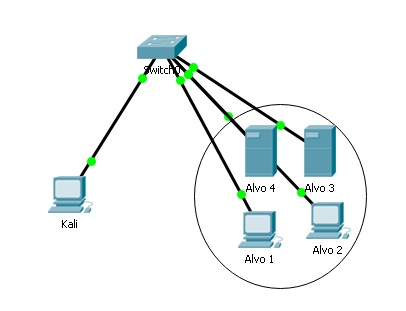
**ATIVIDADE 3 - CAPTURA HTTP E HTTPS:**

**Nome: Igor Correa RA: 15.00588-7**

**Nome: Rodrigo Franciozi RA: 14.04014-0  
Nome: RA:**

1. Topologia – Cenário 1:



1. Topologia – Cenário 2:
2. 
3. Objetivos

Parte I: Preparando as máquinas virtuais

Parte II: Captura e visualização de um tráfego HTTP (cenário 1); Captura e visualização de um tráfego HTTPS externo (cenário 2); Captura e visualização de um tráfego HTTPS (certificado auto-assinado) com máquinas virtuais (cenário 2)

**Formulário:**

**a. Configurando a Rede no Linux:**

a.1. Configurando IP e Máscara (manual por linha de comando)

# ifconfig -a // verificando todas as interfaces no sistema

# ifconfig < interface > <IP > //configuração da rede no Linux

# ifconfig eth0 10.32.0.10 netmask 255.255.255.0 //configuração de endereço IP 10.32.0.10 com a máscara 255.255.255.0 usando a interface de rede eth0 (a interface pode ser enp0s1 ou outras)

# ifconfig eth0 10.32.0.10/24 //configuração de endereço IP 10.32.0.10 com a máscara 255.255.255.0 usando a interface de rede eth0

# ifconfig // verificando endereço que foi configurado no sistema Linux

a.2. Iniciar/para/reiniciar o serviço:

# /etc/init.d/networking start

# /etc/init.d/networking stop

# /etc/init.d/networking restart

a.3. Habilitando/desabilitando interface de rede:

# ifconfig eth0 down

# ifdown eth0

# ifconfig eth0 up

# ifup eth0

**b. Comandos com OpenSSL:**

# openssl help // mostra informações dos comandos básicos, cifras criptográficas e algoritmos de hash

# openssl version -v // mostra a versão do openssl

# openssl req -x509 -nodes -days 365 -sha256 -newkey rsa:2048 -keyout mykey.pem -out mycert.pem // cria um certificado no padrão x.509, rsa de 2048 bits, hash sha256 e certificado gerado.

# cat mycert.pem // visualizando o arquivo do certificado criado

# cat mykey.pem // visualizando a chave privada criada referente ao certificado

# openssl s\_server -cert mycert.pem -key mykey.pem -accept 443 -www // testando o certificado auto-assinado

# openssl s\_cliente -connect 10.32.0.20:443 // conectando com o servidor usando o s\_client

**Parte I – Preparando as Máquinas Virtuais:**

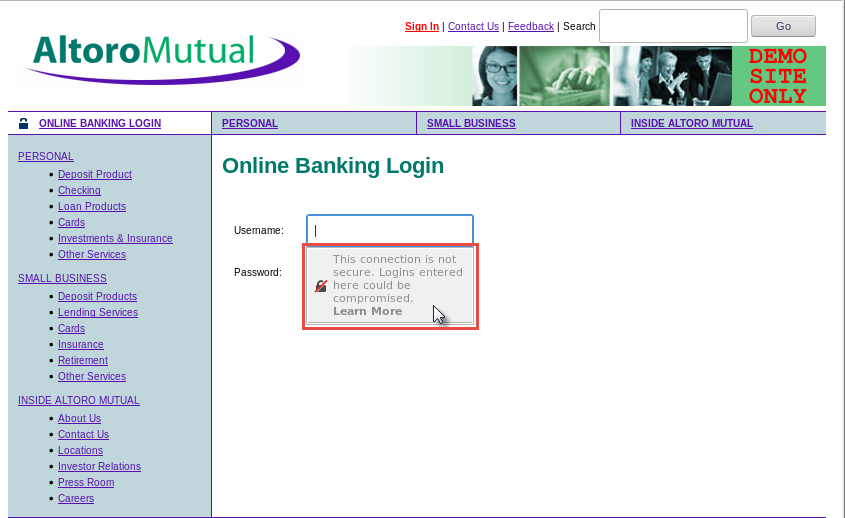
* 1. Deve-se escolher 3 máquinas virtuais para teste (endereço de rede será substituído pelo RA), por exemplo, 12.00822-9 🡪 12.00.82.29 🡪 12.0.82.0 – endereço de rede). Os endereços IPs serão escolhidos valores que compreendem nessa faixa, por exemplo, 12.0.82.10, 12.0.82.20, 12.0.82.30 e a máscara será 255.255.255.0.
  2. Kali Linux para realização dos testes (vetor de ataque);
  3. Uma máquina virtual Linux como alvo (Metasploitable2).

**Parte II – Captura de Tráfego HTTP, HTTPS e Análise com Wireshark:**

1. Cenário **http** com acesso ao servidor http (cenário 1). Considere o seguinte procedimento:

a. Iniciar a captura com o Wireshark 🡪 acessar a seguinte página http:

[**www.altoromutual.com/bank/login.aspx**](http://www.altoromutual.com/bank/login.aspx)



a. Entrar com o usuário **Admin** e a senha **Admin** e clique em **Login**.

b. Verificar se é possível capturar nome de **usuário** e **senha** usando Wireshark? Se sim, como? Demonstrar com o **print** das telas no Wireshark. Dica: usar a seguinte sequência: Analyse 🡪 Follow 🡪 TCP Stream

c. Esse protocolo é considerado seguro? Justifique.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

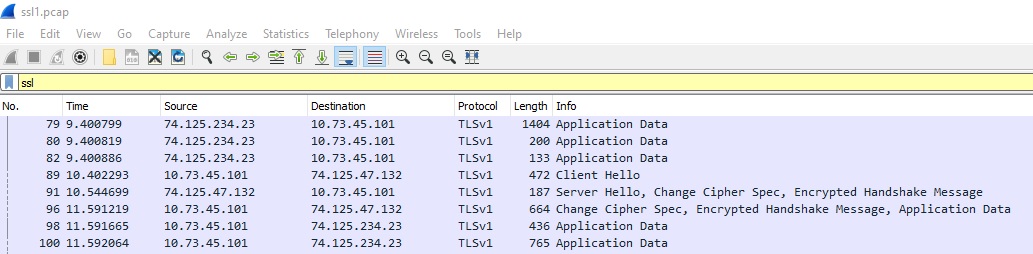
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

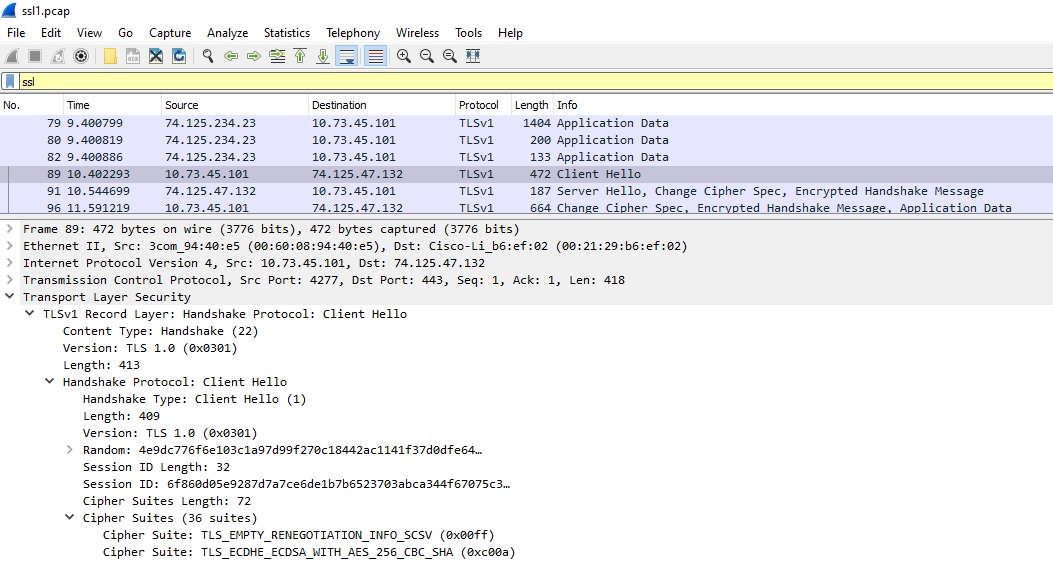
2. Cenário **https** com acesso ao servidor **https** (Cenário 1). Considere o seguinte procedimento:

a. Iniciar a captura com o Wireshark 🡪 acessar uma página https (Office 365 Mauá ou outro local https) 🡪 finalizar a conexão da página 🡪 finalizar a captura com o Wireshark.

b. Após a captura no Wireshark, filtre o tráfego https com **SSL** ou **TLS**.



c. Identifique o conjunto de cifras com as sinalizações **Client Hello** e **Server Hello**.



c.1. A partir de uma captura de dados (Wireshark) usando SSL/TLS, pede-se:  
a. Identificar 3 séries de cifras usadas na sinalização “**client hello**”.

**Cipher Suite: TLS\_AES\_128\_GCM\_SHA256 (0x1301)**

**Cipher Suite: TLS\_CHACHA20\_POLY1305\_SHA256 (0x1303)**

**Cipher Suite: TLS\_AES\_256\_GCM\_SHA384 (0x1302)**

c.2. Qual a série de cifra usada na sinalização “**server hello**” considerando algoritmo de chave simétrica, assimétrica e hash envolvidos para essa sinalização?

**Cipher Suite: TLS\_AES\_256\_GCM\_SHA384 (0x1302)**

c.3. Quais IPs (cliente e servidor) e as portas associadas (cliente e servidor). Qual a versão do protocolo TLS? Existe alguma vulnerabilidade conhecida? Justifique.

**Cliente🡪10.0.2.15:59836**

**Servidor🡪152.199.54.34:443**

**TLSv1.3**

**CVE-2016-8612**

3. Cenário **https** com acesso ao certificado **https** (Cenário 2). Considere o seguinte procedimento:

a. Certificado com **Ubuntu Server** (criando certificado auto-assinado):

$ openssl req -x509 -nodes -days 365 -sha256 -newkey rsa:2048 -keyout mykey.pem -out mycert.pem

- Country Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- Sate or Province Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- Locality Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- Organization Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

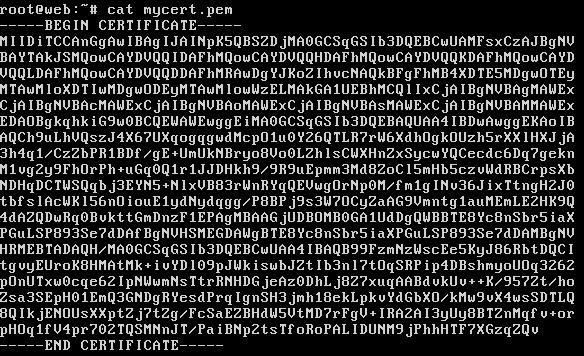
- Organizational Unit Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- Common Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- Email Address: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. Verificando certificado e chave criada:

$ cat mycert.pem



$ cat mykey.pem



c. Testando o certificado criado (Ubuntu Server):

$ openssl s\_server -cert mycert.pem -key mykey.pem -accept 443 -www



d. Conectando o cliente (**Kali**) com o (**Ubuntu Server**) para testar o certificado:

- Abrir o browser no cliente (Kali) e digitar no browser o endereço IP do servidor conforme ajuste de RA (Ubuntu Server) **https://IP , por exemplo, https://10.32.0.20**

d.1 A partir de uma captura de dados (Wireshark) usando SSL/TLS, pede-se:  
d.1.1. Identificar 3 séries de cifras usadas na sinalização “**client hello**”.

**Cipher Suite: TLS\_AES\_128\_GCM\_SHA256 (0x1301)**

**Cipher Suite: TLS\_CHACHA20\_POLY1305\_SHA256 (0x1303)**

**Cipher Suite: TLS\_AES\_256\_GCM\_SHA384 (0x1302)**

d.1.2. Qual a série de cifra usada na sinalização “**server hello**” considerando algoritmo de chave simétrica, assimétrica e hash envolvidos para essa sinalização?

Cipher Suite: TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA (0x002f)

d.1.3. Quais IPs (cliente e servidor) e as portas associadas (cliente e servidor). Qual a versão do protocolo TLS? Existe alguma vulnerabilidade conhecida?

**Cliente🡪10.32.0.10:33706**

**Servidor🡪10.32.0.30:443**

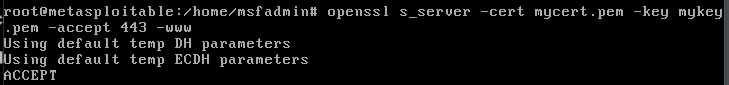
**TLSv1**

**CVE-2016-2852**

4. Repetir o item 3 usando como certificado o **Metasploitable2**. Analisar as diferenças entre as versões do protocolo TLS. Existem vulnerabilidades conhecidas? Justifique.







**Referências:**

1. STALLINGS, W. Criptografia e Segurança de Redes - Princípios e Práticas - 6ed., Pearson, 2015.

3. Documentação Oficial - OpenSSL:

<https://www.openssl.org/>