banyak situs web, hingga volume rata-rata lalu lintas web terenkripsi telah melampaui volume rata-rata lalu lintas tidak terenkripsi menurut data dari Mozilla.

Pada praktikum ini, akan dilakukan beberapa pengujian untuk membandingkan antara koneksi HTTP dan HTTPS, baik dari segi kecepatan, keamanan, dan efisiensi dari seluruh koneksi yang dilakukan tes uji.

B. TUJUAN

Adapun tujuan pembuatan laporan ini ialah untuk:

1. Mengetahui Dasar Teori pada HTTP dan HTTPS.
2. Dapat Mengimplementasikan penggunaan HTTP dan HTTPS.
3. Mengetahui Perbandingan Performa yang dihasilkan oleh HTTP dan HTTPS.
4. Dapat membuktikan implementasi dan performa yang ada pada HTTP dan HTTPS.
5. Dapat melakukan testing HTTP dan HTTPS secara langsung.
6. Sebagai sumber untuk memenuhi nilai tugas Mata Kuliah Keamanan Basis Data.

C. LANDASAN TEORI

Apa itu HTTP?

HTTP adalah singkatan dari *Hypertext Transfer Protocol*, yaitu protokol, perintah, atau sintaks yang ditentukan untuk menyajikan informasi, yang digunakan untuk mentransfer data melalui jaringan. Sebagian besar informasi yang dikirimkan melalui Internet, termasuk konten situs web dan panggilan API, menggunakan protokol HTTP. Ada dua jenis utama pesan HTTP yaitu *request* dan *respond*.

HTTP *request* dihasilkan saat pengguna berinteraksi dengan properti web. Misalnya, jika pengguna mengklik *hyperlink*, browser akan mengirimkan serangkaian permintaan *"HTTP GET"* untuk konten yang muncul di halaman tersebut. Jika seseorang mencari di *Google* "Apa itu HTTP?" dan sebuah artikel muncul di hasil pencarian, ketika pengguna mengklik tautan, browser pengguna akan membuat dan mengirim serangkaian permintaan HTTP untuk mendapatkan informasi yang diperlukan untuk merender halaman.

Permintaan HTTP hanyalah serangkaian baris teks yang mengikuti protokol HTTP. Permintaan GET mungkin terlihat seperti ini:

```
GET /hello.txt HTTP/1.1
User-Agent: curl/7.63.0 libcurl/7.63.0 OpenSSL/1.1.l zlib/1.2.11
Host: www.example.com
Accept-Language: en
```

Apa itu HTTPS?

S dalam HTTPS berarti "Secure". HTTPS menggunakan TLS (atau SSL) untuk mengenkripsi permintaan dan respons HTTP. TLS menggunakan teknologi yang disebut kriptografi kunci publik. Ada dua kunci, yaitu kunci publik dan kunci private. Kunci publik dibagikan dengan perangkat klien melalui sertifikat SSL server. Saat klien membuka koneksi dengan server, kedua perangkat menggunakan kunci publik dan pribadi untuk menyetujui kunci baru, yang disebut kunci sesi, untuk mengenkripsi komunikasi lebih lanjut di antara mereka. Semua permintaan dan respons HTTP kemudian dienkripsi dengan kunci sesi ini, sehingga siapa pun yang menyadap komunikasi hanya dapat melihat rangkaian karakter acak, bukan teks biasa.

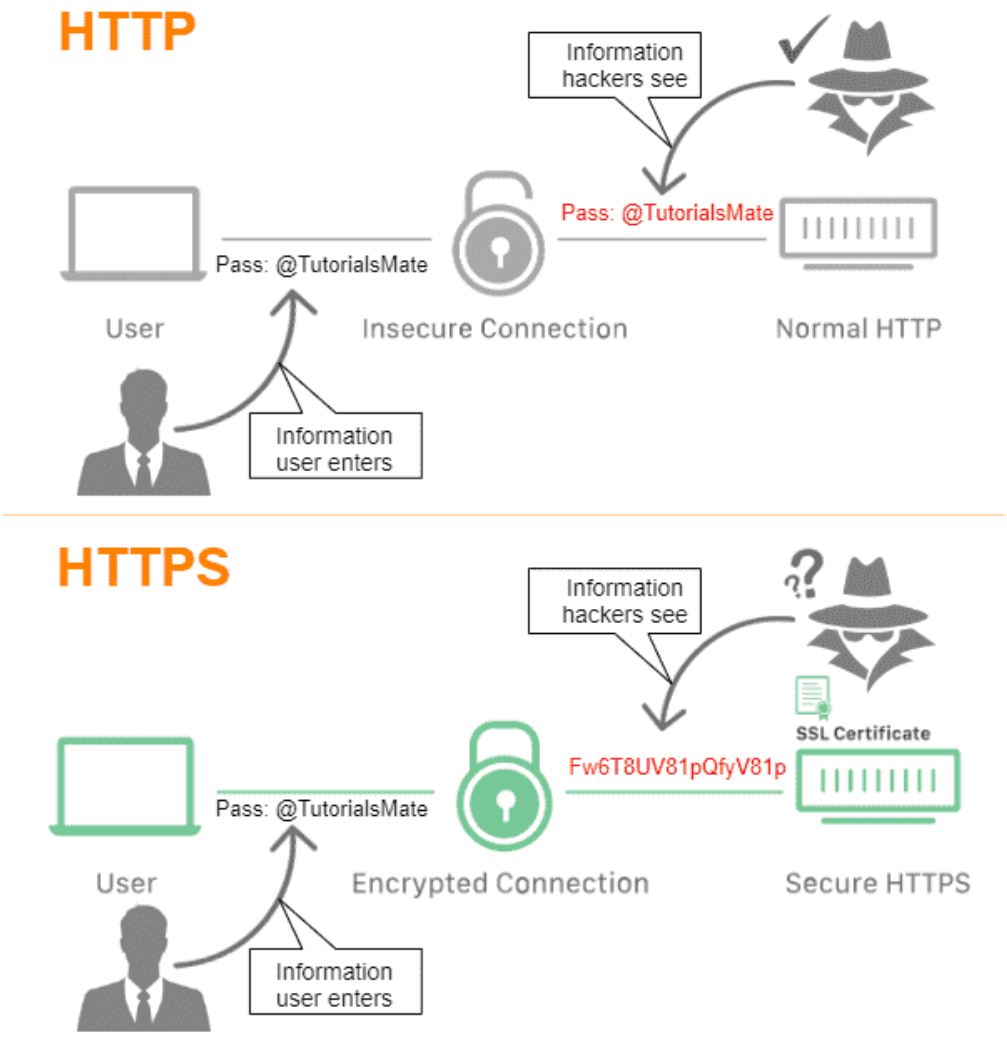
Di HTTP, tidak ada verifikasi identitas karena didasarkan pada prinsip kepercayaan. Arsitek HTTP tidak harus membuat keputusan untuk secara implisit mempercayai semua server web. Namun di Internet modern, autentikasi sangat penting.

Pada HTTP, permintaan disajikan dalam bentuk plain text sedangkan pada HTTPS, permintaan akan dienkripsi terlebih dahulu. Dapat terlihat seperti:

```
t8Fw6T8UV81pQfyhDkhebbz7+oiwldr1j2gHBB3L3RFTRsQCpaSnSBZ78Vme+DpDVJPvZdZUZHpzbbcqmSW1+3xXGsERH
g9YDmpYk0VVDiRvw1H5miNieJeJ/FNUjgH0BmVRWII6+T4MnDwmCMZUI/orxP3HGwYCSIvyzS3MpmmSe4iaWKCOHQ==
```

Apa perbedaan antara HTTP dan HTTPS?

- HTTP adalah singkatan dari HyperText Transfer Protocol dan HTTPS adalah singkatan dari HyperText Transfer Protocol Secure.
- Di HTTP, URL dimulai dengan "http://" sedangkan URL dimulai dengan "https://"
- HTTP menggunakan nomor port 80 untuk komunikasi dan HTTPS menggunakan 443
- HTTP dianggap tidak aman dan HTTPS aman
- HTTP Bekerja di Application Layer dan HTTPS bekerja di Transport Layer
- Di HTTP, Enkripsi tidak ada dan Enkripsi ada di HTTPS seperti yang dibahas di atas
- HTTP tidak memerlukan sertifikat apa pun dan HTTPS memerlukan Sertifikat SSL
- Kecepatan HTTP lebih cepat dari HTTPS dan kecepatan HTTPS lebih lambat dari HTTP
- HTTP tidak meningkatkan peringkat pencarian sementara HTTPS meningkatkan peringkat pencarian.
- HTTP tidak menggunakan tagar data untuk mengamankan data, sedangkan HTTPS akan memiliki data sebelum mengirimkannya dan mengembalikannya ke keadaan semula di sisi penerima.



Source: TutorialsMate

PEMBAHASAN

A. IMPLEMENTASI KONEKSI TERENKRIPSI KE DATABASE

Untuk mengimplementasikan koneksi terenkripsi ke database, perlu memasang SSL pada perangkat database yang digunakan (saat ini PostgreSQL). Setelah berhasil memasang SSL, maka perlu untuk melakukan konfigurasi untuk sertifikat SSL tersebut ke agar koneksi terenkripsi pada database dapat berjalan. Berikut langkah-langkah yang dapat dilakukan untuk mengimplementasikan koneksi terenkripsi pada PostgreSQL:

1. Generate Certificate SSL yang baru pada server

```
root@nisr:~# openssl req -new -text -out server.req
...+.......+..+...+.+.........+..++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+*.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.........+..........
+..+.+.....+.+...+.....+......+..........+......+........+...+..........+...+............+..+...+
....+........+.+......+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

...+...+..+...++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*........+........
......+..........+...+.....+....+.........+.....+.+.........+..+...+.+...+.....+...+.........+...
......+......+.+..+.................+..+....+...+...+..+................+......+.+......+........+.
...+...+.....+...+..........+.....+......+.+.....+....++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++*..+...+...+..........+.........+..+......+....+...+........+...+.+...+....
.+......+......+....+......+.........+.......................+.....+....+.................+........
....+.+.....+.+...+..........+.........+.+.....+.........+.+..+.......+...+.........+......+....
.+.......+..+.+............+.....+...+.............+...........+...+.+...............+.+..........
.+.+..+......+................+....+................+........+...+....+........+...+......+.+...+..
.........+................+........+.......+............+...+...+..+....+........+......+.+.......
............+.....+.......+........+.+.....+.+...........+.........+.....+.....+.....+....+.......
..+...........+.........+...+...+......+.+.........+.....+...+...+.........+...+..........+...+..
.+..................+....+...+.....+.+............+..+.......................+.+.........+..+.......
..+...+............+................+.+.....+...........+...+...+..........+.....+.+.....+........
..+.+.....+......+...+....+...........+......+.+...+..+.......+...+...+..+.......+................
......+............+...+.....+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
```

Hapus passphrase untuk Automatic Server Start Up.

```
root@nisr:~# openssl rsa -in privkey.pem -out server.key && rm privkey.pem
Enter pass phrase for privkey.pem:
writing RSA key
```

Modifikasi file menjadi self-signed dan generate .key dan .crt file

```
root@nisr:~# openssl req -x509 -in server.req -text -key server.key -out server.crt
Warning: No -copy_extensions given; ignoring any extensions in the request
```

Pindahkan file SSL ke directory yang diinginkan, biasanya file SSL akan disimpan pada directory data cluster PostgreSQL. Lakukan perintah berikut:

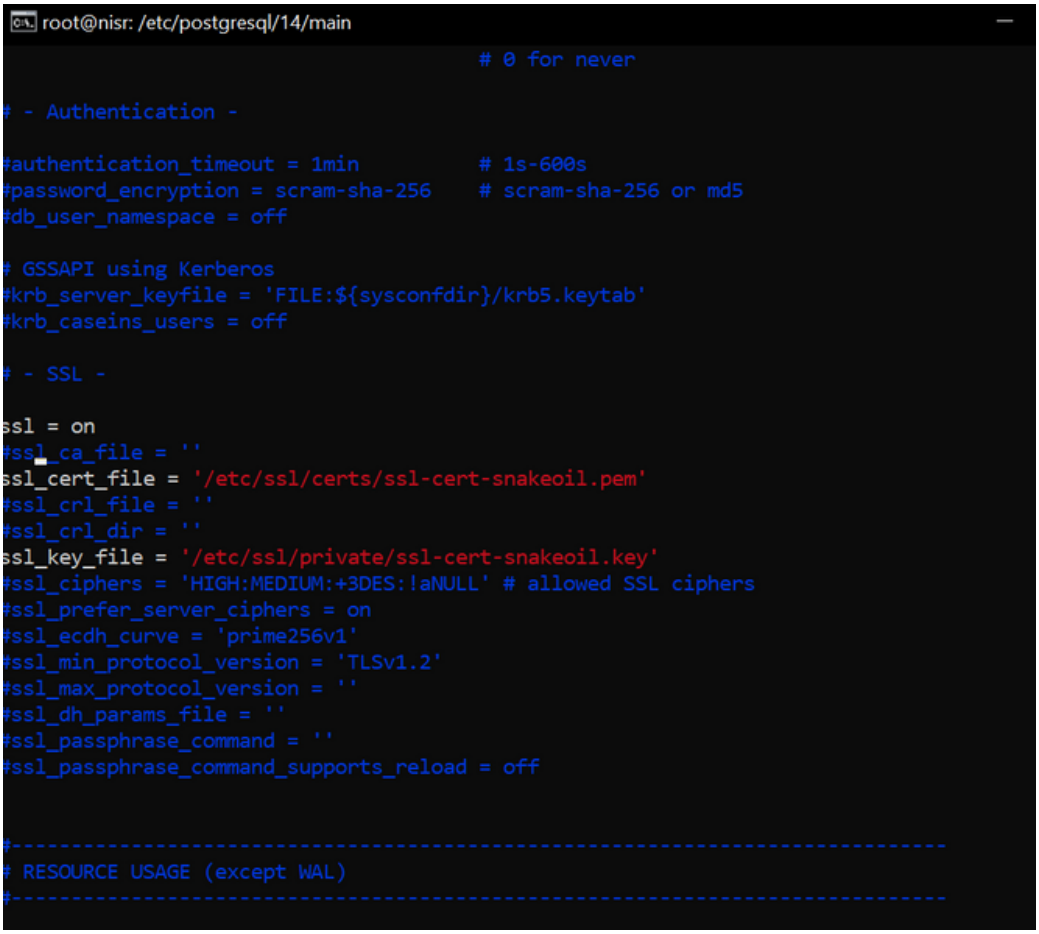
```
root@nisr:~# cp server.key server.crt $PGDATA
```

Ubah permissions dari file SSL agar file tersebut tidak dapat diakses selain oleh user PostgreSQL.

```
root@nisr:~# chmod og-rwx server.key
```

2. Setting SSL di file postgresql.conf

Dapat dilakukan dengan cara masuk ke direktori /etc/postgresql/14/main lalu mengetikan vim postgrestql.conf. Pastikan ssl sudah on.



Restart database agar konfigurasi dapat berjalan dengan baik.

```
root@nisr:/etc/postgresql/14/main# systemctl restart postgresql
```

3. Cek Status SSL

Masuk ke dalam user yang digunakan untuk mengakses database.

```
root@nisr:/etc/postgresql/14/main# su - postgres
```

Dapat melihat status ssl dengan perintah:

```
postgres@nisr:~$ psql -c "show ssl;"
Password for user postgres:
 ssl
-----
 on
(1 row)
```

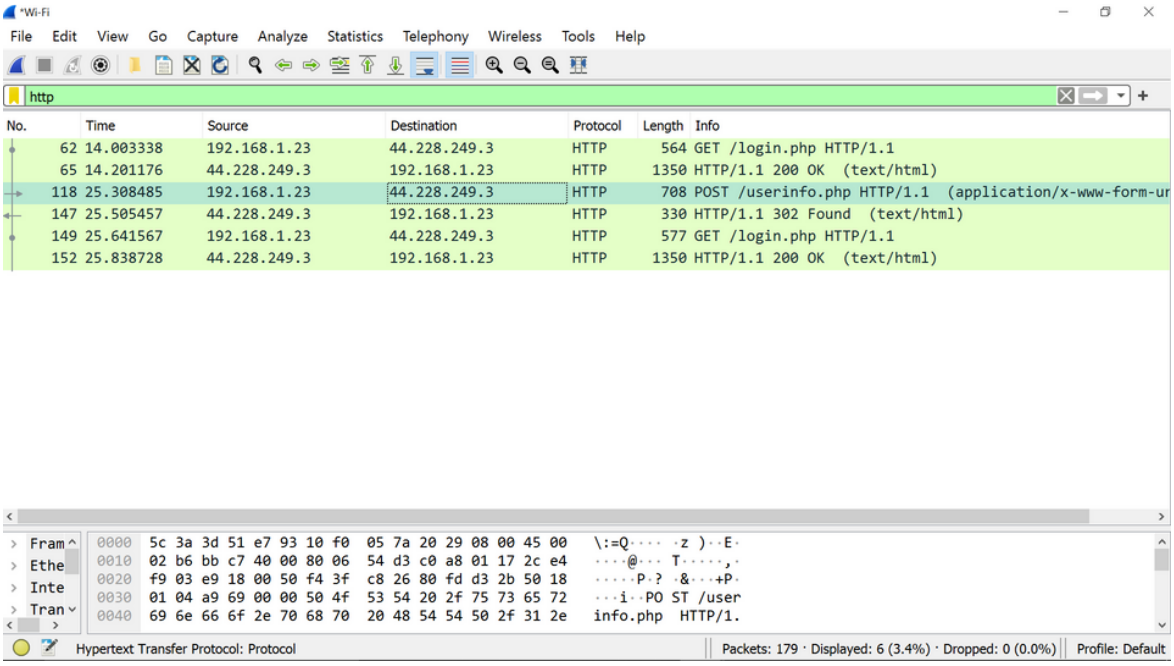
Jika output dari query tersebut adalah on, maka SSL sudah aktif dan siap digunakan.

B. PERBADINGAN KONEKSI YANG TIDAK DAN DENGAN ENKRIPSI

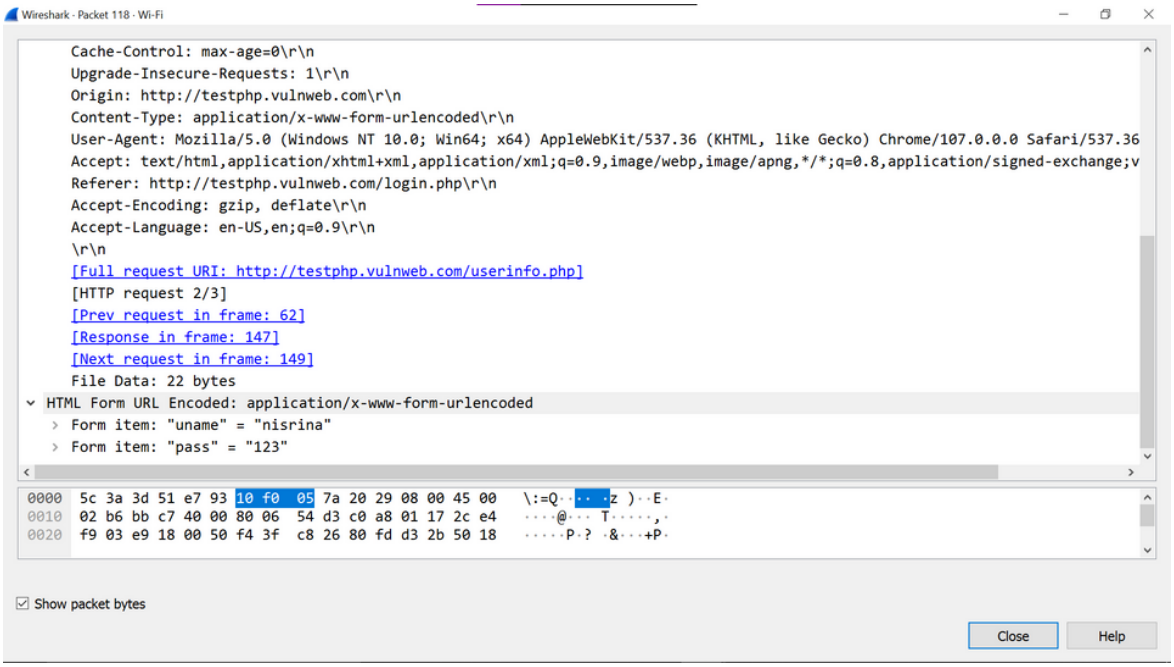
Membandingkan koneksi yang tidak terenkripsi (contoh HTTP) dengan koneksi yang sudah terenkripsi (contoh HTTPS) perlu menggunakan metode seperti GET, POST, dsb. Dalam mengidentifikasi perbedaaannya, dapat menggunakan tools bernama wireshark untuk melihat paket-paket yang sedang berjalan.

1. Koneksi HTTP

Untuk mempermudah pencarian protokol http, dapat langsung mencarinya pada kolom search. Lalu klik pada metode POST /userinfo.php untuk melihat detail paket yang masuk.

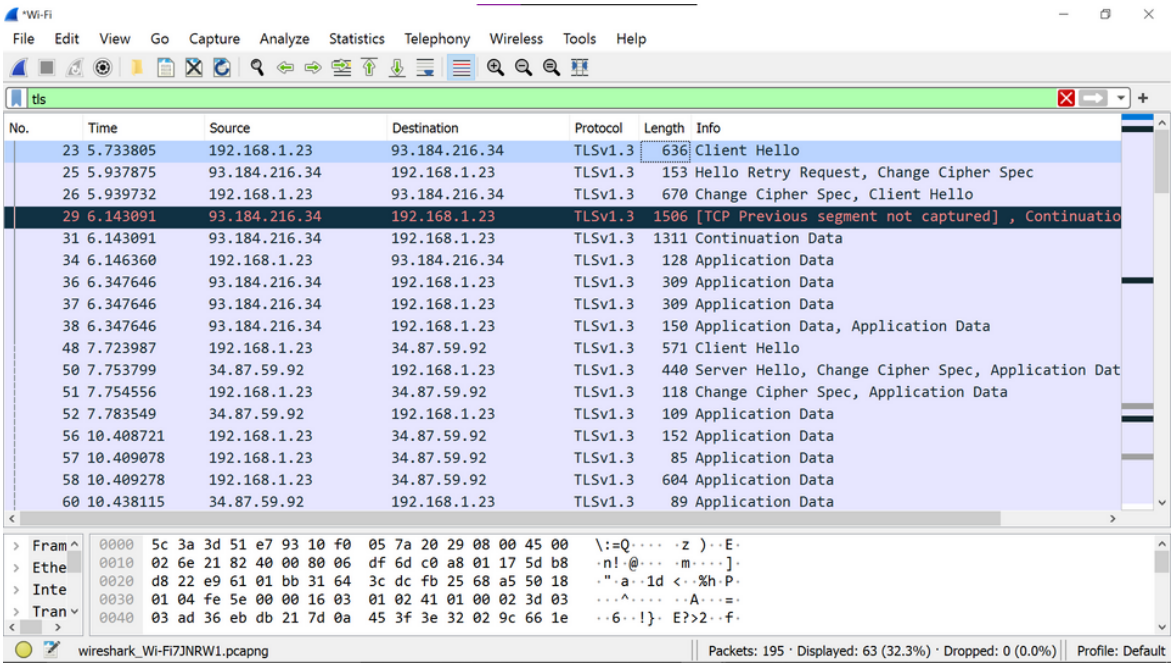


Perhatikan pada kolom HTML Form URL Encoded, tertulis plaintext username dan password untuk login (menggunakan website vulnweb.com untuk melakukan pengujian).

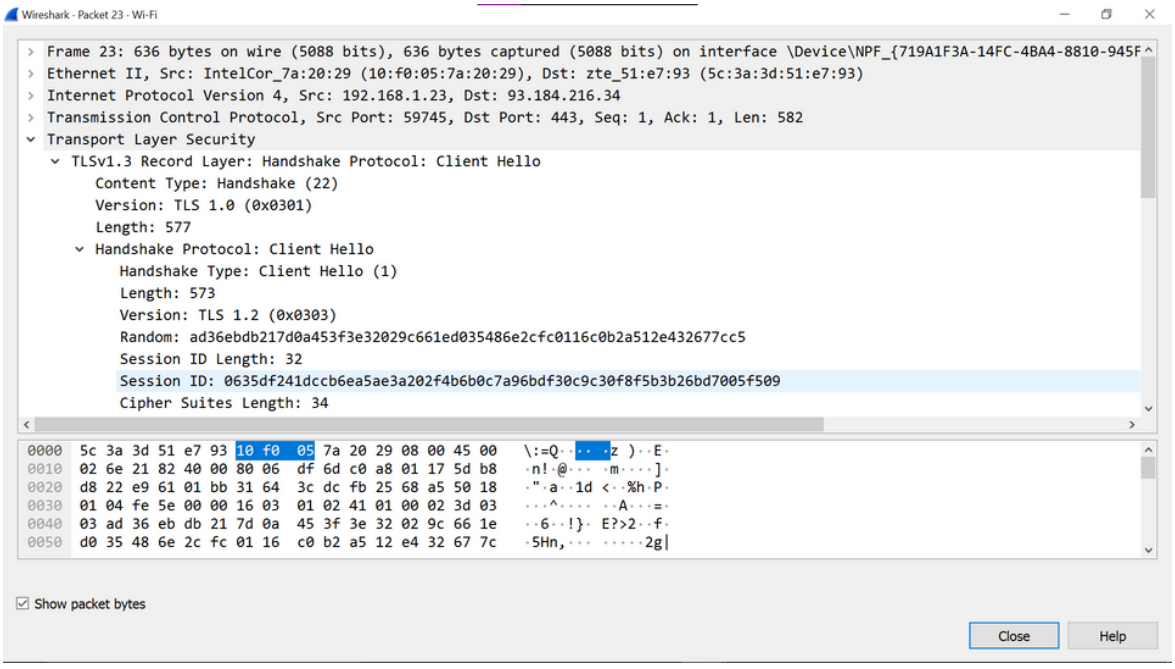


2. Koneksi HTTPS

Untuk mencari protokol HTTPS dapat mengetikan TLS pada kolom pencarian. Lalu perhatikan Paket CLient Hello sebagai penanda sedang melakukan login.



Terlihat bahwa seluruh informasi yang ada di dalamnya sudah terenkripsi yang menyebabkan wireshark tidak bisa membaca username dan password secara plaintext seperti pada HTTP protokol.



3. Analisis Singkat

Jadi, perbandingan antara protokol HTTP dan HTTPS terletak pada paket-paket yang dapat di-scanning. Pada HTTP, paket yang masuk berupa plaintext tanpa proteksi sedangkan pada HTTPS, paket sudah dienkripsi dengan berbagai jenis kriptografi yang menyebabkan lebih sulit untuk melakukan pengintaian pada protokol ini.

C. PERFORMA PENGGUNAAN KONEKSI TERENKRIPSI DENGAN 1000 HIT

Pada bagian performa, akan mencoba membandingkan antara kecepatan akses dari HTTP dan HTTPS ke database dengan 1000 hit. Untuk melihat performa pada postgres, dapat menggunakan pgbench.

1. Koneksi HTTPS

```
postgres@nisr:~$ pgbench -i -s 10 kambasdat
Password:
dropping old tables...
creating tables...
generating data (client-side)...
1000000 of 1000000 tuples (100%) done (elapsed 4.05 s, remaining 0.00 s)
vacuuming...
creating primary keys...
done in 7.36 s (drop tables 0.01 s, create tables 0.02 s, client-side generate 4.16 s, vacuum 1.17 s, primary keys 2.00 s).
postgres@nisr:~$
```

2. Koneksi HTTP

```
postgres@nisr:~$ pgbench -i -s 10 nisrina
Password:
dropping old tables...
NOTICE:  table "pgbench_accounts" does not exist, skipping
NOTICE:  table "pgbench_branches" does not exist, skipping
NOTICE:  table "pgbench_history" does not exist, skipping
NOTICE:  table "pgbench_tellers" does not exist, skipping
creating tables...
generating data (client-side)...
1000000 of 1000000 tuples (100%) done (elapsed 2.24 s, remaining 0.00 s)
vacuuming...
creating primary keys...
done in 3.97 s (drop tables 0.01 s, create tables 0.04 s, client-side generate 2.29 s, vacuum 0.66 s, primary keys 0.99 s).
postgres@nisr:~$
```

3. Analisis Singkat

Dari perbandingan performa kedua koneksi yaitu HTTP dan HTTPS, tidak terdapat perbedaan yang signifikan. Tetapi dapat terlihat bahwa kecepatan dari koneksi HTTP (yang tidak terenkripsi) lebih cepat daripada koneksi HTTPS (yang terenkripsi). Hal ini dapat disebabkan karena, pada jaringan yang terenkripsi, terdapat lebih banyak fitur guna mengenkripsi paket-paket yang masuk dan yang keluar sedangkan pada HTTP tidak diperlukan pengenkripsian. Hal ini sesuai dengan hukum bahwa keamanan akan berbanding terbalik dengan kenyamanan.

PENUTUP

A. KESIMPULAN

Pengimplementasian SSL pada produk digital perlu dilakukan untuk menghindari pengintaian data. Meskipun penggunaan HTTPS diperlukan untuk keamanan aplikasi web, database, dsb, tetapi hal ini bukanla satu-satunya cara untuk melakukan pengamanan. Karena kekurangan dalam penerapan TLS yang mendasari mungkin memiliki impor keamanan yang signifikan pada lapisan aplikasi.

Perbandingan antara protokol HTTP dan HTTPS terletak pada paket-paket yang dapat di-scanning. Pada HTTP, paket yang masuk berupa plaintext tanpa proteksi sedangkan pada HTTPS, paket sudah dienkripsi dengan berbagai jenis kriptografi yang menyebabkan lebih sulit untuk melakukan pengintaian pada protokol ini.

Dari perbandingan performa kedua koneksi yaitu HTTP dan HTTPS, tidak terdapat perbedaan yang signifikan. Tetapi dapat terlihat bahwa kecepatan dari koneksi HTTP (yang tidak terenkripsi) lebih cepat daripada koneksi HTTPS (yang terenkripsi). Hal ini dapat disebabkan karena, pada jaringan yang terenkripsi, terdapat lebih banyak fitur guna mengenkripsi paket-paket yang masuk dan yang keluar sedangkan pada HTTP tidak diperlukan pengenkripsian. Hal ini sesuai dengan hukum bahwa keamanan akan berbanding terbalik dengan kenyamanan.

DAFTAR PUSTAKA

Porter, Adrienne. et. all. 2017. Measuring HTTPS Adoption on the Web. Included in the Proceedings of the 26th USENIX Security Symposium August 16–18, 2017. Vancouver, BC, Canada ISBN 978-1-931971-40-9.

Franco et. all. Man-in-the-middle attack to the HTTPS protocol. Article in IEEE Security and Privacy Magazine · March 2009 DOI: 10.1109/MSP.2009.12 · Source: IEEE Xplore.

Shwan. et. all. 2017. Analysis of HTTP and HTTPS Usage on the University Internet Backbone Links. Journal of Industrial and Intelligent Information Vol. 2, No. 1, March 2014.

Stefano et. all. Postcards from the Post-HTTP World: Amplification of HTTPS Vulnerabilities in the Web Ecosystem. IEEE 2019.

Xubin, He. A Performance Analysis of Secure HTTP Protocol. IEEE, Department of Electrical and Computer Engineering Tennessee Technological University Cookeville, TN 38505, U.S.A

S. A. K. Tanoli, I. Khan, N. Rajatheva, T. Issariyakul, and T. Erke, “Trace-based analysis for campus-wide wireless LAN over advanced training system,” in Proc. First International Conference on Future Information Networks, 2009