

Nama : Ferza Reyaldi
NIM : 09021281924060
Mata Kuliah : Keamanan Jaringan Komputer

Tugas Pertemuan 4 (Pertemuan 14)

Mencoba enkripsi / dekripsi dengan bantuan aplikasi yang disediakan oleh situs ini
<https://www.ti89.com/cryptotut/substitution.htm>

Jawab:

Enkripsi / dekripsi menggunakan Caesar Chiper

Cryptography > Substitution Ciphers (45 min.)

Objectives:

- 1) Understand what Substitution Ciphers are and how they work.
- 2) Distinguish between Mono- and Polyalphabetic Ciphers
- 3) Encrypt using the Caesar, the Atbash and the Playfair Cipher.
- 3) Learn how to break Substitution Ciphers.

On this page I will introduce you to Substitution Ciphers. Instead of shuffling, we now substitute plain letters. Precisely:

In Substitution Ciphers same plain letters are replaced by same cipher letters.

On the previous page, we learned the weakness of transposition ciphers: The plain letters made up the cipher text. Thus, combining the cipher letters in a clever manner must yield the plain letter eventually. An attempt to improve the security level of the encrypted messages is to hide the plain letters by replacing them with other letters. Most of the ciphers in the tutorial are Substitution Ciphers. Therefore, I will be brief with the introduction of such Ciphers on this page.

Example 1: (Caesar Cipher) The simplest of all substitution ciphers is the one in which the cipher letters results from shifting plain letters by the same distance. Among those, the best known is called "Caesar Cipher", used by Julius Caesar, in which each A is encrypted as D, B as E, C as F,... etc.

Plain text: Cipher text:

ENCODE ==> <== DECODE

Exercise 1: Use the Caesar Cipher to encode "A man a plan a canal" by hand.

Exercise 2: Decode the ciphertext "fdhvdvudodg" by hand.

Example 2: (Atbash Cipher) Try to break the following monoalphabetic substitution cipher.

Related web sources:

- [Enigma and the Codebreakers](#)
- [Yahoo's Encryption & Security](#)
- [Britannica.com](#)
- [Dictionary.com](#)
- [Glossary](#)
- [PBS Online](#)
- [Introduction to Cryptography](#)
- [Enigma History](#)
- [Enigma Emulator](#)

Cryptography > Substitution Ciphers (45 min.)

Objectives:

- 1) Understand what Substitution Ciphers are and how they work.
- 2) Distinguish between Mono- and Polyalphabetic Ciphers
- 3) Encrypt using the Caesar, the Atbash and the Playfair Cipher.
- 3) Learn how to break Substitution Ciphers.

On this page I will introduce you to Substitution Ciphers. Instead of shuffling, we now substitute plain letters. Precisely:

In Substitution Ciphers same plain letters are replaced by same cipher letters.

On the previous page, we learned the weakness of transposition ciphers: The plain letters made up the cipher text. Thus, combining the cipher letters in a clever manner must yield the plain letter eventually. An attempt to improve the security level of the encrypted messages is to hide the plain letters by replacing them with other letters. Most of the ciphers in the tutorial are Substitution Ciphers. Therefore, I will be brief with the introduction of such Ciphers on this page.

Example 1: (Caesar Cipher) The simplest of all substitution ciphers is the one in which the cipher letters results from shifting plain letters by the same distance. Among those, the best known is called "Caesar Cipher", used by Julius Caesar, in which each A is encrypted as D, B as E, C as F,... etc.

Plain text: Cipher text:

ENCODE ==> <== DECODE

Exercise 1: Use the Caesar Cipher to encode "A man a plan a canal" by hand.

Exercise 2: Decode the ciphertext "fdhvdvudodg" by hand.

Example 2: (Atbash Cipher) Try to break the following monoalphabetic substitution cipher.

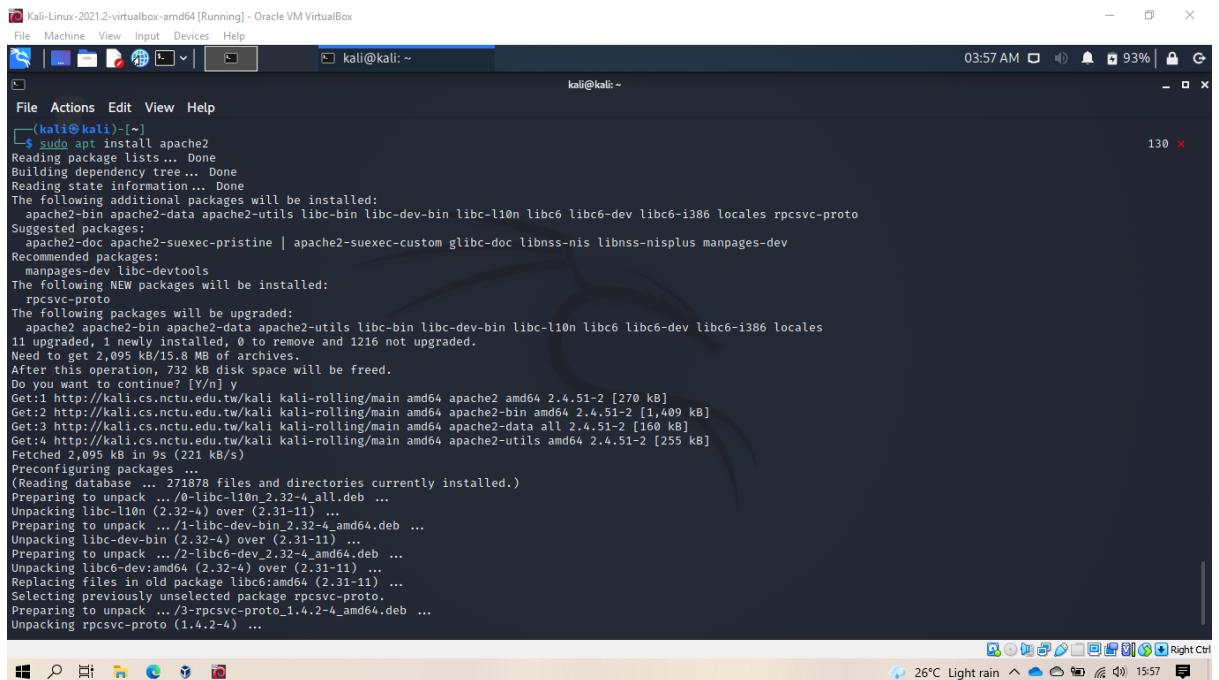
Related web sources:

- [Enigma and the Codebreakers](#)
- [Yahoo's Encryption & Security](#)
- [Britannica.com](#)
- [Dictionary.com](#)
- [Glossary](#)
- [PBS Online](#)
- [Introduction to Cryptography](#)
- [Enigma History](#)
- [Enigma Emulator](#)

Menganalisa traffic HTTP menggunakan Wireshark

Jawab:

1) Instal web server apache



```
Kali-Linux-2021.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
└─$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libc-bin libc-dev-bin libc-l10n libc6 libc6-dev libc6-i386 locales rpcsvc-proto
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom glibc-doc libnss-nis libnss-nisplus manpages-dev
Recommended packages:
  manpages-dev libc-devtools
The following NEW packages will be installed:
  rpcsvc-proto
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils libc-bin libc-dev-bin libc-l10n libc6 libc6-dev libc6-i386 locales
11 upgraded, 1 newly installed, 0 to remove and 1216 not upgraded.
Need to get 2,095 kB/15.8 MB of archives.
After this operation, 732 kB disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://kali.cs.nctu.edu.tw/kali kali-rolling/main amd64 apache2 amd64 2.4.51-2 [270 kB]
Get:2 http://kali.cs.nctu.edu.tw/kali kali-rolling/main amd64 apache2-bin amd64 2.4.51-2 [1,409 kB]
Get:3 http://kali.cs.nctu.edu.tw/kali kali-rolling/main amd64 apache2-data all 2.4.51-2 [160 kB]
Get:4 http://kali.cs.nctu.edu.tw/kali kali-rolling/main amd64 apache2-utils amd64 2.4.51-2 [255 kB]
Fetched 2,095 kB in 9s (221 kB/s)
Preconfiguring packages ...
(Reading database ... 271878 files and directories currently installed.)
Preparing to unpack .../0-libc-l10n_2.32-4_all.deb ...
Unpacking libc-l10n (2.32-4) over (2.31-11) ...
Preparing to unpack .../1-libc-dev-bin_2.32-4_amd64.deb ...
Unpacking libc-dev-bin (2.32-4) over (2.31-11) ...
Preparing to unpack .../2-libc6-dev_2.32-4_amd64.deb ...
Unpacking libc6-dev:amd64 (2.32-4) over (2.31-11) ...
Replacing files in old package libc6:amd64 (2.31-11) ...
Selecting previously unselected package rpcsvc-proto.
Preparing to unpack .../3-rpcsvc-proto_1.4.2-4_amd64.deb ...
Unpacking rpcsvc-proto (1.4.2-4) ...
```

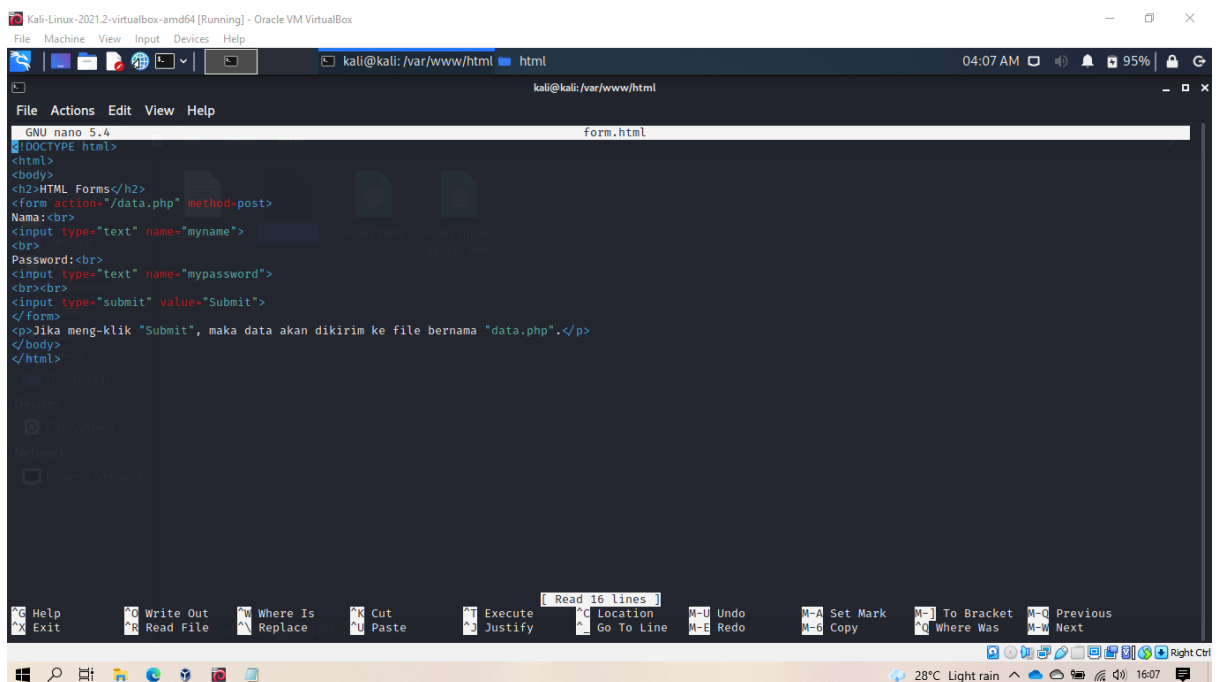
2) Buat file form.html dan data.php pada folder /var/www/html



```
(kali@kali)-[/var/www/html]
└─$ sudo touch form.html

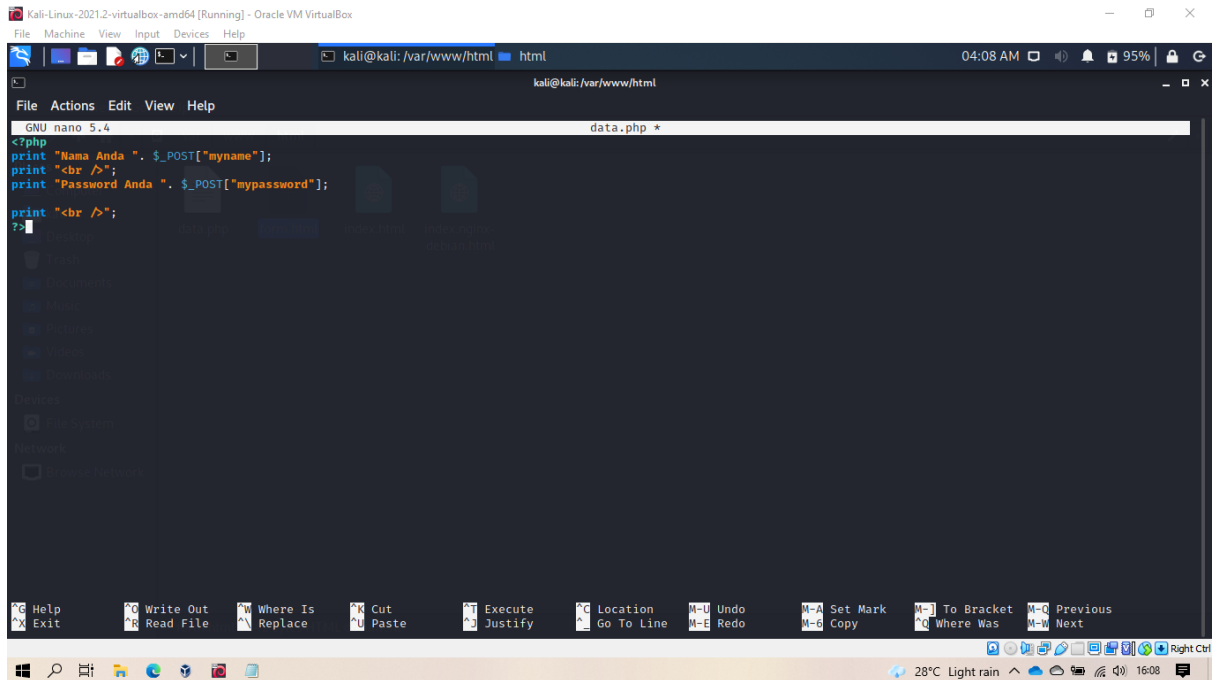
(kali@kali)-[/var/www/html]
└─$ sudo touch data.php
```

3) Tulis script pada form.html.



```
Kali-Linux-2021.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: /var/www/html
File Actions Edit View Help
GNU nano 5.4 form.html
┌DOCType html>
<html>
<body>
<h2>HTML Forms</h2>
<form action="/data.php" method="post">
  Name:<br>
  <input type="text" name="myname">
  <br>
  Password:<br>
  <input type="text" name="mypassword">
  <br><br>
  <input type="submit" value="Submit">
</form>
<p>Jika meng-klik "Submit", maka data akan dikirim ke file bernama "data.php".</p>
</body>
</html>
```

4) Tuis script pada file data.php.

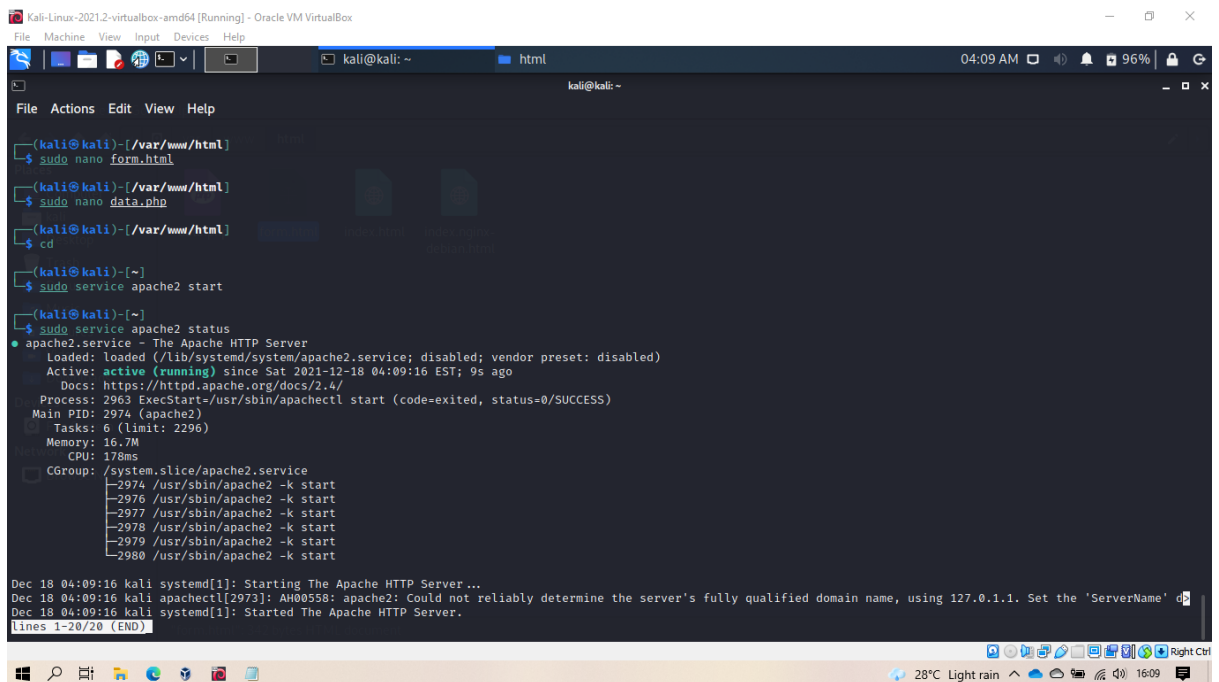


The screenshot shows a Kali Linux terminal window with the nano editor open, editing the file `data.php` located at `/var/www/html/`. The code in the file is as follows:

```
<?php
print "Nama Anda ". $_POST["myname"];
print "<br />";
print "Password Anda ". $_POST["mypassword"];
print "<br />";
?>
```

The terminal window also shows the system's taskbar at the bottom with various application icons and system status indicators like temperature and weather.

5) Aktifkan web server apache.



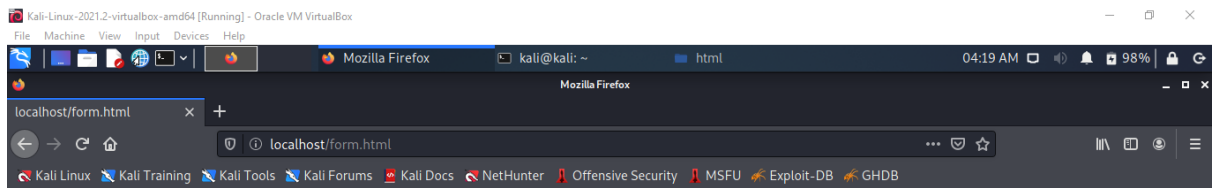
The screenshot shows a Kali Linux terminal window with the following commands and output:

```
(kali@kali)-[/var/www/html]
$ sudo nano form.html
(kali@kali)-[/var/www/html]
$ sudo nano data.php
(kali@kali)-[/var/www/html]
$ cd
(kali@kali)-[~]
$ sudo service apache2 start
(kali@kali)-[~]
$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-12-18 04:09:16 EST; 9s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2963 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 2974 (apache2)
       Tasks: 6 (limit: 2296)
      Memory: 16.7M
         CPU: 178ms
    CGroup: /system.slice/apache2.service
            └─2974 /usr/sbin/apache2 -k start
              └─2976 /usr/sbin/apache2 -k start
                └─2977 /usr/sbin/apache2 -k start
                  └─2978 /usr/sbin/apache2 -k start
                    └─2979 /usr/sbin/apache2 -k start
                      └─2980 /usr/sbin/apache2 -k start

Dec 18 04:09:16 kali systemd[1]: Starting The Apache HTTP Server...
Dec 18 04:09:16 kali apachectl[2973]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName'
Dec 18 04:09:16 kali systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

The terminal window also shows the system's taskbar at the bottom with various application icons and system status indicators like temperature and weather.

- 6) IP web server saya adalah 192.168.56.101.
- 7) Buka 192.168.56.101/form.html atau localhost/form.html.
- 8) Pada kotak Nama dan Password ketikkan sesuatu, lalu klik Submit



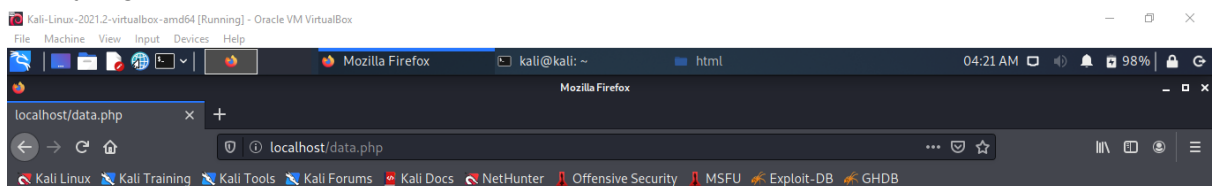
HTML Forms

Nama:

Password:

Jika meng-klik "Submit", maka data akan dikirim ke file bernama "data.php".

- 9) Data yang diketikkan pada localhost/form.html ditampilkan pada localhost/data.php.



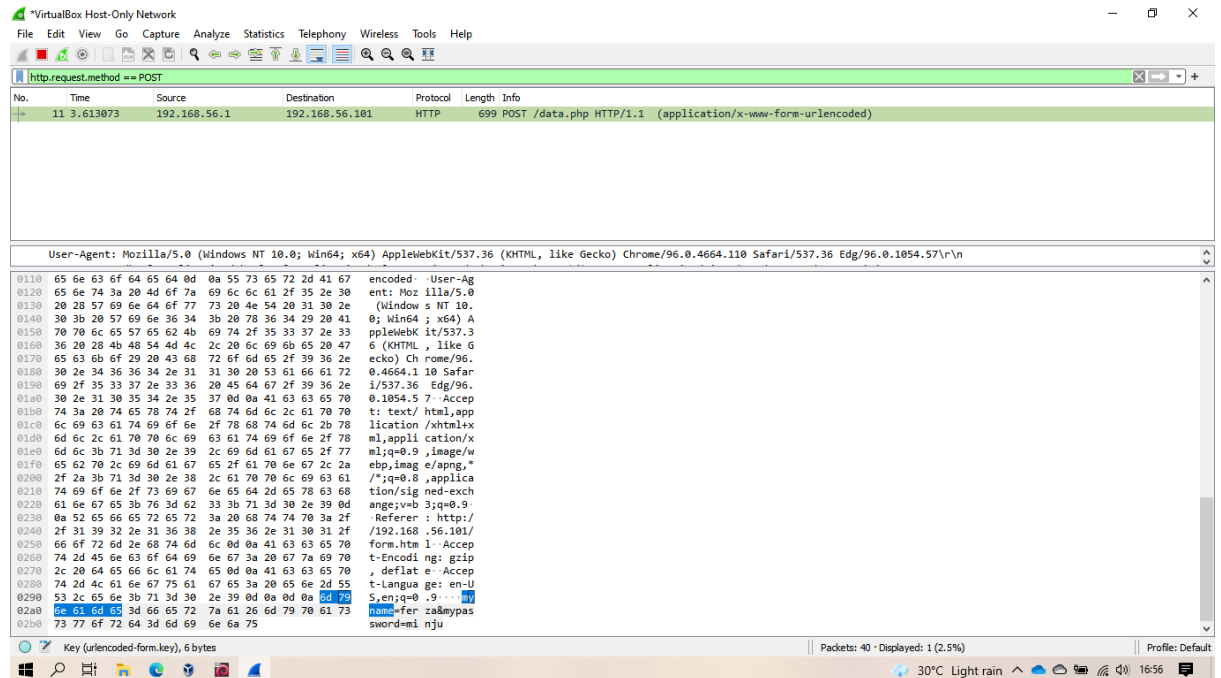
Nama Anda ferza
Password Anda minju

- 10) Jalankan Wireshark pada komputer sniffer untuk menganalisa traffic HTTP antara Web client dan Web server. Mulailah melakukan capture traffic, klik Capture > Start.

Analisa paket yang baru saja dicapture, apakah Anda berhasil menemukan kata “ferza” atau “minju” ?

Jawab:

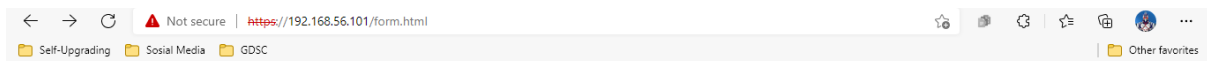
- 1) Pada kotak filter, ketikkan **http.request.method == POST**.
- 2) Kemudian klik enter.



Berdasarkan hasil analisis paket yang telah dicapture, dapat ditemukan kedua kata tersebut dikarenakan tidak menggunakan SSL certificate, sehingga paket yang dikirim tidak dienkripsi oleh server.

Analisa traffic https!

Jawab:

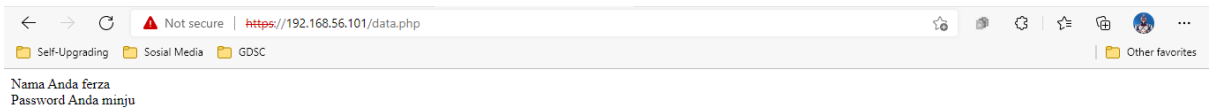


HTML Forms

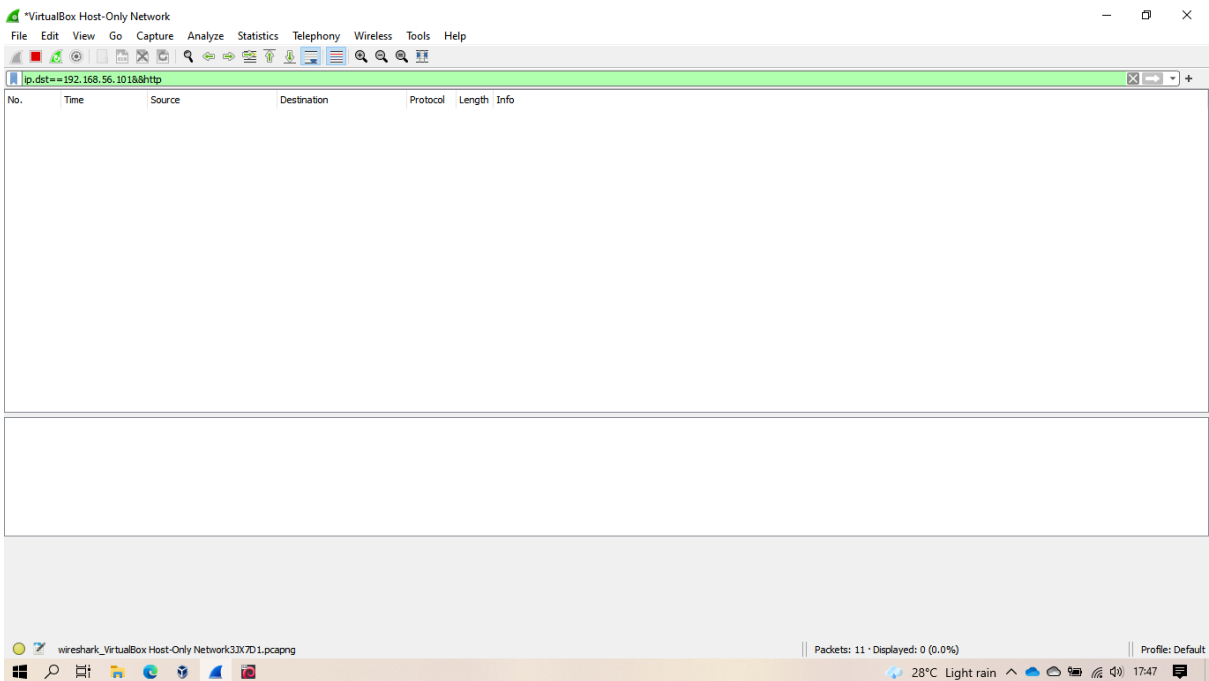
Nama:
ferza
Password:
minju

Submit

Jika meng-klik "Submit", maka data akan dikirim ke file bernama "data.php".



Ketika diterapkan filter **ip.dst==192.168.56.101&&http**, tidak ditemukan paket apapun yang berarti tidak ada paket dengan destinasi 192.168.56.101 dengan protokol http dikarenakan kita telah menerapkan SSL certificate (HTTPS).



Kemudian saya coba menggunakan filter **tcp.port == 443**

The image shows a Wireshark network traffic capture from a VirtualBox Host-Only Network. The filter applied is **tcp.port == 443**. The packet list shows a series of TLS/SSL handshake packets between 192.168.56.1 and 192.168.56.1. Packet 31 is highlighted, showing a 'Change Cipher Spec' message. The packet details pane shows the 'Internet Protocol Version 4' section, indicating the source and destination addresses. The packet bytes pane shows the raw data of the packet, including the TLS record structure.

No.	Time	Source	Destination	Protocol	Length	Info
22	0.005652	192.168.56.101	192.168.56.1	TLSv1.3	294	Server Hello, Change Cipher Spec, Application Data, Application Data
23	0.006130	192.168.56.1	192.168.56.101	TLSv1.3	84	Change Cipher Spec, Application Data
24	0.006397	192.168.56.1	192.168.56.101	TCP	54	51770 → 443 [FIN, ACK] Seq=595 Ack=241 Win=2102016 Len=0
25	0.006437	192.168.56.101	192.168.56.1	TCP	60	443 → 51770 [ACK] Seq=241 Ack=595 Win=64128 Len=0
26	0.006773	192.168.56.101	192.168.56.1	TCP	60	443 → 51770 [FIN, ACK] Seq=241 Ack=595 Win=64128 Len=0
27	0.006870	192.168.56.1	192.168.56.101	TCP	54	51770 → 443 [ACK] Seq=596 Ack=242 Win=2102016 Len=0
28	0.007247	192.168.56.101	192.168.56.1	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data, Application Data
29	0.007263	192.168.56.101	192.168.56.1	TLSv1.3	200	Application Data, Application Data
30	0.007367	192.168.56.1	192.168.56.101	TCP	54	51771 → 443 [ACK] Seq=518 Ack=1607 Win=2102272 Len=0
31	0.007736	192.168.56.1	192.168.56.101	TLSv1.3	84	Change Cipher Spec, Application Data
32	0.007868	192.168.56.1	192.168.56.101	TCP	54	51771 → 443 [FIN, ACK] Seq=548 Ack=1607 Win=2102272 Len=0
33	0.007918	192.168.56.101	192.168.56.1	TCP	60	443 → 51771 [ACK] Seq=1607 Ack=548 Win=64128 Len=0
34	0.008331	192.168.56.101	192.168.56.1	TCP	60	443 → 51771 [FIN, ACK] Seq=1607 Ack=549 Win=64128 Len=0
35	0.008439	192.168.56.1	192.168.56.101	TCP	54	51771 → 443 [ACK] Seq=549 Ack=1608 Win=2102272 Len=0
36	0.008738	192.168.56.1	192.168.56.101	TCP	66	51772 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
37	0.008892	192.168.56.101	192.168.56.1	TCP	66	443 → 51772 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
38	0.008979	192.168.56.1	192.168.56.101	TCP	54	51772 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
39	0.009207	192.168.56.1	192.168.56.101	TLSv1.3	571	Client Hello
40	0.009402	192.168.56.101	192.168.56.1	TCP	60	443 → 51772 [ACK] Seq=1 Ack=518 Win=64128 Len=0
41	0.011522	192.168.56.101	192.168.56.1	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data, Application Data

Terlihat pada paket no 31, bahwa telah dilakukan enkripsi yang ditandai dengan info **'change chiper spec...'** sehingga password yang telah kita buat tidak bisa kita lihat langsung melalui paket.