HTTP-Basic: Apache

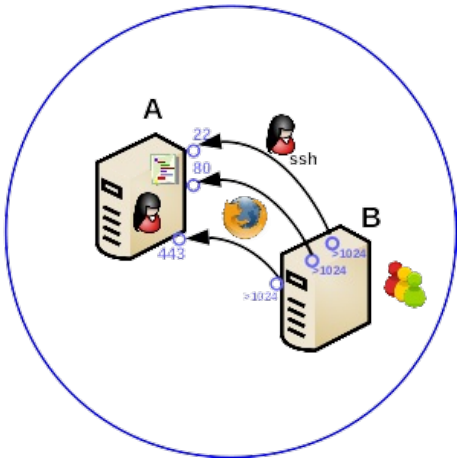
ESCENARIO

Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0

Máquina virtual A:
Rede Interna
Servidor SSH: openssh-server
Servidor Web: Apache (apache2)
ISO: Kali Live amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client (ssh)
Cliente Web: Navegador (firefox)
ISO: Kali Live amd64
IP/MS: 192.168.120.101/24

BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB



LIMITACIÓN DE RESPONSABILIDADE O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

NOTAS:

- Prerrequisitos:
  - **Servizo Web: Apache**
  - **Cifrado asimétrico: Certificado Apache**
- Servidor Web Apache:
  - Paquete apache2 (# apt update && apt -y install apache2).
  - **Nomenclatura versións: 2.X.revision**, onde:
    - X toma valor par ➜ a versión é estable
    - X toma valor impar ➜ a versión é de desenvolvemento
  - Cliente ssh GNU/Linux: **comando ssh (paquete openssh-client)**
  - Servidor SSH GNU/Linux: Paquete **openssh-server** (# apt update && apt -y install openssh-server).
    Ficheiro de configuración: **/etc/ssh/sshd_config (man sshd_config)**

Resumo Prácticas Exemplos

- No **Exemplo1. Control de acceso** imos tratar o tipo de control de acceso: autentificación http basic e os arquivos tipo **.htaccess**.

  HTTP proporciona un método de autenticación básico de usuarios: basic. Este método ante unha petición do cliente(navegador web) ao servidor cando se solicita unha URL amosará un diálogo pedindo usuario e contrasinal. Unha vez autenticado o usuario, o cliente volverá facer a petición ao servidor pero agora enviando o usuario e contrasinal, en texto claro (sen cifrar) proporcionados no diálogo. É recomendable entón se se emprega este método que se faga combinado con conexión SSL (HTTPS).

- No **Exemplo2. Sniffer tcpdump** imos capturar os paquetes http tidos lugar entre ás máquinas cliente e servidor. Revisados estes paquetes poderemos comprobar que a comunicación HTTP ten lugar sen cifrar e o HTTP Basic envía a información do usuario/contrasinal codificada en base64. Deste xeito, descodificando o base64 podemos saber o usuario/contrasinal empregados na comunicación.

- No **Exemplo3. Sniffer Wireshark** imos realizar de novo o Exemplo2 pero agora co programa Wireshark. A diferencia do tcpdump seremos capaz de seguir a comunicación de forma gráfica mediante a opción Follow do paquete seleccionado e unha vez atopada a codificación base64 do usuario/contrasinal o propio wireshark xa amosa tamén esa información descodificada.

- No **Exemplo4. HTTPS e Sniffer tcpdump** imos comprobar que non seremos quen de visualizar en texto claro o usuario/contrasinal empregados na Autenticación HTTP Basic, debido ao emprego do procotolo **HTTPS**.

- No **Exemplo5. HTTPS e Sniffer Wireshark** imos comprobar que non seremos quen de visualizar en texto claro o usuario/contrasinal empregados na Autenticación HTTP Basic, debido ao emprego do procotolo **HTTPS**.

Servizo Web - Apache

Máquina virtual A: Kali amd64

1. Na contorna gráfica abrir un terminal e executar:
   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ passwd kali #Cambiar o contrasinal do usuario kali. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final).

2. Cambiar hostname da máquina virtual A. Por kaliA como hostname:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# echo 'kaliA' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kali:~# echo 'kernel.hostname=kaliA' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
   kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario **kali**.

3. Configurar a rede:
   Na contorna gráfica abrir un terminal e executar:

   kali@kaliA:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kaliA:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@kaliA:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a configuración de rede e non ter conflicto con este xestor.
   root@kaliA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliA:~# ip addr add 192.168.120.100/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.100 e máscara de subrede: 255.255.255.0.
   root@kaliA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliA:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0

4. Comprobar estado do Servidor SSH:
   root@kaliA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.
   root@kaliA:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   root@kaliA:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.
   root@kaliA:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.
   root@kaliA:~# /etc/init.d/ssh start #Arrancar o servidor SSH.
   root@kaliA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.
   root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
   root@kaliA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)

root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
root@kaliA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled
root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
root@kaliA:~# ssh -v kali@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario kali e o seu contrasinal. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.
kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.
root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kaliA:~$

Máquina virtual B: Kali amd64

5. Configuración da rede. Na contorna gráfica abrir un terminal e executar:
   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@kali:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a configuración de rede e non ter conflicto con este xestor.
   root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
   root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
   root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
   root@kali:~# echo '192.168.120.100 kaliA' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome kaliA, para que atenda á IP 192.168.120.100
   root@kali:~# ping -c4 kaliA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A

6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:
   root@kali:~# echo 'kaliB' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kali:~# echo 'kernel.hostname=kaliB' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
   kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario **kali**.

7. **B** $\overset{\text{SSH}}{\rightarrow}$ **A** Acceder mediante SSH dende a máquina virtual B á máquina virtual A. Dende agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:
   Na contorna gráfica abrir un terminal e executar:

   kali@kaliB:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   kali@kaliB:~$ nc -vz kaliA 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   kali@kaliB:~$ ssh -v kali@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario **kali** a través da conexión cifrada SSH.
      kali@kaliA:~$

8. Activar Servidor Web Apache:
      kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
      root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
      root@kaliA:~# /etc/init.d/apache2 start #Iniciar o servidor web Apache.
      root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
      root@kaliA:~# nc -vz 192.168.120.100 80 #Mediante o comando nc(netcat) comprobar se o porto 80 do servidor web Apache está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 80 é o porto TCP a escanear.

No caso da distribución Kali xa temos instalado o servidor web Apache, pero nunha distribución baseada en Debian poderiamos instalalo do seguinte xeito:

```
# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
# apt search apache2 #Buscar calquera paquete que coincida co patrón de búsqueda apache2
# apt -y install apache2 #Instalar o paquete apache2, é dicir, instalar o servidor HTTP apache2. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

9. Lanzar na máquina virtual B (Kali) un navegador e visitar a IP 192.168.120.100 ou a URL http://192.168.120.100

10. Permisos apache:

```
root@kaliA:~# chown -R www-data. /var/www/html/ #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio DocumentRoot de Apache: /var/www/html
root@kaliA:~# chmod 444 /var/www/html/index.html #Cambiar a só lectura os permisos ugo do ficheiro index.html situado en /var/www/html, é dicir, establecer os permisos r--r--r-- (soamente lectura para o usuario propietario, o grupo propietario e o resto do mundo)
root@kaliA:~# /etc/init.d/apache2 restart #Reiniciar o servidor web Apache.
root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
```

11. Actualizar na máquina virtual B (Kali) a páxina referente á URL http://192.168.120.100

12. **Exemplo1. Control de acceso por HTTP Basic.**

Na autenticación HTTP Basic é moi típico utilizar **arquivos .htaccess** nos directorios que queremos controlar o acceso. Os arquivos .htaccess son ficheiros de configuración do propio directorio onde exista.

Para usar arquivos .htaccess, necesítase ter unha configuración no servidor que permita poñer directivas de autenticación nestes arquivos, mediante a directiva AllowOverride, tal como segue: AllowOverride AuthConfig

> **NOTA:** Visitar o seguinte enlace para ver unha explicación, máis polo miúdo, sobre á autenticación http basic: **Autenticación y autorización**
> 

Procedemento:

1. Modificar arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

   <Directory /var/www/html/auth-empresa>
   AllowOverride Authconfig
   </Directory>

2. Crear o contrasinal para o usuario ana no ficheiro de contrasinais /etc/apache2/web.htpasswd:

   root@kaliA:~# htpasswd -c /etc/apache2/web.htpasswd ana #Pór **123456** como contrasinal do usuario *ana*

3. Crear o contrasinal para o usuario brais no ficheiro de contrasinais /etc/apache2/web.htpasswd:

   root@kaliA:~# htpasswd /etc/apache2/web.htpasswd brais #Pór **654321** como contrasinal do usuario *brais*

4. Configuralo servidor para o acceso sexa permitido mediante autenticación: usuario/contrasinal empregando un arquivo .htaccess:

   root@kaliA:~# cat /root/.htaccess #Amosar contido arquivo .htacesss

   AuthType Basic
   AuthName "Web con Autenticacion Basic"
   AuthBasicProvider file
   AuthUserFile /etc/apache2/web.htpasswd
   ##Require valid-user
   Require user ana

5. Xerar o directorio /var/www/html/auth-empresa, o ficheiro secret.txt dentro deste e mover o arquivo .htaccess, onde establecemos os permisos, para que Apache poida acceder a ese ficheiro secret.txt

   root@kaliA:~# mkdir /var/www/html/auth-empresa #Crear o directorio /var/www/html/auth-empresa
   root@kaliA:~# echo 'S3c3eT contido' > /var/www/html/auth-empresa/secret.txt Crear o ficheiro /var/www/html/auth-empresa/secret.txt co contido: *S3cr3T contido*
   root@kaliA:~# mv /root/.htaccess /var/www/html/auth-empresa/ #Mover o arquivo /root/.htaccess ao directorio /var/www/html/auth-empresa/
   root@kaliA:~# chown -R www-data. /var/www/html/auth-empresa #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio /var/www/html/auth-empresa
   root@kaliA:~# chmod 400 /var/www/html/auth-empresa/.htaccess #Cambiar a só lectura os permisos **ugo** do ficheiro .htaccess situado en /var/www/html/auth-empresa, é dicir, establecer os permisos r-------- (soamente lectura para o usuario propietario)

6. Actualizar a configuración de Apache para ter en conta os novos cambios:

   root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache

7. Actualizar o arquivo /etc/hosts no cliente kaliB:

root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kaliA:~$ exit #Saír da consola remota ssh na que estamos a traballar, para voltar á consola local do usuario **kali** na máquina kaliB.
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@kaliB:~# echo '192.168.120.100 auth-empresa.local' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome auth-empresa.local para que atenda á IP 192.168.120.100
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kaliB:~$

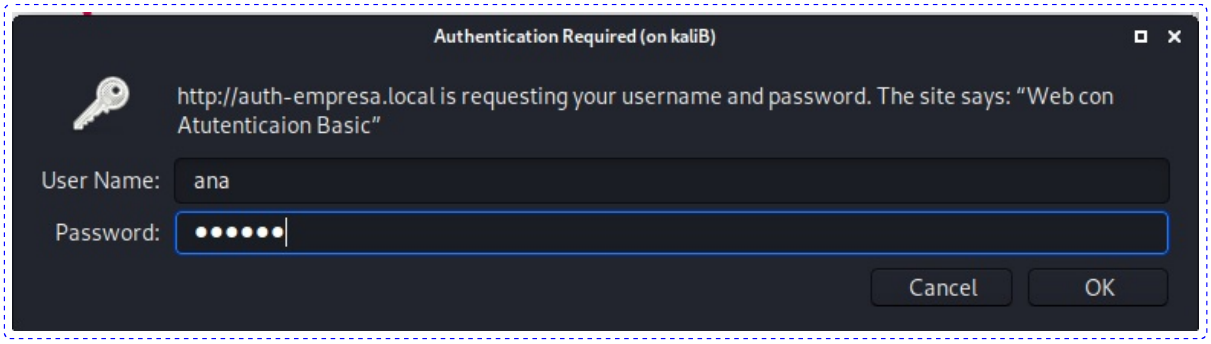
8. Acceder dende o equipo cliente kaliB ás seguintes direccións web:

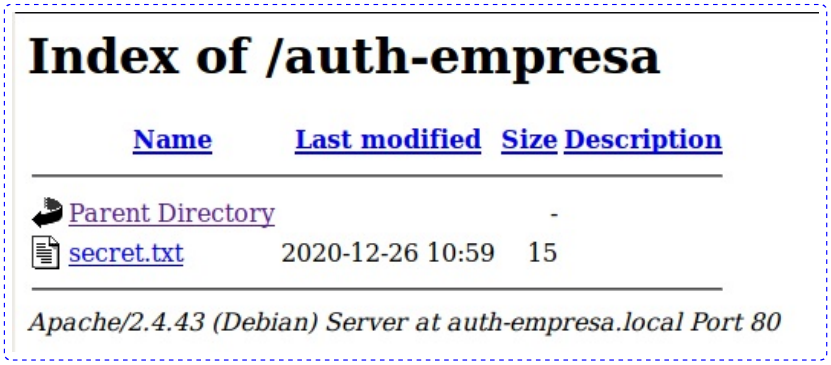
http://auth-empresa.local
http://auth-empresa.local/auth-empresa

IMPORTANTE: Como estamos a empregar o sitio por defecto de Apache (DocumenRoot ➜ /var/www/html), aínda que configuramos a autenticación, durante todo o proceso o arquivo secret.txt antes de recargar o servidor estivo visible e accesible. Mellor sería ter configurado isto mediante VirtualHost, tal que non se activaría o sitio ata que executaramos o comando a2ensite correspondente:

a2ensite + reload ➜ Unha ver activado o sitio(a2ensite) e recargado(reload) o servidor a páxina(VirtualHost) carga.

9. Introducir usuario e contrasinal no formulario HTTP Basic





13. **Exemplo2. Sniffer tcpdump.**

Imos realizar de novo o Exemplo1 pero agora empregaremos na máquina virtual KaliA(servidor Web) un sniffer para revisar a comunicación vía web que se establece entre as máquinas cliente e servidor.

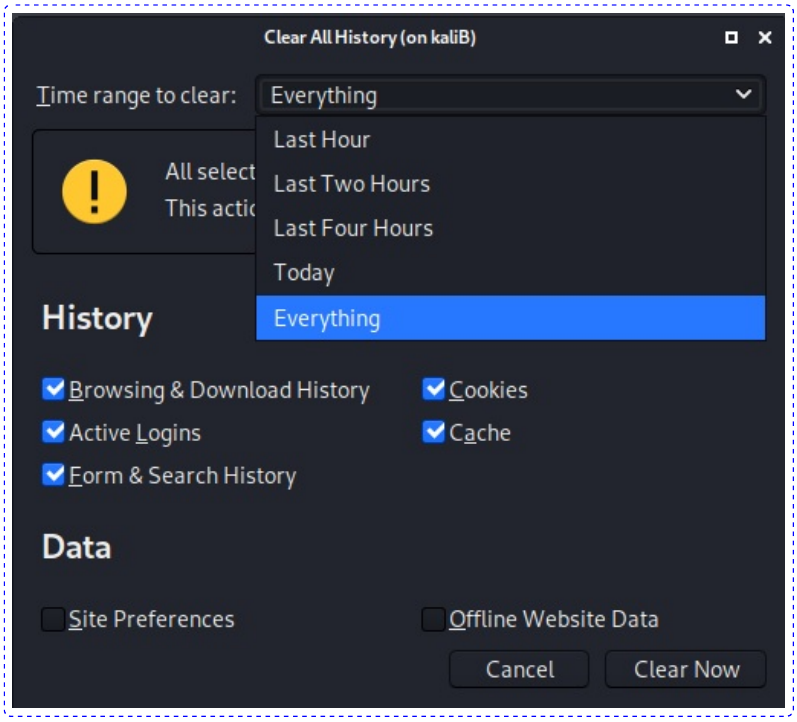
Procedemento:

1. No Servidor Web(kaliA) capturar os paquetes que teñen lugar a través do porto TCP 80 nun ficheiro para posterior consulta:

```
kali@kaliB:~$ ssh kali@192.168.120.100 #Acceder como o usuario kali a través da conexión cifrada SSH.
kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@kaliA:~# tcpdump 'tcp port 80' -vv -w file.pcap #Capturar no ficheiro file.pcap os paquetes que teñen lugar co porto TCP 80
```

2. Acceder ao navegador do equipo cliente kaliB e eliminar as cookies/historial:

   Premer as teclas Shift+Ctrl+Supr
   Escoller Everything
   Premer en Clear Now



3. Acceder dende o equipo cliente kaliB á seguinte dirección web:

   http://auth-empresa.local/auth-empresa

4. Introducir usuario/contrasinal no formulario de autenticación HTTP Basic solicitado e revisar como a consola onde temos lanzado o comando **tcpdump** está a capturar paquetes.

5. Unha vez que accedamos á páxina tras a correcta autenticación paramos a execución do comando **tcpdump** premendo Ctrl+C

```
root@kaliA:~# tcpdump 'tcp port 80' -vv -w file.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), captur
e size 262144 bytes
^C36 packets captured
36 packets received by filter
0 packets dropped by kernel
root@kaliA:~#
```

6. Agora revisamos o ficheiro coa captura de paquetes

   root@kaliA:~# tcpdump -vv -r file.pcap | less #Revisar o capturado no ficheiro file.pcap cos paquetes que tiveron lugar co porto TCP 80

7. Buscar o bloque correspondente a introducir usuario/contrasinal. Podemos ver que temos unha cadea en base64 a cal podemos descodificar obtendo usuario/contrasinal en texto claro

   root@kaliA:~# echo 'cadeaAtopada' | base64 -d #Descodificar cadea base64, é dicir, visualizar en texto claro o usuario e contrasinal empregados na Autenticación HTTP Basic.

```
    kaliB.33176 > kaliA.http: Flags [P.], cksum 0x4c1a (correct
), seq 1:371, ack 1, win 502, options [nop,nop,TS val 288195121
3 ecr 23399950], length 370: HTTP, length: 370
        GET /auth-empresa/ HTTP/1.1
        Host: auth-empresa.local
        User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Ge
cko/20100101 Firefox/68.0
        Accept: text/html,application/xhtml+xml,application/xml
;q=0.9,*/*;q=0.8
        Accept-Language: en-US,en;q=0.5
        Accept-Encoding: gzip, deflate
        Connection: keep-alive
        Upgrade-Insecure-Requests: 1
        Authorization: Basic YW5hOjEyMzQ1Ng==

12:09:08.307648 IP (tos 0x0, ttl 64, id 15325, offset 0, flags
[DF], proto TCP (6), length 52)
    kaliA.http > kaliB.33176: Flags [.], cksum 0x7241 (incorrec
t → 0x2656), seq 1, ack 371, win 507, options [nop,nop,TS val
:
```

```
root@kaliA:~# tcpdump -vv -r file.pcap | less
reading from file file.pcap, link-type EN10MB (Ethernet)
^[[Broot@kaliA:~# echo 'YW5hOjEyMzQ1Ng==' | base64 -d
ana:123456root@kaliA:~#
```

14. **Exemplo3. Sniffer Wireshark.**

Imos realizar de novo o Exemplo2 pero agora empregaremos para a lectura do ficheiro capturado file.pcap o sniffer Wireshark, para revisar a comunicación vía web que se establece entre as máquinas cliente e servidor.

Procedemento:

1. No Servidor Web(kaliA) temos capturado o ficheiro **file.pcap** (ver Exemplo2), o cal imos copialo na casa do usuario *kali*:

```
root@kaliA:~# cp -pv file.pcap /home/kali#Copiar preservando permisos(-p) e en modo detallado(-v) o ficheiro file.pcap dentro do directorio /home/kali
root@kaliA:~# chown kali. /home/kali/file.pcap #Cambiar usuario propietario kali e grupo propietario kali ao ficheiro /home/kali/file.pcap
root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
kali@kaliA:~$ exit #Pechar o terminal saíndo da consola local do usuario kali para voltar á consola local do usuario kali no host kaliB.
kali@kaliB:~$
```

2. Copiar o ficheiro file.pcap no host kaliB

```
kali@kaliB:~$ scp kali@192.168.120.100:file.pcap . #Estando situado no HostB, copiar de A a B o arquivo /home/kali/file.pcap, é dicir, copiar en B o ficheiro file.pcap existente no HostA no cartafol da casa do usuario, que é o que simboliza o caracter ':'
```

3. Arrancar Wireshark:

```
kali@kaliB:~$ wireshark & #Lanzar o programa wireshark (sniffer) para poder visualizar o que acontece na rede (protocolos, paquetes). O caracter & serve para executar en segundo plano o programa e así devolver o prompt da consola para poder seguir traballando nela.
```

4. Abrir o arquivo file.pcap para o seu estudo:

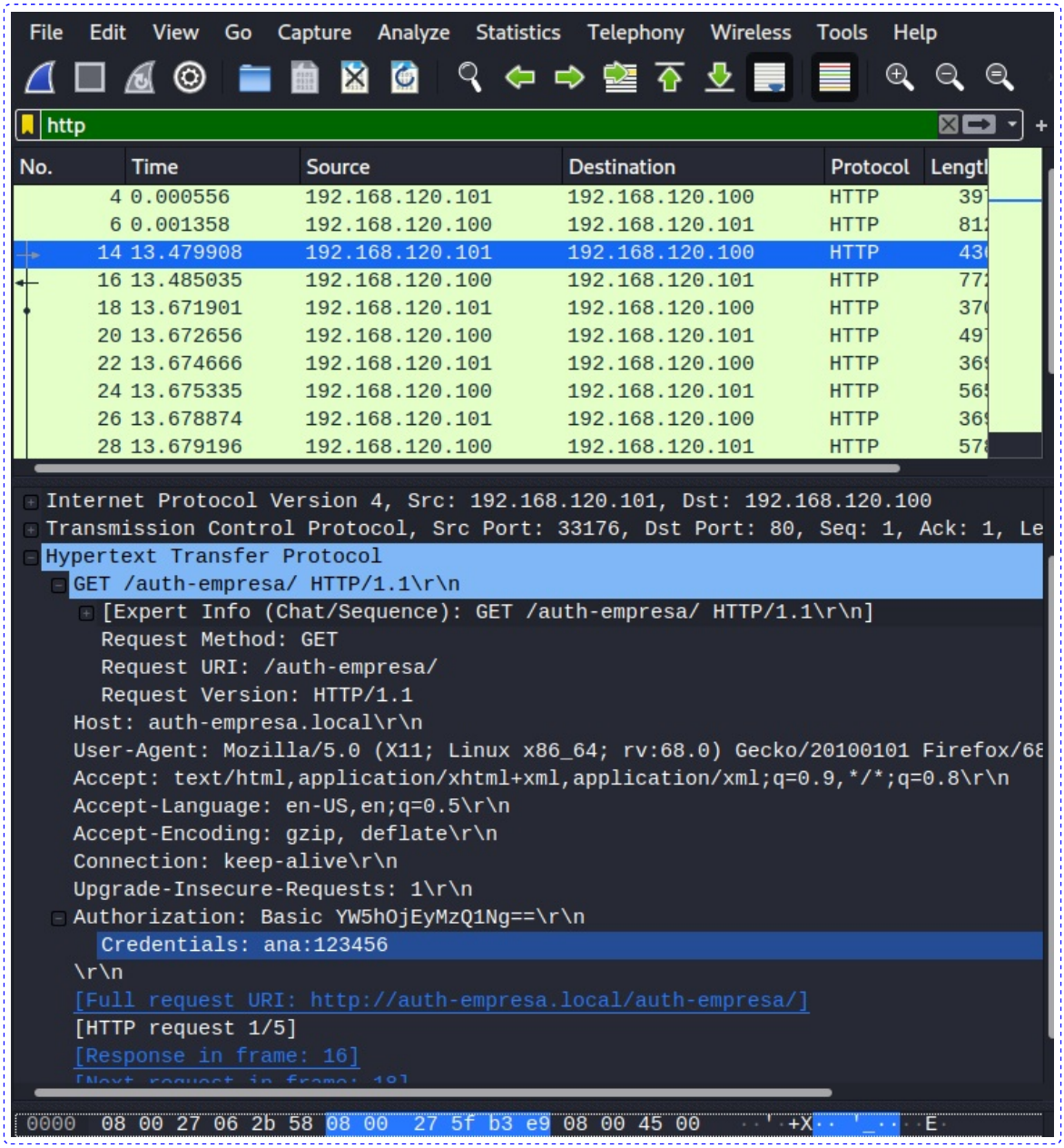
File ➜ Open ➜ /home/kali/file.pcap

5. Podemos buscar o bloque correspondente a introducir usuario/contrasinal de forma análoga a como fixemos co tcpdump(Ver Exemplo2), ou podemos empregar a mellora de funcionalidade que nos permite a GUI de Wireshark. Así:

   Filtramos por http e imos seleccionando os paquetes HTTP de orixe 192.168.120.101 que atopemos ➜ Na sección Hypertext Transfer Protocol atoparemos unha cadea en base64 a cal podemos descodificar obtendo usuario/contrasinal en texto claro:

```
kali@kaliB:~$ echo 'cadeaAtopada' | base64 -d #Descodificar cadea base64, é dicir, visualizar en texto claro o usuario e contrasinal empregados na Autenticación HTTP Basic.
```

   Tamén podemos facer clic no icono + que aparece á esquerda desa cadea e xa o Wireshark é quen de descodificala, amosando así o usuario e contrasinal empregados na Autenticación HTTP Basic.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Lengt
4 0.000556 192.168.120.101 192.168.120.100 HTTP 39
6 0.001358 192.168.120.100 192.168.120.101 HTTP 81
14 13.479908 192.168.120.101 192.168.120.100 HTTP 43
16 13.485035 192.168.120.100 192.168.120.101 HTTP 77
18 13.671901 192.168.120.101 192.168.120.100 HTTP 37
20 13.672656 192.168.120.100 192.168.120.101 HTTP 49
22 13.674666 192.168.120.101 192.168.120.100 HTTP 36
24 13.675335 192.168.120.100 192.168.120.101 HTTP 56
26 13.678874 192.168.120.101 192.168.120.100 HTTP 36
28 13.679196 192.168.120.100 192.168.120.101 HTTP 57
Internet Protocol Version 4, Src: 192.168.120.101, Dst: 192.168.120.100
Transmission Control Protocol, Src Port: 33176, Dst Port: 80, Seq: 1, Ack: 1, Le
Hypertext Transfer Protocol
GET /auth-empresa/ HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /auth-empresa/ HTTP/1.1\r\n]
Request Method: GET
Request URI: /auth-empresa/
Request Version: HTTP/1.1
Host: auth-empresa.local\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
Authorization: Basic YW5hOjEyMzQ1Ng==\r\n
Credentials: ana:123456
\r\n
[Full request URI: http://auth-empresa.local/auth-empresa/]
[HTTP request 1/5]
[Response in frame: 16]
0000 08 00 27 06 2b 58 08 00 27 5f b3 e9 08 00 45 00

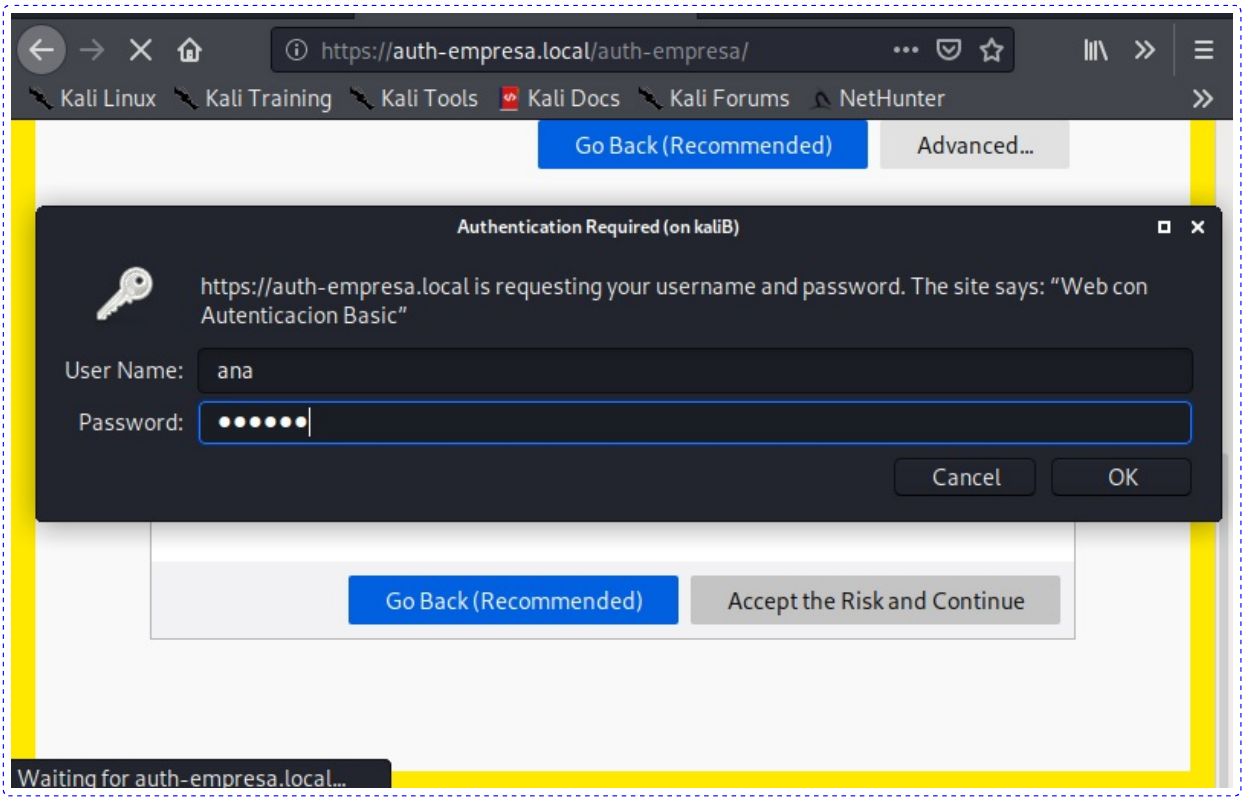
15. **Exemplo4. HTTPS e sniffer tcpdump.**

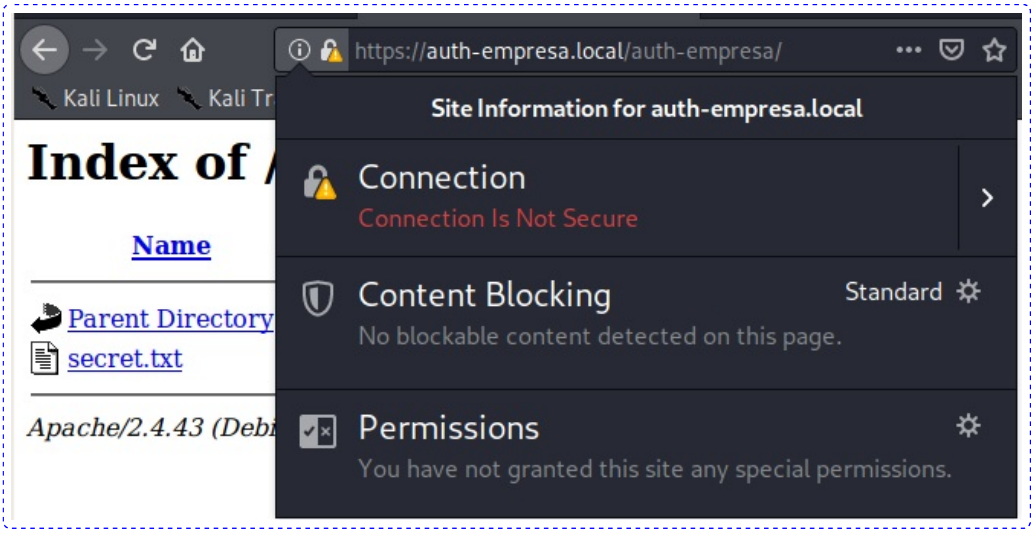
Prerrequisito:

- **Cifrado asimétrico: Certificado Apache**

Realizar de novo o Exemplo2 pero agora as peticións web faranse mediante o protocolo **HTTPS** e non HTTP.

Agora non seremos quen de visualizar en texto claro o usuario/contrasinal empregados na Autenticación HTTP Basic, debido ao emprego do procotolo **HTTPS**.





```
root@kaliA:~# tcpdump 'tcp port 443' -vv -w file2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), captur
e size 262144 bytes
^C109 packets captured
109 packets received by filter
0 packets dropped by kernel
root@kaliA:~#
```

```
    kaliB.49972 > kaliA.https: Flags [.], cksum 0×da33 (correct
), seq 1, ack 1, win 502, options [nop,nop,TS val 2885021236 ec
r 26439884], length 0
12:59:48.246416 IP (tos 0×0, ttl 64, id 14737, offset 0, flags
[DF], proto TCP (6), length 569)
    kaliB.49972 > kaliA.https: Flags [P.], cksum 0×7a65 (correc
t), seq 1:518, ack 1, win 502, options [nop,nop,TS val 28850212
46 ecr 26439884], length 517
12:59:48.246461 IP (tos 0×0, ttl 64, id 63070, offset 0, flags
[DF], proto TCP (6), length 52)
    kaliA.https > kaliB.49972: Flags [.], cksum 0×7241 (incorre
ct → 0×d816), seq 1, ack 518, win 506, options [nop,nop,TS val
 26439894 ecr 2885021246], length 0
12:59:48.249822 IP (tos 0×0, ttl 64, id 63071, offset 0, flags
[DF], proto TCP (6), length 1340)
    kaliA.https > kaliB.49972: Flags [P.], cksum 0×7749 (incorr
ect → 0×d3d0), seq 1:1289, ack 518, win 506, options [nop,nop,
TS val 26439898 ecr 2885021246], length 1288
12:59:48.250801 IP (tos 0×0, ttl 64, id 14738, offset 0, flags
:
```

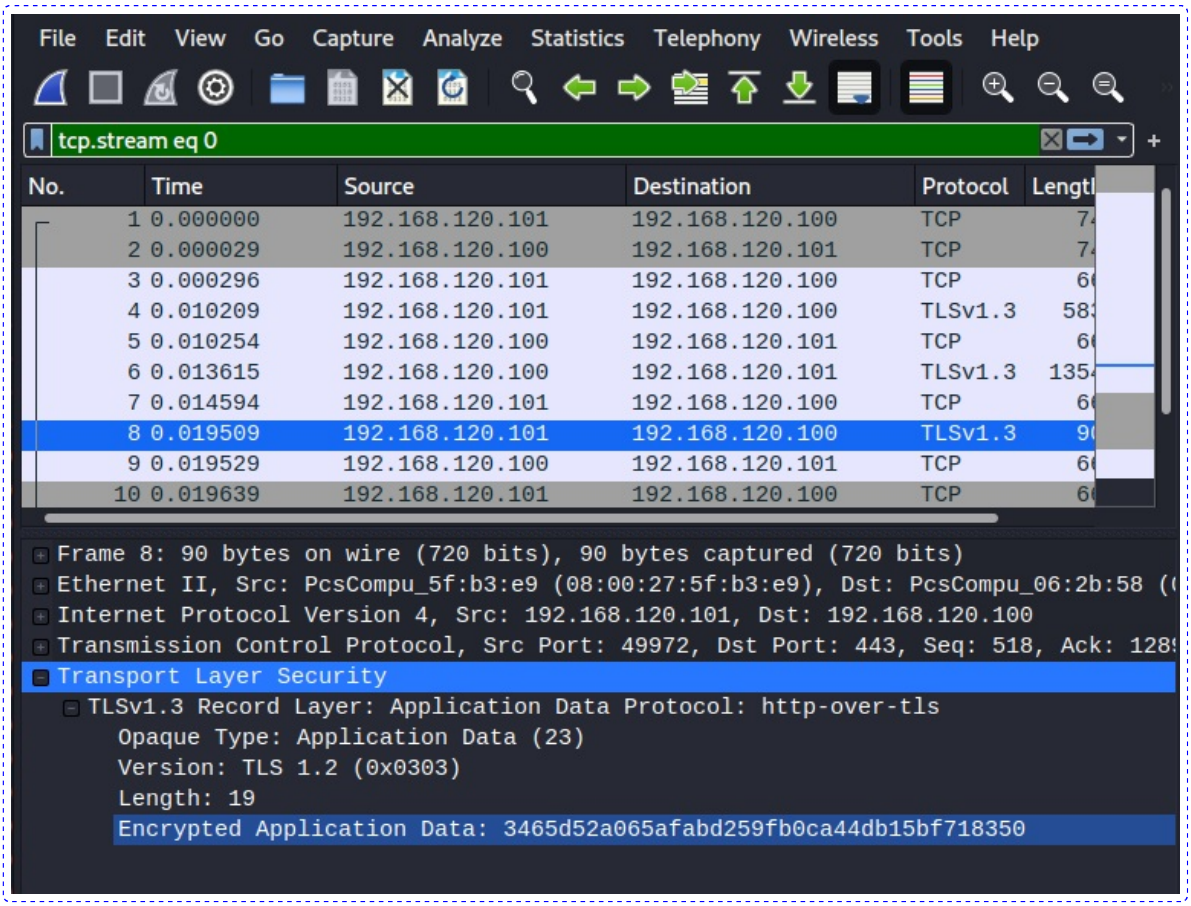
16. **Exemplo5. HTTPS e Sniffer Wireshark.**

    Prerrequisito:

    - **Cifrado asimétrico: Certificado Apache**

    Realizar de novo o Exemplo3 pero agora as peticións web faranse mediante o protocolo **HTTPS** e non HTTP.

    Agora non seremos quen de visualizar en texto claro o usuario/contrasinal empregados na Autenticación HTTP Basic, debido ao emprego do procotolo **HTTPS**.



Ricardo Feijoo Costa