Universidad de Buenos Aires
Facultad de Ingeniería
66.69 – Criptografía y Seguridad Informática

TP – Tunel IPSEC con Certificados

Objetivo

El objetivo de este trabajo práctico es la creación de una entidad certificante utilizando el conjunto de scripts (easy-rsa) incluido en el CD provisto.

Se desarrollan dos ejemplos:

1. Creación de una autoridad certificante
2. Mediante esta entidad se emitirán dos certificados, que serán utilizados en una conexión autenticada de ipsec.
3. Se anulara uno de los certificados generando la CRL correspondiente, la conexión no podrá establecerse .
4. Se emitirá un nuevo certificado de servidor para iniciar la conexión correctamente.
5. Se iniciara un servidor web con SSL

1- CREACION DE LA AUTORIDAD CERTIFICANTE:

1. Iniciar una PC con el CD provisto
2. Ubicarse en el directorio /crypto/ipsec-CA
   # cd /crypto/ipsec-CA
3. Editar el archivo /crypto/conf/config.sh y setear todas las variables, verificar las variables de entorno:
   - Numero de Grupo
   - KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG y KEY_EMAIL
4. Generar la AC ejecutando ***s01-generarCA.sh***

./clean-all	El primer paso (script clean-all) borra y regenera el directorio de certificados y genera los archivos: • serial Contiene una sola línea con el numero de serie del próximo certificado a generar. (“01”) • index.txt Se genera vació. Luego se va incorporando el hash y el cn de cada certificado generado.
./build-ca	Genera el certificado autofirmado de la Autoridad Certificante Completar el campo “Common Name” con un string que identifique a la entidad (p. ej. “TPGrupoX Criptografía CA”, el grupo será asignado por el profesor) Los demás campos se pueden dejar con su valor default. Luego de completar estos pasos tendremos en el directorio /crypto/var/rootCA los archivos ca.crt y ca.key, que corresponden al certificado y clave privada de la Autoridad Certificante respectivamente.

2- GENERACION DE SOLICITUDES DE CERTIFICADOS PARA R1 y R2:

Genera 2 pares de claves privadas y publicas, y 2 solicitudes de firma de las claves publicas que seran firmadas posteriormente por la CA.

s02-generaSolicitudes.sh

(ejecutado en el equipo1)

datos necesarios:

Información acerca de los dueños de los certificados. Los campos se pueden dejar con el valor predeterminado, **excepto por el common name que debe ser el hostname del router correspondiente (r1 - r2)**.

Además se requerirá una clave para encriptar las claves privadas del par de claves, que debe ser utilizada en el paso11. Adicionalmente, para la parte 2 del TP será necesario recordar los datos no solo de la clave, sino también de la información del certificado del router 1.

Nota: Verificar que las fechas de R1 y R2 sean correctas (iguales).

detalles paso por paso:

- **rm -r -f $ROUTERS_PATH (/crypto/var/routers)**
  mkdir $ROUTERS_PATH
  Genera un directorio vació para guardar los datos de los certificados y los certificados mismos

- **$D/build-req-pass r1**
  Genera el par de claves y la petición de firma para el router 1

- **$D/build-req-pass r2**
  Genera el par de claves y la petición de firma para el router 2

s03-firmaSolicitudes.sh
(ejecutado en el equipo1)

resumen:
Con la clave privada de la autoridad certificante se firman las 2 peticiones, convirtiendo estas en certificados.

datos necesarios:
Es necesario confirmar la firma de los certificados 2 veces para cada uno.

detalles paso por paso:

- **$D/sign-req r1**
  Firma la solicitud del router 1

- **$D/sign-req r2**
  Firma la solicitud del router 2

- **cp $CAPATH/r1.crt $ROUTERS_PATH/r1.pem**
  cp $CAPATH/r2.crt $ROUTERS_PATH/r2.pem
  Copia los certificados al directorio generado previamente

- **mv $CAPATH/r1.key $ROUTERS_PATH**
  mv $CAPATH/r2.key $ROUTERS_PATH
  Mueve las claves privadas (encriptadas) al directorio generado previamente

Verificar y analizar el contenido de los directorios /crypto/var/rootCA y /crypto/var/routers

s04-generaCRL.sh
(ejecutado en el equipo1)

resumen:
Se genera una lista de certificados revocados, firmada por la misma autoridad certificante. Inicialmente se encuentra vacía.

datos necesarios:
automatico.

detalles paso por paso:

- **$D/make-crl crl.pem**
  Corre el script que genera la lista de certificados revocados

s05-getIDs.sh
(ejecutado en el equipo1)

resumen:
Obtiene información de los dueños de los certificados desde el campo Subject de los certificados generados, Esta informacion de identificacion se utilizara posteriormente para generar el archivo de configuración IPsec de ambos routers.

datos necesarios:
automatico.

detalles paso por paso:

- **idleft=`grep "Subject:" $ROUTERS_PATH/r1.pem |cut -d: -f 2`**
  idright=`grep "Subject:" $ROUTERS_PATH/r2.pem |cut -d: -f 2`
  Crea las variables idleft e idright cuyo contenido son los datos de los dueños de los certificados

- **rm -f $OPERATING_PATH/ids**
  (
  echo " leftid=\"${idleft:1}\""
  echo " rightid=\"${idright:1}\""
) > $OPERATING_PATH/ids
  Crea un archivo cuyo contenido son las variables generadas previamente

3- CONFIGURACION DE TUNEL IPSEC UTILIZANDO CERTIFICADOS

CONFIGURACION DE R1:

s00-configurarInterfacesLocales.sh

(ejecutado en el equipo1)

resumen:
Configura las interfaces del equipo1, llamando local al equipo donde reside la autoridad certificante.

datos necesarios:
automático.

detalles paso por paso:

- **killall pump**
  Elimina el proceso que obtiene la configuración de las interfaces mediante DHCP, para evitar que estas sean reconfiguradas

- **ifconfig $R1_PRIV_INTERFAZ $R1_PRIV_IP netmask $R1_PRIV_MASCARA**
  Configura la interfase publica y la mascara correspondiente

- **ifconfig $R1_PUB_INTERFAZ $R1_PUB_IP netmask $R1_PUB_MASCARA**
  Configura la interfase privada y la mascara correspondiente

s06-instalaCertificados.sh

(ejecutado en el equipo1)

resumen:

Copia los certificados, la autoridad certificante y la lista de certificados revocados a las ubicaciones correspondientes para el programa freesWan que genera el túnel IPsec.

datos necesarios:
automático.

detalles paso por paso:

- **mkdir -p $IP_CERTS_PATH/certs**
  rm -f $IP_CERTS_PATH/cacerts/cacert.pem
  rm -f $IP_CERTS_PATH/certs/r1.pem
  rm -f $IP_CERTS_PATH/certs/r2.pem
  rm -f $IP_CERTS_PATH/private/r1.key
  rm -f $IP_CERTS_PATH/crls/crl.pem
  Borra cualquier certificado previo con el mismo nombre que exista en los directorios

- **cp -f $CAPATH/ca.crt $IP_CERTS_PATH/cacerts/cacert.pem**
  cp -f $CAPATH/crl.pem $IP_CERTS_PATH/crls
  mv -f $ROUTERS_PATH/r1.pem $IP_CERTS_PATH/certs
  mv -f $ROUTERS_PATH/r1.key $IP_CERTS_PATH/private
  Copia los certificados del router 1, el certificado de la autoridad certificante, y la lista de certificados revocados a los destinos correspondientes

s07-preparaArchivo.sh

(ejecutado en el equipo1)

resumen:
Prepara un archivo con la información necesaria para el equipo2 remoto: el certificado autofirmado de la autoridad certificante, el certificado firmado por la autoridad certificante, la lista de certificados revocados, y la información de los dueños de los certificados.

datos necesarios:
automático.

detalles paso por paso:

- **cp -f $CAPATH/ca.crt $ROUTERS_PATH/cacert.pem**
  cp -f $CAPATH/crl.pem $ROUTERS_PATH
  cp -f $OPERATING_PATH/ids $ROUTERS_PATH
  Copia los archivos necesarios para el host remoto a 1 directorio

- **zip $OPERATING_PATH/forr2.zip -r $ROUTERS_PATH**
  comprime el directorio en 1 solo archivo ZIP

- **rm -r -f $ROUTERS_PATH**
  borra el directorio

CONFIGURACION DE R2:

s08r-configurarInterfacesRemotas.sh
(ejecutado en el equipo2 remoto)

resumen:
Semejante al paso 0, configura las interfaces en el equipo2 remoto.

datos necesarios:
automático.

detalles paso por paso:

- **killall pump**
  Elimina el proceso que obtiene la configuración de las interfaces mediante DHCP, para evitar que estas sean reconfiguradas

- **ifconfig $R2_PRIV_INTERFAZ $R2_PRIV_IP netmask $R2_PRIV_MASCARA**
  Configura la interfase publica y la mascara correspondiente.

- **ifconfig $R2_PUB_INTERFAZ $R2_PUB_IP netmask $R2_PUB_MASCARA**
  Configura la interfase privada y la mascara correspondiente.

s09r-obtenerCertificado.sh
(ejecutado en el equipo2 remoto)

resumen:
Copia el archivo preparado en el equipo1 al equipo2 remoto, e instala los certificados a las ubicaciones correspondientes.

datos necesarios:
Confirmacion de la identidad del equipo al cual se conecta, y clave para entrar como usuario root: crypto.

detalles paso por paso:

- **scp root@$R1_PUB_IP:$OPERATING_PATH/forr2.zip $OPERATING_PATH**
  Copia del equipo1 al equipo2 remoto el archivo comprimido generado previamente

- **unzip -j $OPERATING_PATH/forr2.zip -d $ROUTERS_PATH**
  Descompacta el archivo recibido

- **mkdir -p $IP_CERTS_PATH/certs**
  rm -f $IP_CERTS_PATH/certs/r2.pem
  rm -f $IP_CERTS_PATH/private/r2.key
  rm -f $IP_CERTS_PATH/cacerts/cacert.pem
  rm -f $IP_CERTS_PATH/crl.pem
  Borra cualquier certificado previo con el mismo nombre

- **mv -f $ROUTERS_PATH/cacert.pem $IP_CERTS_PATH/cacerts**

mv -f $ROUTERS_PATH/r2.pem $IP_CERTS_PATH/certs
mv -f $ROUTERS_PATH/r2.key $IP_CERTS_PATH/private
mv -f $ROUTERS_PATH/ids $OPERATING_PATH
mv -f $ROUTERS_PATH/crl.pem $IP_CERTS_PATH/crls
Mueve los certificados a los destinos correspondientes

- **rm -f -r $ROUTERS_PATH**
  Borra el directorio donde se descompactó la información

GENERACION DE ipsec.conf en R1 y en R2:

s10c-generarIPsecConf.sh

(ejecutado en ambos equipos)

resumen:
Genera el archivo de configuración para generar el túnel IPsec y utilizar certificados en la autenticación.

datos necesarios:
automático.

detalles paso por paso:

1. Borra el previo archivo de configuración
2. Genera el /etc/ipsec.conf

- **(**
  # Configuracion General
  echo "config setup"
  echo " interfaces=\"ipsec0=$R1_PUB_INTERFAZ\""
  echo " klipsdebug=all"
  echo " plutodebug=all"
  echo " plutoload=%search"
  echo " plutostart=%search"
  echo " uniqueids=yes"
  echo " forwardcontrol=yes"
  echo ""
  # Conexion Default
  echo "conn %default"
  echo " keyingtries=0"
  echo " disablearrivalcheck=no"
  echo ""
  # Conexion R1-R2
  echo "conn crypto"
  echo " left=$R1_PUB_IP"
  echo " leftsubnet=$R1_PRIV_RED/$R1_PRIV_MASCARA"
  echo " right=$R2_PUB_IP"
  echo " rightsubnet=$R2_PRIV_RED/$R2_PRIV_MASCARA"
  echo " authby=rsasig"
  echo " auto=add"
  echo " leftrsasigkey=%cert"
  echo " rightrsasigkey=%cert"
  echo " leftcert=$IP_CERTS_PATH/certs/r1.pem"

echo " rightcert=$IP_CERTS_PATH/certs/r2.pem"
) > $ARCHIVO_IPSEC
genera un archivo de configuración nuevo para utilizar un túnel ESP con autenticación por certificados

- **cat $OPERATING_PATH/ids >> $ARCHIVO_IPSEC**
  Anexa al archivo de configuración los datos de la identidad de los usuarios, extraídos previamente en el paso 5

GENERACION DE ipsec.secrets en R1 y en R2:

s11c-generarIPsecSec.sh

(ejecutado en ambos equipos)

resumen:
Genera el archivo donde se guarda la clave necesaria para decodificar la clave privada del par de claves en el respectivo equipo.

datos necesarios:
la clave con la cual se encuentra encriptada la clave privada correspondiente, que se introdujo en el paso 2.

detalles paso por paso:

- **PASS_KEY=""**
  Inicializa la variable PASS_KEY

- ```
  until [ "$PASS_KEY" != "" ];
  do
    echo "Ingrese la clave del certificado correspondiente al host actual"
    read PASS_KEY
  done
  ```
  Obtiene un password desde el teclado

- **keyf=`find $IP_CERTS_PATH/private | grep key`**
  Obtiene el nombre del archivo que contiene el certificado (nota: tiene que existir 1 solo archivo de certificado, de lo contrario este script no funcionara correctamente)

- **rm -f $ARCHIVO_IPSEC_SEC**
  (
  echo ": RSA $keyf \"$PASS_KEY\""
) > $ARCHIVO_IPSEC_SEC
  Genera el archivo donde se guarda la clave de con la que se encuentra encriptada la clave privada del par de claves, cuya clave asociada publica figura en el certificado
  ```

Prueba del Túnel:

Una vez completado los pasos anteriores, se inicia el servicio de IPsec en ambos routers mediante el comando

/etc/init.d/ipsec start

Se levanta el túnel ejecutando el comando

ipsec auto --up crypto (en cualquiera de los routers.)

4- CRL: Anulación de Certificado de R1

srs0-revocarR1.sh

(ejecutado en el equipo2)

resumen:
Revoca el certificado inicialmente otorgado al router local (r1.pem), y actualiza la lista de certificados revocados, instalándola en el directorio correspondiente.

datos necesarios:
automático.

detalles paso por paso:

- **$D/revoke-full r1**
  Corre el script que revoca el certificado y genera una nueva lista de certificados revocados (CRL)

- **rm -f $IP_CERTS_PATH/crls/crl.pem**
  Borra la lista previamente instalada de certificados revocados

- **cp -f $CAPATH/crl.pem $IP_CERTS_PATH/crls**
  Instala la nueva CRL

srs1r-instalarCRLnuevo.sh

(ejecutado en el equipo2 remoto)

resumen:
Copia desde el equipo1 la lista de certificados revocados actualizada, y la instala en el directorio correspondiente.

datos necesarios:
Password de root del equipo1 (crypto).

detalles paso por paso:

- **rm -f $IP_CERTS_PATH/crls/crl.pem**
  Borra la lista previamente instalada de certificados revocados

- **scp root@$R1_PUB_IP:$CAPATH/crl.pem $IP_CERTS_PATH/crls**
  Instala la nueva CRL obtenida del equipo1

En este punto se puede intentar iniciar el túnel, reiniciando el servicio IPsec como se explico previamente y tratando de levantar el túnel, lo cual fracasara y dará un error del tipo '**INVALID_KEY_INFORMATION**', indicando que hubo un error en la etapa de negociación y comprobando que el router 2 esta rechazando efectivamente el certificado.

srs2-generarNuevoCertificado.sh

(ejecutado en el equipo1)

resumen:
Genera un nuevo certificado para el router 1 para reemplazar al que fue previamente revocado.

datos necesarios:
Datos acerca del dueño del primer certificado, y clave para encriptar la clave privada. También será necesario confirmar la firma de la petición de certificado, ya que se realizan ambos pasos en este script.
IMPORTANTE: Para evitar reconfigurar los archivos de configuracion y claves, se aconseja utilizar los mismos datos que se ingresaron en el paso 2 en cuanto al certificado del router local. En caso contrario, habrá que actualizar el archivo **/etc/ipsec.secrets** en el router local con el nuevo password, y el archivo **/etc/ipsec.conf** en el router remoto, en la línea de **leftid=...** con los nuevos datos del router 1.

detalles paso por paso:

- **$D/build-key-pass r1n**
  Corre el script que genera un certificado y lo firma

- **rm -f $IP_CERTS_PATH/certs/r1.pem**
  rm -f $IP_CERTS_PATH/private/r1.key
  Borra el certificado viejo y la clave privada vieja

- **cp -f $CAPATH/r1n.crt $IP_CERTS_PATH/certs/r1.pem**
  mv -f $CAPATH/r1n.key $IP_CERTS_PATH/private/r1.key
  Instala el nuevo certificado con el mismo nombre que el anterior, y procede de la misma forma con la clave privada nueva

En este punto se puede intentar iniciar el tunel, reiniciando el servicio IPsec como se explico previamente y tratando de levantar el tunel, el cual si los datos introducidos fueron correctos se iniciara de la misma forma que en la primera parte sin problemas.

5- "Debug" de ipsec

Para poder corroborar que el túnel se esta iniciando mediante certificados, se puede observar realizar una captura de protocolo, y observar que el ultimo mensaje no encriptado del protocolo ISAKMP posee una petición de envió de certificado. Adicionalmente, se puede observar los registros de los programas, para lo cual hay que ejecutar el script **debug.sh** antes de iniciar el servicio de IPsec, y luego realizar los pasos. El resultado detallado del enlace se podrá observar en el archivo **/var/log/auth.log**

INFORME A PRESENTAR

CONFIGURACION DE IPSEC CON CERTIFICADOS

1. Informar como quedan los archivos de configuración comparado con un tunel ipsec normal.
2. Realizar y comparar un trace tcpdump con rsa normal y con certificados.

CONFIGURACION DE SERVIDOR WEB APACHE SEGURO

(scripts en directorio /crypto/apache-ssl)

1. Generar un certificado nuevo
2. implantar el certificado en el servidor WEB y configurar el SSL del apache
   - copiar el certificado como /crypto/conf/apache/crt/ssl-server.crt
   - copiar la key generada como /crypto/conf/apache/crt/ssl-server.key
   - Si se crea un certificado con clave, poner la clave con la que se protege el certificado en el archivo /crypto/conf/apache/.key.sh
3. Configurar el apache
   - Ejecutar 05-configurarApache.sh
4. En el servidor router2 iniciar el servicio apache.
   /etc/init.d/apache2 force-reload
   Nota: el archivo de configuración de apache esta en /crypto/conf/apache/http.conf
5. Conectarse al mismo desde un navegador y verificar que use el certificado.
   - ¿El navegador presenta alguna advertencia? En caso afirmativo ¿a que se debe? ¿Cómo lo puede solucionar?
6. escribir un informe de configuración de apache. Detallando las configuraciones necesarias. (ver documentación de mod_ssl)

DESAFIO:

- Conseguir que el cliente web se autentique con el servidor. (El cliente debe presentar un certificado en el logon).