

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini bertujuan untuk menganalisis kerentanan penggunaan OpenSSL pada server *localhost* yang menggunakan protokol HTTPS, namun mengalami kendala dengan sertifikat CA yang tidak valid. *Server* yang digunakan dalam penelitian ini dibangun menggunakan *Node.js*. Analisis dilakukan dengan metode *sniffing* untuk mengidentifikasi potensi ancaman serangan SSL *hijacking*.

4.2 KONFIGURASI SERVER MENGGUNAKAN OPENSLL DAN NODE.JS

Untuk memastikan keamanan komunikasi data pada *server*, konfigurasi HTTPS menggunakan OpenSSL dan *Node.js* menjadi pilihan yang tepat. Proses ini melibatkan pembuatan sertifikat SSL/TLS dengan OpenSSL dan pengaturan *server Node.js* untuk menggunakan sertifikat tersebut, sehingga koneksi antara klien dan *server* dapat terenkripsi dengan baik.

Menggunakan *Windows* sebagai sistem operasi, langkah-langkah konfigurasi dimulai dengan instalasi OpenSSL dan *Node.js*. Selanjutnya, sertifikat SSL/TLS yang dibuat menggunakan OpenSSL dan kemudian diterapkan pada *server Node.js*. Tahap ini memastikan bahwa data yang dikirimkan melalui jaringan tidak dapat dengan mudah diakses atau dimodifikasi oleh pihak yang tidak berwenang. Panduan ini akan menjelaskan langkah-langkah detail untuk menyelesaikan konfigurasi tersebut.

4.2.1 Instalasi OpenSSL

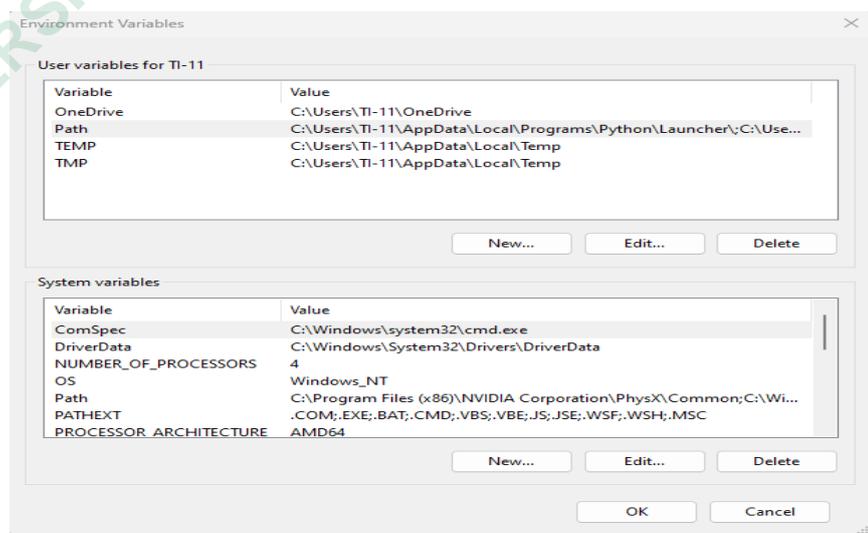
Langkah pertama adalah menginstal OpenSSL pada sistem operasi yang digunakan. Dalam penelitian ini, sistem operasi yang digunakan adalah *Windows* 11. Berikut langkah-langkahnya:

1. Kunjungi situs resmi OpenSSL dan unduh versi terbaru dari OpenSSL *Installer* untuk *Windows*.
 - a. Unduh OpenSSL v3.3.1 dapat dilihat pada Gambar 4.1.

Download Win32/Win64 OpenSSL		
Download Win32/Win64 OpenSSL today using the links below!		
File	Type	Description
Win64 OpenSSL v3.3.1 Light EXE MSI	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.3.1 (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.3.1 EXE MSI	217MB Installer	Installs Win64 OpenSSL v3.3.1 (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.3.1 Light EXE MSI	4MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v3.3.1 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.3.1 EXE MSI	175MB Installer	Installs Win32 OpenSSL v3.3.1 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.3.1 Light for ARM (EXPERIMENTAL) EXE MSI	6MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.3.1 for ARM64 devices (Only install this VERY EXPERIMENTAL build if you want to try 64-bit OpenSSL for Windows on ARM processors). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.3.1 for ARM (EXPERIMENTAL) EXE MSI	170MB Installer	Installs Win64 OpenSSL v3.3.1 for ARM64 devices (Only install this VERY EXPERIMENTAL build if you want to try 64-bit OpenSSL for Windows on ARM processors). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.2.2 Light EXE MSI	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.2.2 (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.2.2 EXE MSI	202MB Installer	Installs Win64 OpenSSL v3.2.2 (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.2.2 Light EXE MSI	4MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v3.2.2 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

Gambar 4. 1 Unduh OpenSSL

2. Jalankan *file installer* yang telah diunduh dan ikuti petunjuk instalasi.
3. *Set Environment Variables*
 - a. Setelah instalasi selesai, tambahkan *path* direktori *bin* OpenSSL ke *environment variables* sistem. Buka "*Control Panel*" > "*System and Security*" > "*System*" > "*Advanced system settings*" > "*Environment Variables*". Konfigurasi dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Menambahkan *Path Variable*

- b. Tambahkan *path* `C:\Program Files\OpenSSL-Win64\bin` (atau direktori instalasi yang sesuai) ke variabel *Path*.

4. Verifikasi Instalasi

- a. Buka *Command Prompt* dan ketik `openssl version` untuk memverifikasi bahwa OpenSSL telah terinstal dengan benar.

```
C:\Program Files\OpenSSL-Win64\bin>openssl version
OpenSSL 3.3.1 4 Jun 2024 (Library: OpenSSL 3.3.1 4 Jun 2024)
```

Gambar 4.3 Versi OpenSSL

- b. Pada Gambar 4.3 setelah memasukkan perintah `openssl version` pada *Command Prompt* maka akan muncul versi OpenSSL dan instalasi selesai.

4.2.2 Instalasi Node.js

Berikut adalah langkah-langkah untuk menginstal Node.js :

1. Kunjungi situs resmi Node.js dan unduh *installer* versi terbaru.
 - a. Unduh Node.js dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Unduh Node.js

2. Jalankan file *installer* yang telah diunduh dan ikuti petunjuk instalasi.

3. Verifikasi Instalasi : Buka *Command Prompt* dan ketikkan perintah `node -v` dan `npm -v` untuk memastikan Node.js dan *npm* telah terinstal dengan benar.

4.2.3 Pembuatan Sertifikat SSL

Setelah OpenSSL dan Node.js terinstal, langkah berikutnya adalah membuat sertifikat SSL *self-signed*:

1. Buka *Command Prompt* dan navigasikan ke direktori file untuk menyimpan sertifikat dan kunci.
2. Jalankan perintah berikut untuk membuat sertifikat *self-signed*:

```
a. openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365
```

- b. Ikuti petunjuk untuk mengisi detail sertifikat seperti negara, organisasi, dan nama domain. Setelah selesai, dua file akan dihasilkan: `key.pem` (*private key*) dan `cert.pem` (*public certificate*).

4.2.4 Konfigurasi Server Node.js

Setelah sertifikat SSL dibuat, langkah selanjutnya adalah mengkonfigurasi server Node.js untuk menggunakan HTTPS:

1. Membuat file server

Buat file *JavaScript* baru `server.js` dan tambahkan kode berikut:

```

JS server.js
C: > Windows > System32 > my-secure-server > JS server.js > ...
1  const fs = require('fs');
2  const https = require('https');
3  const express = require('express');
4  const bodyParser = require('body-parser');
5  const app = express();
6
7  // Sertifikat SSL
8  const privateKey = fs.readFileSync('localhost.key', 'utf8');
9  const certificate = fs.readFileSync('localhost.crt', 'utf8');
10 const credentials = { key: privateKey, cert: certificate };
11
12 // Middleware untuk mengurai data dari form POST
13 app.use(bodyParser.urlencoded({ extended: true }));
14
15 // Middleware untuk melayani file statis
16 app.use(express.static('public'));
17
18 // Route untuk halaman login
19 app.get('/', (req, res) => {
20   res.sendFile(__dirname + '/public/login.html');
21 });
22
23 // Route untuk memproses login
24 app.post('/login', (req, res) => {
25   const { username, password } = req.body;
26   // Proses login sederhana
27   if (username === 'admin' && password === 'password') {
28     res.send('Login berhasil!');
29   } else {
30     res.send('Username atau password salah!');
31   }
32 });
33

```

Gambar 4. 5 JavaScript *server.js*

- a. Gambar 4.5 merupakan kode program *JavaScript* yang berfungsi sebagai perintah untuk menjalankan *server*.

2. Membuat tampilan *web server*

Buat file *Html* baru *login.html* dan tambahkan kode berikut:

```

login.html X
C: > Windows > System32 > my-secure-server > public > login.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Login</title>
7 </head>
8 <body>
9   <h1>Login</h1>
10  <form action="/login" method="post">
11    <label for="username">Username:</label>
12    <input type="text" id="username" name="username" required<br><br>
13    <label for="password">Password:</label>
14    <input type="password" id="password" name="password" required<br><br>
15    <button type="submit">Login</button>
16  </form>
17 </body>
18 </html>
19

```

Gambar 4. 6 *Login.html*

- a. Gambar 4.6 berfungsi untuk merubah tampilan halaman *web*.
3. Jalankan *server*
Buka *Command Prompt*, navigasikan ke direktori tempat *file server.js* berada, dan jalankan perintah berikut: `node server.js`

```

C:\Windows\System32\my-secure-server>node server.js
Server berjalan di https://localhost

```

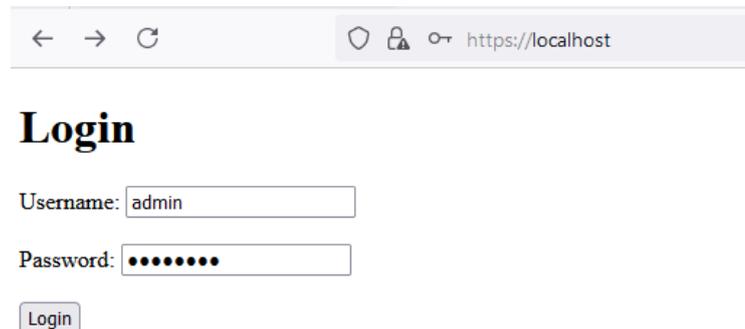
Gambar 4. 7 *Node server.js*

- a. Gambar 4.7 merupakan tampilan *Command Prompt* pada saat menjalankan *Node.js*.

4.3 ANALISIS KERENTANAN

4.3.1 Permasalahan Sertifikat CA Tidak Valid

Sertifikat *self-signed* yang digunakan pada *server* tidak diterbitkan oleh *Certificate Authority* (CA) yang terpercaya. Ketika sertifikat *self-signed* digunakan, browser akan menampilkan peringatan keamanan yang menunjukkan bahwa sertifikat tidak dapat dipercaya. Hal ini dapat dilihat pada Gambar 4.8 dimana muncul peringatan keamanan pada browser.

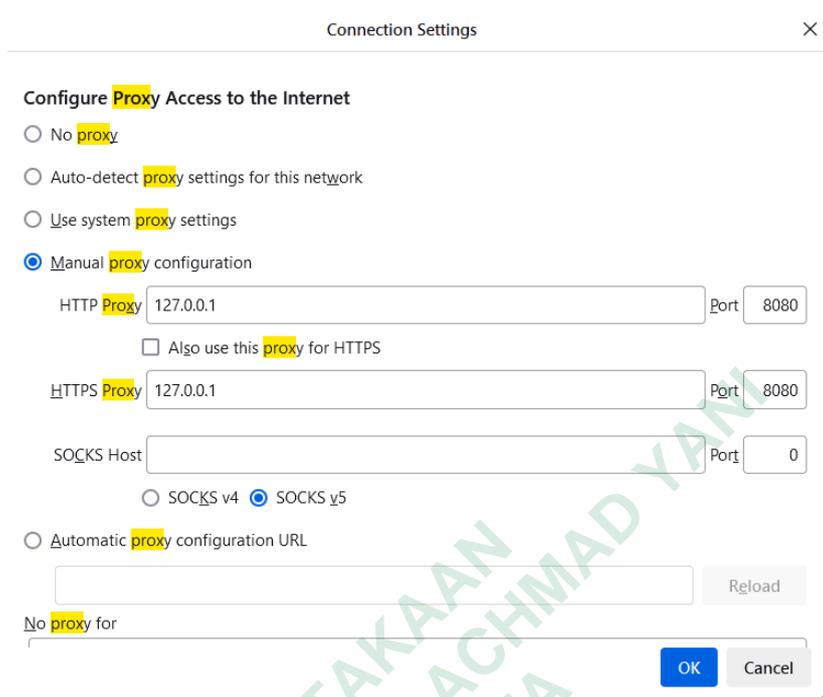


Gambar 4. 8 Tampilan *web server*

4.3.2 Instalasi Mitmproxy

Metode *sniffing* dilakukan untuk menganalisis lalu lintas jaringan dan mengidentifikasi potensi serangan. *SSL hijacking* dapat terjadi ketika berhasil mencegat dan memodifikasi lalu lintas HTTPS antara klien dan *server*. Berikut langkah-langkahnya:

1. Unduh *Python* dan instal ikuti petunjuk dengan benar.
2. Instal *mitmproxy*
 - a. Buka *Command Prompt* dan jalankan perintah berikut untuk menginstal *mitmproxy*: `pip install mitmproxy`
 - b. Setelah instalasi selesai, verifikasi bahwa *mitmproxy* telah terinstal dengan menjalankan perintah berikut: `mitmproxy --version`
 - c. Jalankan *mitmproxy* dengan perintah berikut di *Command Prompt*: `mitmproxy`
 - d. Atur browser *klien* untuk menggunakan *mitmproxy* sebagai proxy. *Setting proxy* HTTP dan HTTPS ke 127.0.0.1 dengan port 8080 (*port default mitmproxy*). *Setting* dapat dilihat pada Gambar 4.9.



Gambar 4. 9 *Setting proxy browser user*

3. Saat menggunakan *mitmproxy* untuk mencegah lalu lintas HTTPS, *device user* perlu menginstal sertifikat CA *mitmproxy* di browser. Setelah menjalankan *mitmproxy*, buka *http://mitm.it* di browser *user* dan ikuti petunjuk untuk mengunduh dan menginstal sertifikat CA.

4.3.3 Simulasi Serangan SSL Hijacking Menggunakan Metode Sniffing

Tujuan dari simulasi ini adalah untuk menunjukkan bagaimana serangan dapat dilakukan dan untuk mengidentifikasi potensi kerentanan dalam penggunaan OpenSSL pada *server* Node.js dengan sertifikat *self-signed*. Berikut langkah-langkahnya:

1. Jalankan *server* Node.js.
2. Jalankan *mitmproxy* untuk menangkap lalu lintas.
3. Buka *browser user* yang sudah di *setting* dan buka halaman *https://localhost*.
4. Analisis paket data.

Time	Method	Host	Path	Status	Content-Type	Size
17:29:45	HTTP GET	_ectportal.firefox.com	/canonical.html	200	text/html	980 146ms
17:29:46	HTTP GET	_ectportal.firefox.com	/success.txt?ipw4	200	text/plain	8b 76ms
17:29:46	HTTP GET	_ectportal.firefox.com	/success.txt?ipw4	200	text/plain	8b 59ms
17:30:05	HTTP GET	127.0.0.1	/	err	Forwarded to...	
17:30:05	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/f22fbc65-d685-47b6-9fa7-185a2b8d883d	200	text/plain	20b 215ms
17:30:06	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/56be10b9-d6c8-4ec7-a314-d78158f7b14	200	text/plain	20b 272ms
17:30:06	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/use-counters/1/6d907b9e-7602-4d2d-af18-6fa1ed81fbd	200	text/plain	20b 254ms
17:30:06	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/c666f06-f3d4-4124-4497-d116cc0f04	200	text/plain	20b 270ms
17:30:07	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/messaging-system/1/4aadf48c-1beb-4638-90fc-d753127a3857	200	text/plain	20b 252ms
17:30:07	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/events/1/ff6f8ca8-4ef7-4659-bd36-48241f385435	200	text/plain	20b 276ms
17:30:07	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/pageidnew/1/669b0b63-3063-4771-930f-4317044dae5	200	text/plain	20b 253ms
17:30:07	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/afFed964-13b8-4a84-8589-96548d1ffe01	200	text/plain	20b 210ms
17:30:07	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/9130cfd-d838-4d83-8644-9453893849f	200	text/plain	20b 253ms
17:30:08	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/baseline/1/c9ecbf58-e608-4192-b11c-be1fe37c1607	200	text/plain	20b 265ms
17:30:08	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/9580b70a-e1fa-47f1-8d45-9cb0493b11c	200	text/plain	20b 263ms
17:30:08	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/443c6459-cca1-43ec-858f-4adfa26ef225	200	text/plain	20b 237ms
17:30:09	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/58f8d6f8-efca-4462-0f54-aa967c729b6	200	text/plain	20b 256ms
17:30:09	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/newtab/1/d6f311ab-91a4-4d07-bf95-3ec74ab276b	200	text/plain	20b 238ms
17:30:09	HTTPS POST	_telemetry.mozilla.org	/submit/firefox-desktop/use-counters/1/553c972d-bfcd-45b6-a85f-e1797ce5a5e	200	text/plain	20b 217ms
17:30:11	HTTPS GET	www.facebook.com	/	200	text/html	17.5k 308ms
17:30:12	HTTPS GET	www.facebook.com	/security/htcs-plxl.gif	200	image/gif	55b 241ms
17:30:13	HTTPS POST	www.facebook.com	/ajax/bz?_a=18_ccg-EXCELLENt8_dyn-7xeE5aQ1PyUbfP41tupUmglU29zE6u7E3ruSux60V01uPE4H	200	[no content]	212ms
17:30:21	HTTPS POST	www.facebook.com	/ajax/bz?_a=18_ccg-EXCELLENt8_dyn-7xeE5aQ1PyUbfP41tupUmglU29zE6u7E3ruSux60V01uPE4H	200	[no content]	215ms
17:30:32	HTTPS POST	www.facebook.com	/ajax/bz?_a=18_ccg-EXCELLENt8_dyn-7xeE5aQ1PyUbfP41tupUmglU29zE6u7E3ruSux60V01uPE4H	200	[no content]	245ms
17:30:32	HTTPS POST	www.facebook.com	/ajax/bz?_a=18_ccg-EXCELLENt8_dyn-7xeE5aQ1PyUbfP41tupUmglU29zE6u7E3ruSux60V01uPE4H	200	[no content]	266ms
17:30:38	HTTPS GET	www.google.com	/complete/search?client=firefox&channel=fen8q-por	200	text/javascript	378b 65ms
17:30:38	HTTPS GET	pted.tb0.gstatic.com	/images?q=tb0:Am69c5_D_pF3DKyAVn4Bw-_N_l_KFbzQZvG5S6eA7c0am13rV8F3BP1zIvWk5-18	200	image/png	3.5k 47ms
17:30:41	HTTPS GET	www.google.com	/complete/search?client=firefox&channel=fen8q-por	200	text/javascript	460b 70ms
17:30:41	HTTPS GET	www.google.com	/complete/search?client=firefox&channel=fen8q-por	200	text/javascript	135b 106ms
17:30:41	HTTPS GET	www.google.com	/search?client=firefox-b-d&q=pordikunjaya	200	text/html	97.9k 569ms
17:30:42	HTTPS GET	www.google.com	/pagead/1p-conversion/16521538460/?gad_source=1&adview_type=1&adview_query_id=CjYGl-gL	204	[no content]	57ms
17:30:42	HTTPS POST	www.google.com	/gen_204?atyp=c18e1-8p8-Zu_KL75zsmP3N14gAc8ct-s1h8v-t18m-mV8aQ1d-8p8-zpVMD_JemMPz66s	204	[no content]	36ms
17:30:42	HTTPS GET	...googleadservices.com	/pagead/conversion/16521538460/?gad_source=1&adview_type=1&adview_query_id=CjYGl-gL	204	[no content]	53ms
17:30:42	HTTPS GET	www.google.com	/complete/search?q&cp=8&client=gws-wi-serp&ssi=1&gs_pcr=2&hl=id&authuser=0&pp=pordik	200	application/json	8.7k 331ms
17:30:42	HTTPS GET	www.google.com	/complete/search?pp=1&2unja&pp=8&lient=desktop&hl=id-on-focus-serp&ssi=1&gs	200	application/json	423b 222ms
17:30:42	HTTPS POST	www.google.com	/gen_204?atyp=c18e1-8p8-Zu_KL75zsmP3N14gAc8ct-s1h8v-t18m-mV8aQ1d-8p8-zpVMD_JemMPz66s	204	[no content]	61ms
17:30:42	HTTPS POST	www.google.com	/xjs/_/js/k-xjs.s.id.1yDnbalFVjc.O/ck-xjs.s.cdHEspE1GjO.L.F4.O/am-AHyBQaAgACDBGQAAAA	200	text/javascript	156k 424ms
17:30:42	HTTPS GET	www.google.com	/xjs/_/js/k-xjs.s.id.1yDnbalFVjc.O/am-AHyBQaAgACDBGQAAAA	200	text/javascript	160k 213ms
17:30:42	HTTPS GET	www.google.com	/client_204?atyp=1&hl=1&8d1h-8314dp-1&el-8p8-Zu_KL75zsmP3N14gAc8ct-s1h8v-t18m-mV8aQ1d-8p8-zpVMD_JemMPz66s	204	[no content]	95ms
17:30:42	HTTPS POST	www.google.com	/gen_204?atyp=1&el-8p8-Zu_KL75zsmP3N14gAc8ct-s1h8v-t18m-mV8aQ1d-8p8-zpVMD_JemMPz66s	204	[no content]	133ms
17:30:43	HTTPS GET	accounts.google.com	/RotateCookiesPage?og_pld=1&ort=3&origIn=https%3A%2F%2Fwww.google.com&exp_id=0	200	text/html	280b 64ms
17:30:43	HTTPS GET	www.google.com	/xjs/_/js/k-xjs.s.cdHEspE1GjO.L.F4.O/am-AHyBQaAgACDBGQAAAA	200	text/css	1.3k 75ms
17:30:43	HTTPS GET	www.google.com	/xjs/_/js/k-xjs.s.id.1yDnbalFVjc.O/am-AHyBQaAgACDBGQAAAA	200	text/javascript	139k 153ms
17:30:43	HTTPS POST	www.google.com	/gen_204?atyp=1&el-8p8-Zu_KL75zsmP3N14gAc8ct-s1h8v-t18m-mV8aQ1d-8p8-zpVMD_JemMPz66s	204	[no content]	97ms

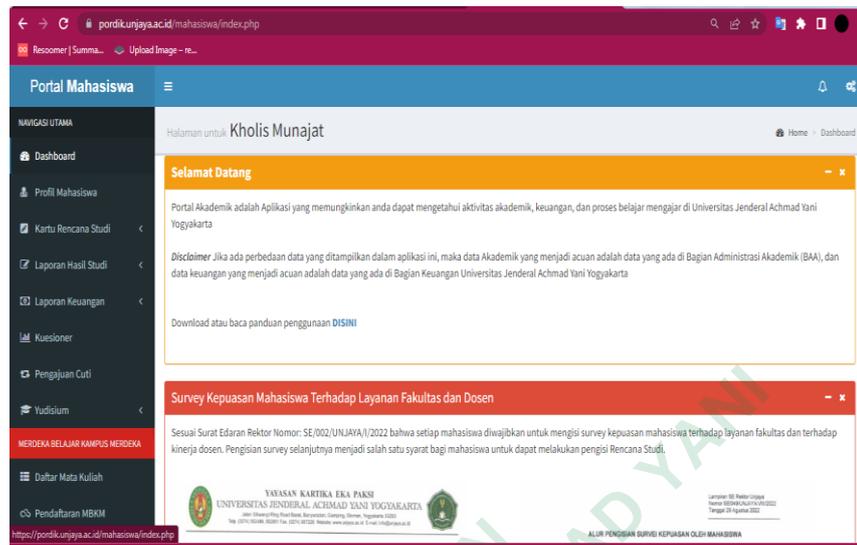
Gambar 4. 10 Tampilan capture data mitmproxy

- a. Pada Gambar 4.10 data <https://localhost> tidak terdeteksi akibat dari Sertifikat *self-signed* yang digunakan pada server tidak diterbitkan oleh Certificate Authority (CA) yang terpercaya.

4.3.4 Uji Coba Serangan Pada Halaman Web Lain

Setelah melakukan simulasi serangan SSL hijacking pada server lokal, langkah selanjutnya adalah melakukan uji coba serangan pada halaman web lain untuk mengidentifikasi potensi kerentanan lebih lanjut. Disini user menggunakan halaman web <https://pordik.unjaya.ac.id> sebagai uji coba. Berikut langkah-langkahnya:

1. Buka halaman web <https://pordik.unjaya.ac.id> dan login. Halaman login dapat dilihat pada Gambar 4.11.



Gambar 4. 11 Halaman <https://pordik.unjaya.ac.id>

2. Cek Sertifikat SSL pada halaman <https://pordik.unjaya.ac.id> . Pada halaman tersebut menampilkan dua sertifikat SSL dikarenakan browser sudah di *setting* dengan *mitmproxy*. Hal tersebut dapat dilihat pada Gambar 4.12.

*.unjaya.ac.id		mitmproxy
Subject Name		
Common Name	mitmproxy	
Organization	mitmproxy	
Issuer Name		
Common Name	mitmproxy	
Organization	mitmproxy	
Validity		
Not Before	Tue, 25 Jun 2024 19:19:42 GMT	
Not After	Sun, 25 Jun 2034 19:19:42 GMT	
Public Key Info		
Algorithm	RSA	

Gambar 4. 12 Sertifikat SSL *double*

4.4 EVALUASI HASIL UJI COBA

4.4.1 Hasil Pengamatan

Dengan menggunakan *mitmproxy*, lalu lintas HTTPS dari halaman *web* eksternal dapat dicegat dan dimodifikasi. Hal ini menunjukkan bahwa serangan MITM dapat dilakukan pada situs *web* yang menggunakan sertifikat valid jika penyerang berhasil menempatkan dirinya sebagai *proxy*.

4.4.2 Rekomendasi Keamanan

Serangan ini menyoroti pentingnya kepercayaan pada sertifikat SSL/TLS yang digunakan oleh situs *web*. Meskipun situs *web* menggunakan sertifikat yang valid, lalu lintas bisa dicegat oleh *proxy* dan data sensitif dapat diakses. Berikut rekomendasi dari penulis yang mungkin dapat meminimalisir ancaman serangan :

1. Gunakan mekanisme keamanan tambahan seperti *HTTP Strict Transport Security* (HSTS) untuk memastikan bahwa *browser* hanya melakukan koneksi ke *server* yang menggunakan HTTPS yang valid.
2. Gunakan alat keamanan seperti *Content Security Policy* (CSP) untuk mengurangi risiko modifikasi SSL oleh pihak ketiga.

4.4.3 Langkah Mitigasi

Untuk mengurangi risiko serangan SSL *hijacking* yang telah diidentifikasi dalam penelitian ini dikarenakan penulis berperan sebagai pengguna (*user*) dan penyerang (*attacker*), Salah satu poin utama adalah mitigasi serangan tidak memerlukan penggunaan VPN selama *browser* pengguna tidak dikonfigurasi untuk menggunakan *proxy mitmproxy* dan tidak menginstal sertifikat *mitmproxy*.