Progress Report 3

Kelompok 2
Dave Travis - 2201020008
Muhammad Noval - 2201020014
Rizsky Parsadanta R. - 2201020117
Arya Winata - 2201020001

Judul proyek	:	Analisis Serangan Sniffing & Mitigasinya di Wireshark
Tujuan proyek	:	Memahami cara kerja penyerangan *packet sniffing* dan metode pencegahannya.
Target proyek (Minggu 3)	:	Capture dan analisis HTTP paket login

Pengantar:

Pada progress minggu kedua ini, kami akan menangkap dan menganalisa paket HTTP saat pengguna login ke halaman web, lalu menunjukkan bahwa *password* (dan data sensitif lainnya) dikirim dalam bentuk *plain text*, sehingga bisa dengan mudah dilihat oleh penyerang.

Dikarenakan pada progress minggu ke dua kami memutuskan untuk menggunakan VirtualBox dalam pembuatan rancangan jaringan nya, maka progress minggu ini akan dilanjutkan menggunakan topologi VirtualBox yang sudah ada.

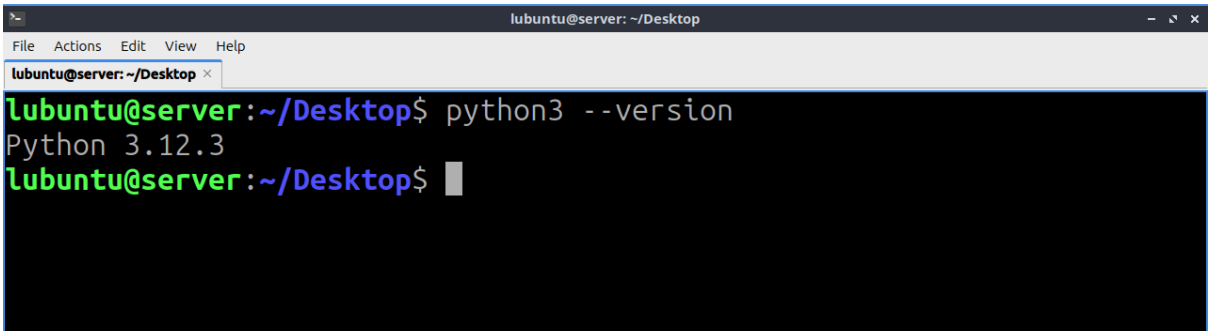
Pelaksanaan via VirtualBox:

VM yang sudah disiapkan berdasarkan progress minggu lalu ialah 3 vm, yaitu klien, router, dan server. Karena saat ini targetnya ialah analisis *traffic* HTTP, sehingga diperlukannya hosting website login sederhana, penguji website login, dan yang melakukan analisis traffic.

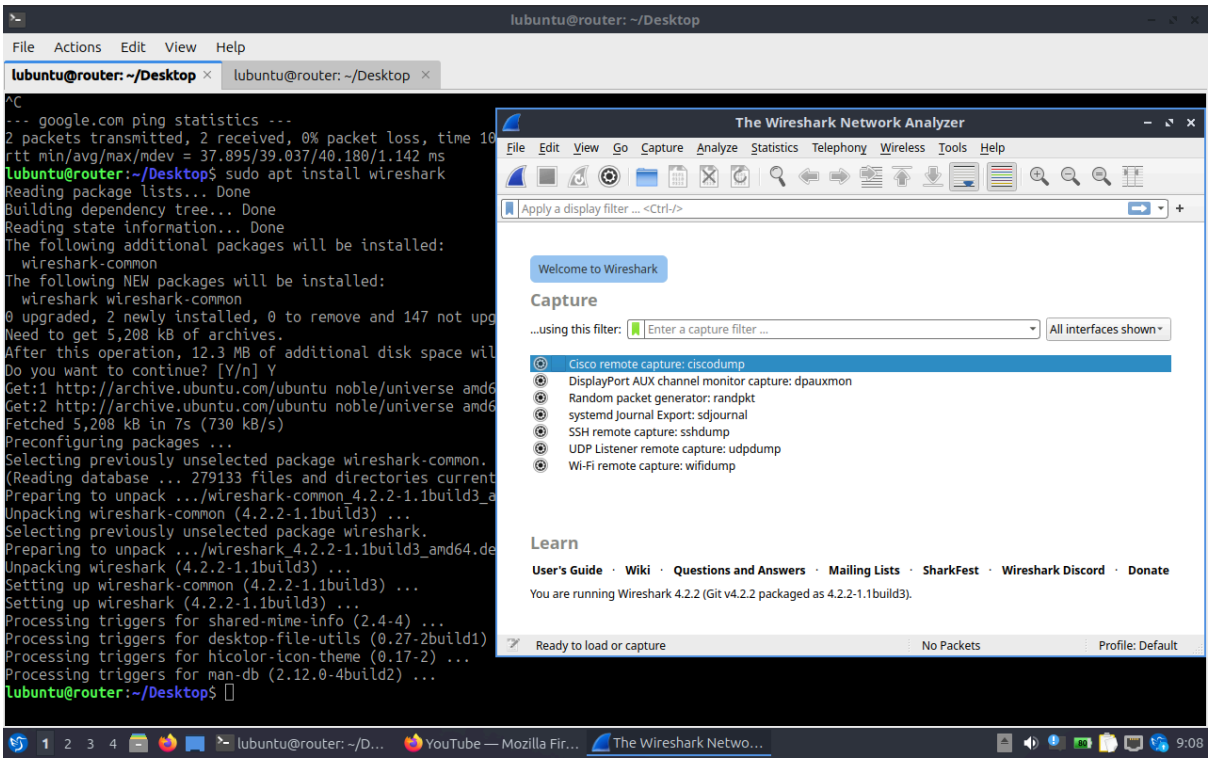
Kami akan membagi ketiga tugas tersebut terhadap masing-masing vm, di mana hosting website login sederhana akan dilakukan oleh vm server, penguji website login oleh vm klien, dan analisis *traffic* via Wireshark oleh vm router (dikarenakan berada di tengah-tengah topologi, penghubung antara vm klien dan vm server).

Persiapan:

Sebelum dilakukannya pengujian, ada baiknya kami mempersiapkan terlebih dahulu tools-tools yang diperlukan pada tahapan ini pada seluruh vm.



Pada vm server, kami memastikan bahwa Python sebagai media hosting website login HTTP sederhana sudah terinstall, walaupun Python sudah default pada OS Linux. Namun, tidak ada salahnya untuk *double check*, bukan?



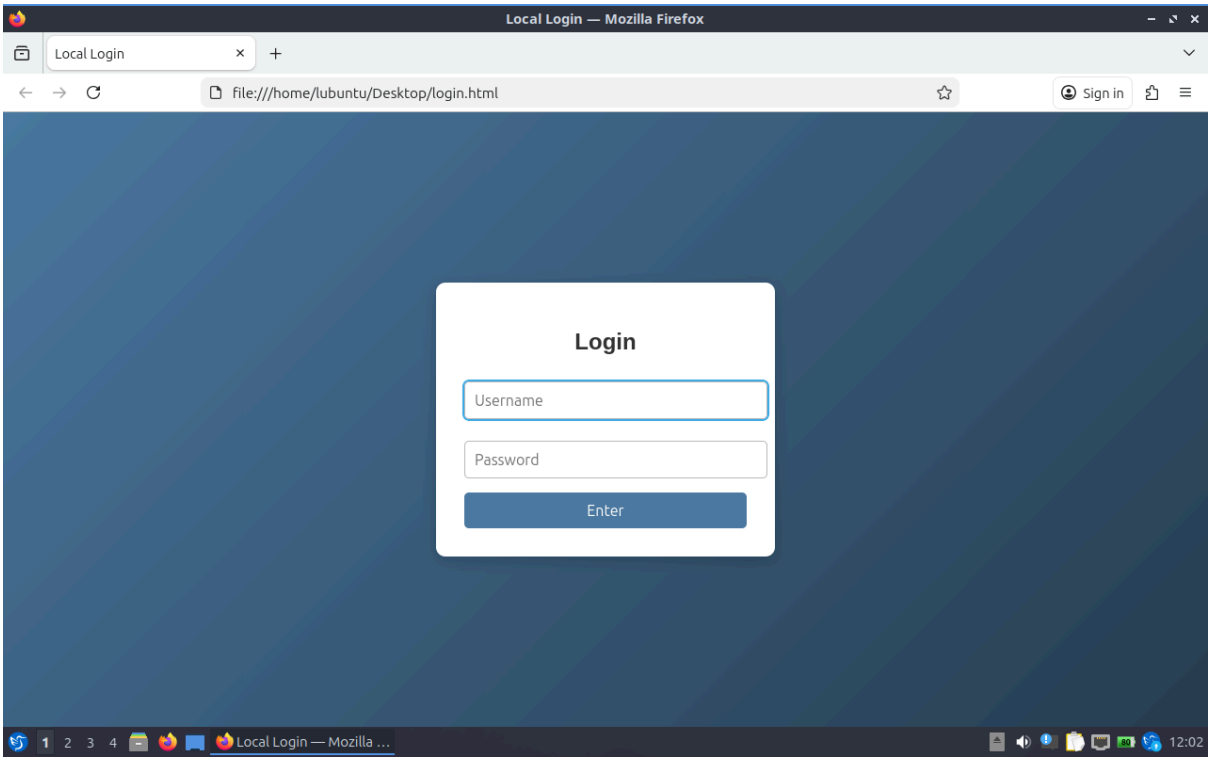
Lalu, pada vm router, kami menginstall Wireshark yang akan digunakan untuk memantau traffic pada percobaan kali ini.

Dan pada vm klien, hanya diperlukan *browser* saja untuk percobaan kali ini, dan kami akan menggunakan Firefox sebagai *browser* bawaan dari Lubuntu.

Percobaan:

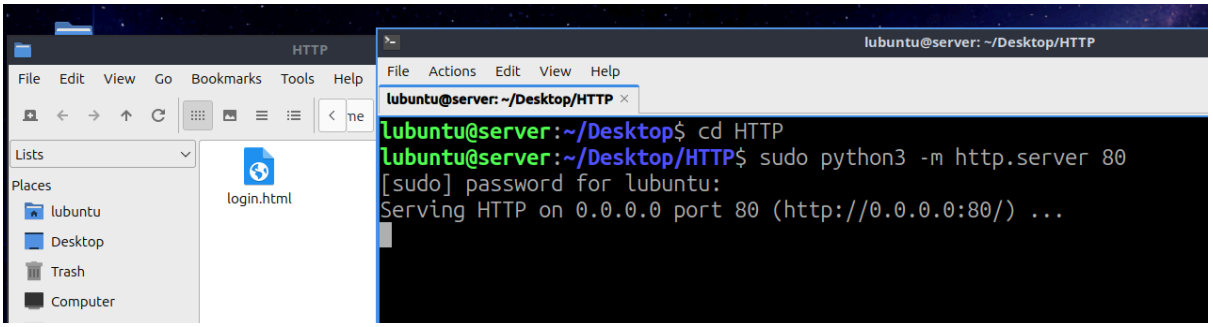
Karena minggu ini tema percobaannya ialah login HTTP, maka pada vm server diperlukannya *website* sederhana yang menyediakan website lokal yang mendukung sistem POST login, sehingga bisa dianalisis *traffic* nya, maka kami membuat nya seperti ini:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Login - HTTP Test</title>
</head>
<body>
  <h2>Form Login (HTTP)</h2>
  <form method="POST" action="/">
    <label>Username: <input type="text" name="username" required></label><br><br>
    <label>Password: <input type="password" name="password" required></label><br><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
```

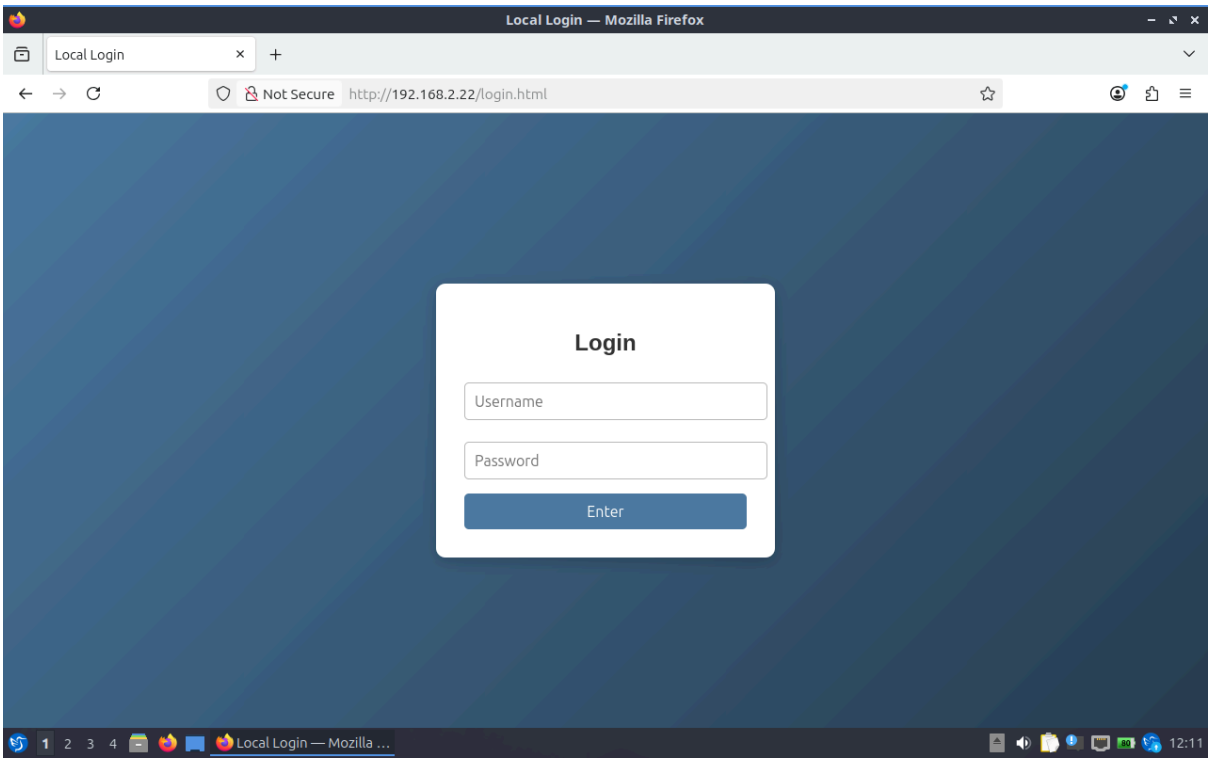


Nah, karena halaman login.html sudah siap, maka saat nya kami menghosting secara lokal dari vm server, sehingga nantinya kalau sudah up, bisa diakses oleh vm klien, karena sudah terhubung via *interface* lokal.

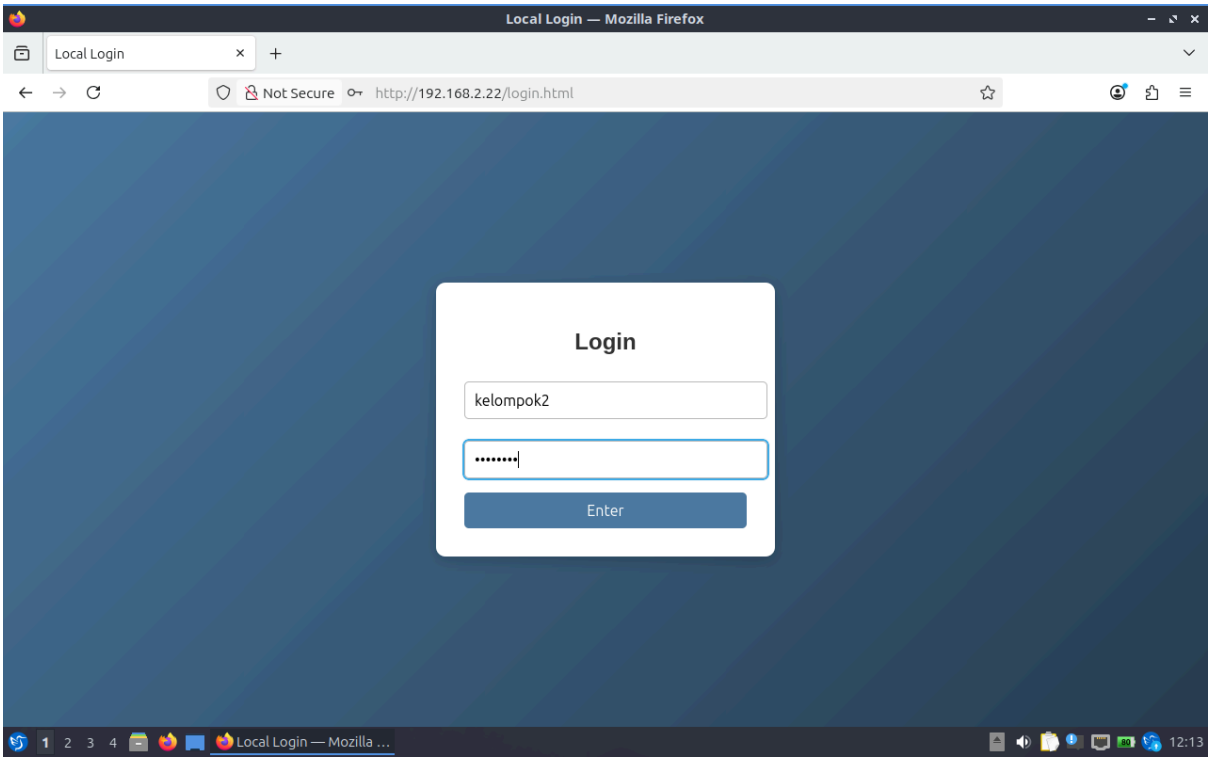
Dikarenakan pada vm server tadi sudah terinstall python, maka tinggal dijalankan web server nya berdasarkan direktori file login.html nya, dan filenya pun sudah siap untuk diakses dari vm klien.



Setelah server siap menjalankan web server dari file login.html tadi, maka kita berpindah ke vm klien dan membuka laman IP address dari vm server, yaitu 192.168.2.22.

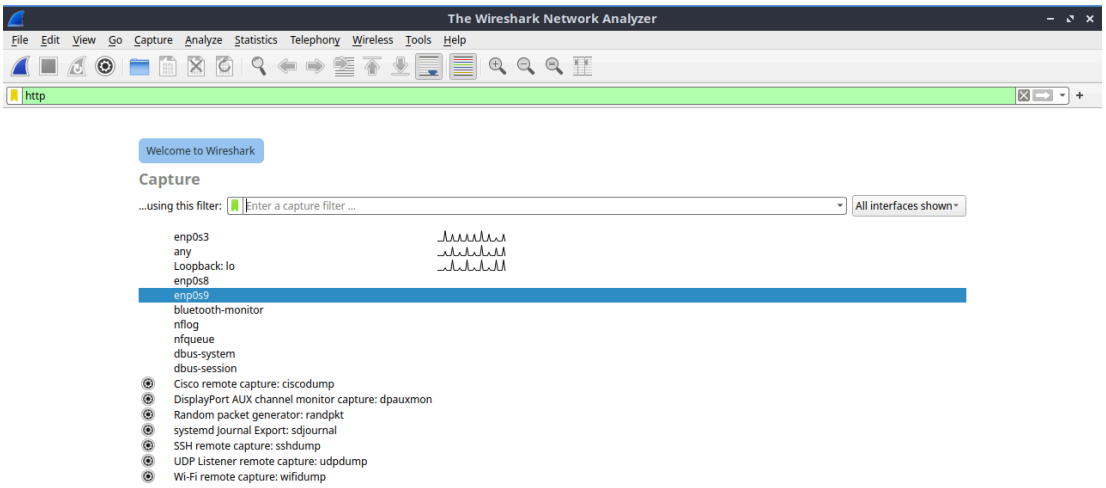


Nah, sudah terlihat halaman loginnya dari vm klien, maka tinggal kita coba login saja.



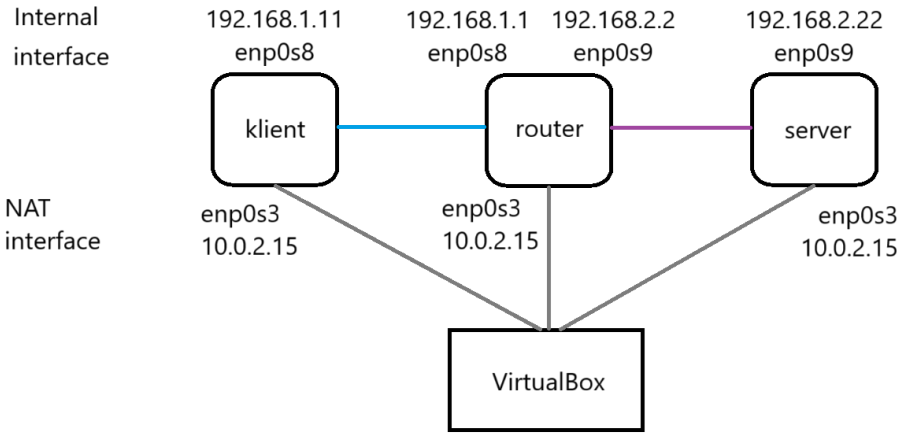
Kami mencoba untuk memasukkan username: kelompok2, password: testuser. Dan setelah siap diisi, maka kami menekan tombol enter, sehingga data dari vm klien dapat masuk ke dalam jaringan vm server, dengan dibuktikan dari pengecekan via Wireshark di vm router, yang mana akan kami jelaskan di bawah.

Nah, karena tugas dari vm router itu sebagai penghubung antara vm klien dan vm server, sudah patut kalau pengecekan dilakukan di vm ini, karena menjadi pusat traffic. Maka dari itu, kami akan mengecek nya dari vm ini.

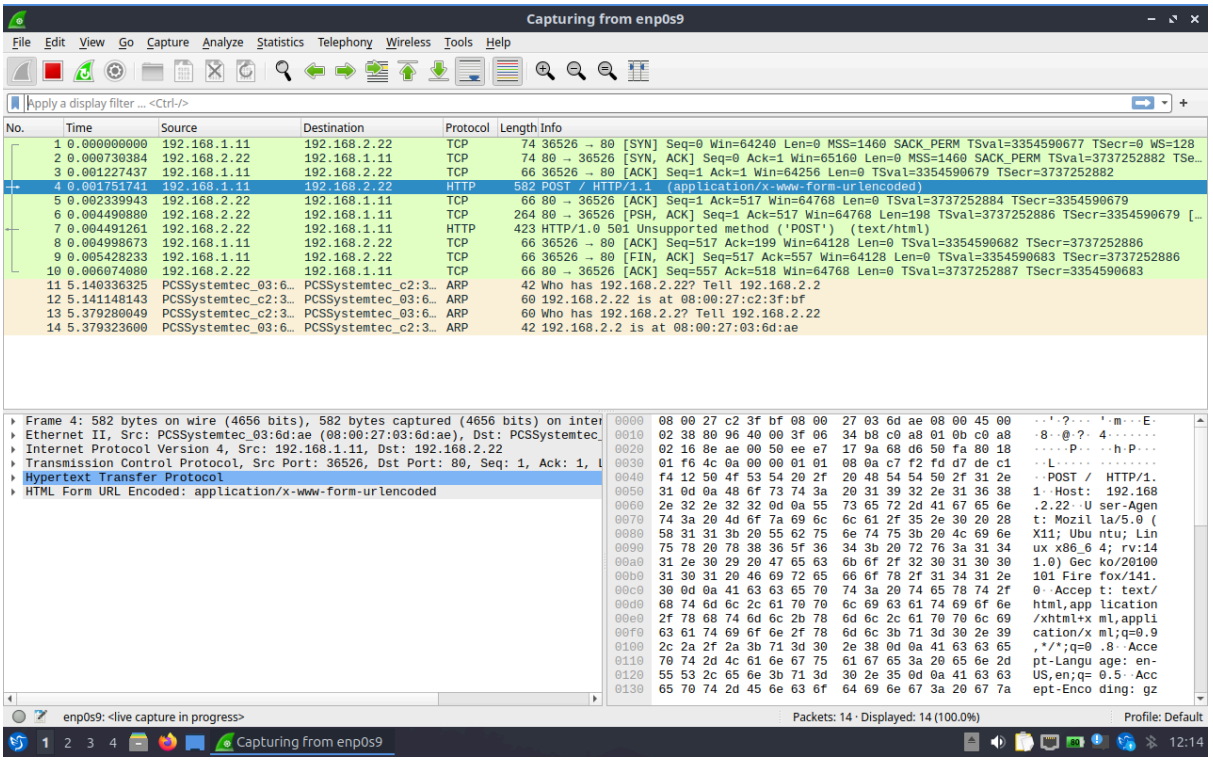


Dan bisa dilihat ada beberapa port yang dapat kita *capture traffic* nya, namun berdasarkan dari topologi jaringan kami:

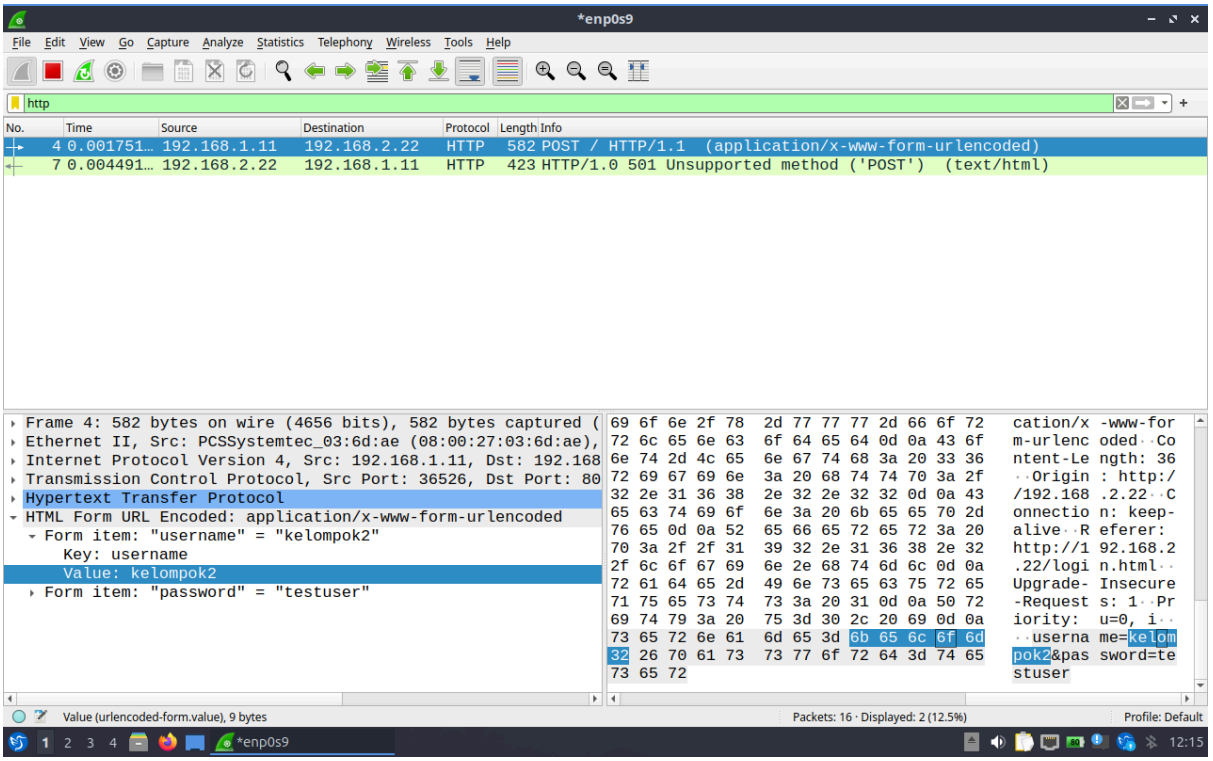
Desain Topologi:



Dapat dilihat kalau yang menghubungkan antara port vm router dan vm server (sebagai pemilik web server) ialah enp0s9, sehingga port ini lah yang akan kita pilih untuk di *capture traffic* nya.



Sudah terlihat kalau ada beberapa paket yang masuk ke dalam vm server, namun karena target kita minggu ini ialah HTTP, maka kita akan filter berdasarkan HTTP.



Nah, sudah terlihat bukan, bahwa protokol HTTP yang masuk ke dalam vm server via POST dari IP source 192.168.1.11, yaitu vm klien (berdasarkan topologi di atas), terlihat plaintext nya yang mengandung text username dan password yang dimasukkan.

Hal ini membuktikan bahwa protokol HTTP yang sering digunakan terlihat tidak aman, karena perangkat yang tidak ada keterlibatan dalam transfer data, dan hanya berada di jaringan yang sama, alias *man in the middle,* dapat dengan mudahnya mendeteksi dan melihat isi dari data transferan antara dua belah pihak.

Mungkin itu saja yang dapat kami sampaikan, kita ketemu lagi pada minggu depan saat kami membahas progress minggu ke tiga: **Mengaktifkan HTTPS & HSTS**. *See ya there, amigos, peace!*

- Kelompok 2