Usando SSH y Seguridad

Primer Taller CEDIA

Hervey Allen
(Brian Candler)

Compendio

- Donde se puede obtener SSH (Secure SHell).
- Como encender y configurar SSH.
- Donde se encuentra el cliente de SSH para Windows.
- Autenticacion del servidor a cliente (llaves host).
- Problemas de que se trate con combiando la llave del host.
- Autenticacion con contraseña del cliente a servidor.
- Autenticacion criptografica del cliente a servidor. (llaves rsa/dsa).

Metodos Criptograficos y Aplics.

En antes habiams mencionado los siguiente aplcicaciones practicas que se aplican a lo siguiente metodos:

- Al nivel de enlace encodificacion de PPP
- Al nivel de la Red IPSEC
- Al nivel de transporte TLS (SSL)
- Al nivel de aplicacion SSH, PGP/GPG

Seguridad de Nivel de la Aplicacion SSH

En esta seccion vamos a hablar sobre SSH al nivel de la aplicacion para hacer autenticacion y encodificacion de los datos.

En el discurso sobre Apache+SSL que viene vamos a hablar sobre usando SSL a nivel de transporte por conecciones seguras de Web.

Anota que hoy en dia se usa SSH version 2.

Las Preocupaciones Principales de Seguridad

SSH se lo aplica directamente cuando uno trata con dos campos de seguridad:

- La confidencialidad
  - Matiendo tus datos seguros de gente intrusa.
- Autenticacion y Autorizacion
  - Es la persona realmente quien se dice que es?

Donde Obtener SSH

Primero vea si SSH esta instalado en tu sistema y que version. La manera mas facil es:

```
ssh -V
```

- Se puede encontrar SSH en:
  - http://www.openssh.com/portable.html
  - Version 3.8 (24 de Febrero, 2004)

Se puede ver mas acerca el paquete de SSH asi:

- rpm -qa | grep ssh
- rpm -qi openssh-n.n.n-n

Inicializer y Configurar OpenSSH

En los PCs de taller ya esta instalado SSH, pero tenemos que inicializarlo.

- service sshd start

- Mira a /etc/ssh/ssh_config y /etc/sshd_config. En sshd_config fija en los defectos. Las versiones de SSH mas nuevas tienen opciones bien sensibles. Fija en:

```
PermitRootLogin yes/no
```

en antes en ssh_config existia esto. Causa problemas.

```
Protocol 1,2
```

Hay un monton de opciones en ssh_config y sshd_config. Deberia leer ambos archivos para asegurar que estan configurado como quieres. Mas encima, man ssh_config y man sshd_config.

Donde Obtener Clientes de SSH por Windows

Hay multiples versiones de clientes de SSH por Windows que son gratis, shareware, o comercial.

Vea http://www.openssh.org/windows.html por una lista.

- Putty:
  http://www.chiark.greenend.org.uk/~sgtatham/putty/
- OpenSSH por Windows (usando Cygwin):
  http://www.networksimplicity.com/openssh/
- **Secure Shell de ssh.com** (gratis por uso personal):
  http://www.ssh.com/products/ssh/download.cfm

F-Secure a http://www.f-secure.com/products/ssh/ es un buen producto si estas dispuesto a pagar.

Referencias Utiles de SSH

• Si quieres un resumen excelente de llaves de SSH RSA/DSA Daniel Robbins de gentoo.org ha escrito un serie de tres papeles que se puede encontrar en las paginas de Developer Works de IBM:

• **Los tres papeles y URLs son:**

OpenSSH Key Management, Part 1
http://www-106.ibm.com/developerworks/library/l-keyc.html
OpenSSH Key Management, Part 2
http://www-106.ibm.com/developerworks/library/l-keyc2/
OpenSSH Key Management, Part 3
http://www-106.ibm.com/developerworks/library/l-keyc3/

Mas Referencias de SSH

Para comparar SSH version 1 y 2 vea a:

http://www.snailbook.com/faq/ssh-1-vs-2.auto.html

Un libro ecelente es:

SSH, The Secure Shell
The Definitive Guide
By Daniel J. Barrett &
Richard Silverman
January 2001
ISBN: 0-596-00011-1

Metodos de coneccion de SSH

Varias cosas pueden pasar mientras que tratas de hacer una coneccion entre tu maquina (cliente) a otro maquina (servidor):

• La llave publia de servidor se pasa al cliente y el cliente se la verifica encontra known_hosts.

• Una contraseña esta usado si la llava publica esta aceptado, o ya la tiene el cliente o

• un intercambio de llaves RSA/DSA pasa y tienes que entrar to contraseña privada de tu llave privada para autenticar.

SSH Datos Utiles

Tienes una eleccion de llaves de autenticacion – RSA es el defecto y normalmente es mejor.

Los archivos importante son:

/etc/ssh/ssh_config
/etc/ssh/sshd_config
~/.ssh/identity and identity.pub
~/.ssh/id_dsa and id_dsa.pub
~/.ssh/id_rsa and id_rsa.pub
~/.ssh/known_hosts and known_hosts2
~/.ssh/authorized_keys and **authorized_keys2**
Y, anota los archivos por el host de llaves de rsa/dsa en /etc/ssh

Corre los comandos “man ssh” y “man sshd” y lea todo las descripciones por el cliente y servidor de ssh.

Intercambiando Llaves de Host

Primera vez conectando con ssh:

```
[hervey@localhost .ssh]$ ssh root@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 66:3c:ab:30:3c:be:5b:28:43:f2:e0:5c:6c:af:c0:d3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
root@localhost's password:
Last login: Tue Mar  2 22:55:33 2004 from localhost.localdomain
```

A este punto el cliente tiene en el archivo ~/.ssh/known_hosts el contenido del archivo /etc/ssh/ssh_host_rsa_key.pub del otro pc.

Proxima coneccion:

```
[hervey@localhost .ssh]$ ssh root@localhost
root@localhost's password:
Last login: Tue Mar  2 22:56:01 2004 from localhost.localdomain
```

Ahora confiado – No necesariamente una cosa buene...

Intercambiando Llaves del Host cont.

Comando	Tipo de Llava Generaod	Archivo Publico
ssh-keygen	RSA (SSH protocol 1)	identity.pub
ssh-keygen -t rsa	RSA (SSH protocol 2)	id_rsa.pub
ssh-keygen -t dsa	DSA (SSH protocol 2)	id_dsa.pub

• Tomaño por defecto de llave es 1024 bits.
• Archivos publicos son de texto.
• Archivos privados estan encifrado (encodificado) si usa una contraña (todovia texto).

Los archivos que se coresponden al intercambio de llaves del host son:

RSA/DSA (SSH Protocolo 2) ==> known_hosts (known_hosts2)
RSA (SSH Protocolo 1) ==> known_hosts

Intercambiando Llaves del Host cont.

Como se decide SSH de que archivos se va a comparar?

Mira en /etc/sshd_config. Por OpenSSH version 2 y 3 el servidor usa protocolo 2 y despues 1 por defecto.

Por defecto los clientes de OpenSSH version 2 se conecta en este orden:

RSA version 2 llave
DSA version 2 llave
Autenticacion de contraseña (A pesar si existe una llave de RSA version 1.

Presta atencion a la configuracion de "HostKeyAlgorithms" en /etc/ssh/ssh_config que se determina el orden. Se puede usar parametros en el comando de ssh para ignorar esta configuracion.

Diferencias de OpenSSH 3.x

- Nota: OpenSSH 3.8 apoya los protocolos 1.3, 1.5 y 2.0. No hay protocolo version 3.0 de SSH.
- Entre OpenSSH 3.x y 2.x el lugar de algunos archivos se cambio.
- OpenSSH 3.x se usa los archivos de authorized_keys y known_hosts por llaves de protocolo 1 y 2.

SSH - “childMagicPhrase”

Conceptos basicos para entender como una coneccion esta hecha usando SSH y una combinacion de RSA/DSA llaves:

- Cliente X contacta con server Y por puerto 22.
- Y se genera un numero aleatorio y lo encodifica usando la llave publica de X. La llave publica de X tiene que residir en Y. Puede usar scp para copiarlo a Y.
- Un numero aleatorio y encodificado esta mandad de vuelta a X.
- X se desencodifica el numero aleatorio usando su llave privado y lo mande de vuelta a Y.
- Si el numero desencodifico es igual al numero original, entonce una coneccion entre X y Y esta hecha.
- *If the decrypted number matches the original encrypted number, then a connection is made.*
- El numero originalmente encriptado y mandado desde Y a X es el “childMagicPhrase”.

SSH - ssh-agent y ssh-add

Puedes usar el ssh-agent para inicializar un proceso con un wrapper (envolvadora). Por ejemplo:

```
ssh-agent /usr/local/bin/bash
```

Entonces, puedes usar ssh-add para agregar tus llaves privadas a memoria bajo la sesion de ssh-agent. Por ejemplo:

```
ssh-add
```

Se agrega la llave privada en ~/.ssh/identity. Puedes agregar otras llaves privadas como:

```
ssh-add ~/.ssh/id_rsa
ssh-add ~/.ssh/id_dsa
```

SSH te va a pedir la contraseña de tus llaves privadas.

Lab de SSH

Ahora practicamos los siguiente conceptos:

- El uso del archivo known_nosts.
- Coneccion de SSH con autenticacion de contraseña.
- Generacion de version 2 llaves de RSA.
- Copiar de llaves publicas.
- Conectando con una contraseña privada de tus llaves con autenticacion basado en llaves.
- Usando el scp con autenticacion de llave de RSA.
- Usando ssh-agent y sss-add para conectarse sin contraseña y sin contraseña de tus llaves.

Lab de SSH cont.

El use del archivo known_hosts

Inicializa ssh: `service sshd start`

Apaga iptables: `iptables -F`

Conectarse al maquina a lado de ti usando ssh:

`ssh root@192.188.58.nn`

“nn” es la direccion IP de tu vecino.

Si esto es la primera vez conectandose a esta maquina usando SSH deberia ver (ejemplo usa localhost a localhost): -->

Lab de SSH cont.

Ejemplo continuado:

```
[hervey@localhost .ssh]$ ssh root@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 66:3c:ab:30:3c:be:5b:28:43:f2:e0:5c:6c:af:c0:d3.
Are you sure you want to continue connecting (yes/no)?
```

Sigue y contesta “yes”, pero hablamos sobre las implicaciones de esto en clase. Hay maneras de evitar esto? Puede ser un ataque de “hombre en el medio”? Que archivo esta creado o cambiado? Porque?

En el proximo slide hablaremos sobre estas temas...

Lab de SSH cont.

Coneccion de ssh con autenticacion de contraseña

Abajo cuando respondiste con “yes” fuiste preguntado para entrar la contraseña del root para localhost:

```
[hervey@localhost .ssh]$ ssh root@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 66:3c:ab:30:3c:be:5b:28:43:f2:e0:5c:6c:af:c0:d3.
Are you sure you want to continue connecting (yes/no)?  yes
```

Y, esto es que deberias haber visto:

```
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
root@localhost's password:
Last login: Tue Mar  2 22:55:33 2004 from localhost.localdomain
```

Ahora tienes una coneccion segura como root a localhost. Hablaremos sobre que paso durante este proceso.

Lab de SSH cont.

Generacion de Llaves de rsa1/rsa2/dsa

Ahora vamos a generar una sola llave de protocolo de RSA por SSH de 2048 bits. Para hacer esto, haz el siguiente comando. Si estas usando la otra maquina haz un logout primero!

Antes de continuar: tal vez tendras que editar /etc/ssh/ssh_config y asegurar que la opcion de “Protocol” esta puesto, que esta puesto a “Protocol 2,1” o “Protocol 2”

```
ssh-keygen -t rsa -b 2048
```

Tambien tiene que dar un lugar por un archivo que tendra la llave y la contraseña para encifrar el archivo de la llave. Que usa una contraseña! Archivos que tengan tus llaves privadas sin contraseña es un oyo de seguridad. Hablaremos porque es asi mientras que terminamos este ejercicio.

Lab de SSH cont.

Generacion de llave de RSA2

La salida del comando “ssh-keygen -t rsa -b 2048”:

```
[hervey@localhost .ssh]$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/hervey/.ssh/id_rsa): [enter]
Enter passphrase (empty for no passphrase): [pw]
Enter same passphrase again: [pw]
Your identification has been saved in
/home/hervey/.ssh/id_rsa.
Your public key has been saved in
/home/hervey/.ssh/id_rsa.pub.
The key fingerprint is:
f1:4f:cb:cd:c2:62:d7:ab:e7:a1:17:5e:4e:4c:e8:54
hervey@localhost.localdomain
```

Lab de SSH cont.

Copiando Llaves Publicas

Ahora que tengas una paar de llaves publicas y privadas tipo RSA2 puedes usarlas. Vamos a copiar la llave publica al mismo computador donde conectaste en antes, grabar esta llave al archivo *known_hosts*, y entonces reconectar al mismo computador y ver la diferencia:

Primero tienes que copiar los archivos de llaves publicas al mismo computador donde conectaste en antes (192.188.58.nn):

```
cd ~/.ssh
scp id_rsa.pub root@192.188.58.nn:/tmp/.
```

Tienes que entrar la contraseña por el computador y el usuario que estas usando. Continuamos con nuestro ejemplo usando tu PC conectando al PC de tu vecino como el usuario root.

Lab de SSH cont.

Copiando Llaves Publicas

La salida del comando en la pagina anterio se vea asi:

```
[hervey@localhost .ssh]$ scp id_rsa.pub root@localhost:/tmp/.
root@localhost's password:
id_rsa.pub          100% |*****************************|   410       00:00
```

Ahora tienes el archivo de la llave publica en el PC de tu vecino. Vas a necesitarlas para usar autenticacion de RSA/DSA publica/privada. El proximo paso es poner estas llaves en los archivos correctos:

Necesitas las llaves de RSA2 en *~/.ssh/authorized_keys*

Puedes tratar de hacer esto solo, o ir a la proxima pagina por los pasos para hacer esto:

Lab de SSH cont.

Copiando Llaves Publicas

Para copiar las llaves publicas a los lugares correctas haz lo siguiente:

```
ssh root@192.188.58.nn
cat /tmp/id_rsa.pub >> /root/.ssh/authorized_keys
rm /tmp/id_rsa.pub
exit
```

Si no estas seguro de que hacen estes comandos vamos a explicarlos en clase. Tambien, se puede hacer esto en varias maneras, y puedes usar los comandos diferente, tambien. Si entiendes que hacen los comandos y tengas un metodo que prefieres, entonces usalo.

Ir a la proxima pagina para conectar con tus llaves publicas/privadas!

Lab de SSH cont.

Coneccion de Llaves Publicas/Privadas

Para conectar usando su llave de protocolo 2 de RSA tipea:

```
ssh root@192.188.58.nn
```

Y, esto es que deberias ver:

```
[hervey@localhost .ssh]$ ssh root@localhost
Enter passphrase for key '/home/hervey/.ssh/id_rsa':
Last login: Tue Mar  2 23:44:13 2004 from localhost.localdom
```

Esto es realmente interesante! No entraste la contraseña del usario root en la maquina 192.188.58.nn, pero usaste el "passphrase" que eligiste por tu llave privada de protocolo 2 de RSA cuando usaste el comando "`ssh-keygen -t rsa -b 2048`" - Esto fue usado para desencodificar el numero aleatorio que fue pasado entre las maquinas (recuerdas el "childMagicPhrase?").

Porque usamos la llave de RSA2? Hablamos sobre esto en clase.

Lab de SSH cont.

Coneccion de Llaves Publicas/Privadas

Primero desconecta de tu sesion de SSH que hiciste en antes:

```
exit
```

Ahora trata de copiar un archivo desde tu maquina a la otra maquina (elige un archivo chico) usando SCP (SeCure coPy):

```
scp filename root@192.188.58.nn:/tmp/.
```

Que notaste? Deberias haber notado que ahora no tienes que entrar la contreseña por la otra maquina, pero tienes que entra tu "passphrase" de tu llave privada de protocolo 2 de RSA.

Esto es normal. SCP y SSH vienen del mismo paquete de software (OpenSSH) y ambos usan llaves de RSA y DSA en la misma manera.

Lab de SSH cont.

Ejemplo de una coneccion de "No Challenge"

Ahora usamos los programas `ssh-agent` y `ssh-add` para hacer un ambiente en tu maquina que te permite conectar a la maquina de tu vecino como root sin tener que entrar una contraseña ni un "passphrase".

Pero, tendras que entrar tu "passphrase" para tu llave privada de RSA protocolo 2 una vez durante la sesion. Hablaremos sobre `ssh-add` y `ssh-agent` en clase, pero lea "`man ssh-agent`" y "`man ssh-add`" para mas informacion:

En la proxima pagina vas a hacer tu ambiente de tu shell bash para contener el "passphrase" privado de tu llave RSA2. Esto te permitira conectar, hacer logout, reconectar, salir, y conectar de nuevo todo las veces que quieres a la maquina de tu vecino solo usando tu "passphrase" privado de RSA2 *una vez*:

Lab de SSH cont.

Ejemplo de una coneccion de "No Challenge"

Seguir estes pasos para hacer una coneccion de "no challenge":

```
ssh-agent /bin/bash
ssh-add
ssh root@192.188.58.nn
```

Que paso? Solo deberias haber tenido que entrar tu "passphrase" de tu llave privada de RSA2 (recuerda, esto es que esta en ~/.ssh/id_rsa) cuando tipeaste `ssh-add`. Y, entonces cuando conectaste no fue necesario entrar ningun contraseña ni "passphrase". (Si, por ser caso, tienes una llava privada de RSA1, vas a tener que entrar el "passphrase" por ~/.ssh/identity).

Ahora por el parte mas entretenido. Salir tu sesion y conectar de nuevo a la misma maquina (por ejemplo, el PC de tu vecino):

```
logout
ssh root@192.188.58.nn
```

Ahora que paso?

Lab de SSH cont.

Notas de Coneccion "No Challenge"

- ssh-add y ssh-agent actuan un poco diferente que solo usar ssh.
- Si no especifas un "passphrase" por los archivos de tus llaves privadas, entonces cuando conectas a otra maquina que tiene tu llave publica, es realmente posible conectar sin usar ningun contraseña. (ssh version 2 requiere un cambio en el archive /etc/ssh/sshd_config para permitir esto). Ojo – Es realmente peligroso no usar ningun contraseña por tus archivos de llaves priavadas.
- Anota que el defecto de ssh-add es mirar al archivo ~/.ssh/indentiy primero.

Lab de SSH cont.

Mas Datos Extras

- Puedes usar el ssh-agent a “envolver” (wrap) otros programs que, tal vez, requieren autenticacion de RSA/DSA pero que no pueden usar multiple “passphrases” ni contraseñas.
- Las ultimas paginas tienen una sesion entera (incluyendo comentarios) de usar los programas de ssh-agent y ssh-add.

Lab de SSH cont.

sesion de ssh-agent/ssh-add

```
[hervey@localhost .ssh]$ which bash                 [Donde esta bash]
/bin/bash
[hervey@localhost .ssh]$ ssh-agent /bin/bash        [Envuelve bash con ssh-agent]
[hervey@localhost .ssh]$ ssh-add  [Agrega, por defecto, llaves privads de rsa/dsa]
Enter passphrase for /home/hervey/.ssh/id_rsa:
Identity added: /home/hervey/.ssh/id_rsa (/home/hervey/.ssh/id_rsa)
Identity added: /home/hervey/.ssh/id_dsa (/home/hervey/.ssh/id_dsa)
[hervey@localhost .ssh]$ ssh-add ~/.ssh/id_rsa [Especificamente agrega llave rsa2]
Enter passphrase for /home/hervey/.ssh/id_rsa:
Identity added: /home/hervey/.ssh/id_rsa (/home/hervey/.ssh/id_rsa)
[hervey@localhost .ssh]$ ssh root@localhost
Last login: Tue Mar  2 23:45:33 2004 from localhost.localdomain
[root@localhost root]#                          [Conecta sin ningun contraseña]
```

Anota: Yo habia corrido “ssh-keygen -t dsa 1024”, asi tengo una llave privada de DSA, tambien.

Lab de SSH cont.

sesion de ssh-agent/ssh-add

```
[root@localhost root]#                          [Salir de sesion de shell]
Connection to localhost closed.
[hervey@localhost .ssh]$ ssh root@localhost
Last login: Tue Mar  2 23:51:28 2004 from localhost.localdomain
[root@localhost root]#          [Conecta de nuevo - no contraseña necesaria!]

[root@localhost root]#                          [Salir de sesion de shell]
Connection to localhost closed.


host6# exit                                     [Selir de sesion de nuevo]
logout
[hervey@localhost .ssh]$
[hervey@localhost .ssh]$ ssh-add -l       [Muestra las firmas de las lavesrsa/dsa]
2048 f1:4f:cb:cd:c2:62:d7:ab:e7:a1:17:5e:4e:4c:e8:54 /home/hervey/.ssh/id_rsa (RSA)
2048 a5:50:c0:b1:94:cc:fa:fa:d8:f9:d5:6a:51:f1:75:f0 /home/hervey/.ssh/id_dsa (DSA)
```

Lab de SSH cont.

terminacion de sesion de ssh-agent/ssh-add

```
[hervey@localhost .ssh]$ ssh-add -d ~/.ssh/id_rsa   [Remover una llave privada]
Identity removed: /home/hervey/.ssh/id_rsa (/home/hervey/.ssh/id_rsa.pub)
[hervey@localhost .ssh]$ ssh-add -l                 [Mostra llaves que quedan]
2048 a5:50:c0:b1:94:cc:fa:fa:d8:f9:d5:6a:51:f1:75:f0 /home/hervey/.ssh/id_dsa
(DSA)
[hervey@localhost .ssh]$
[hervey@localhost .ssh]$ exit              [Salir el shell ssh-agent de bash]
exit
[hervey@localhost .ssh]$
```

No olvides leer sobre todo esto con “`man ssh-agent`,” y “`man ssh-add`”. Hay mucho mas opciones y detalles de como funcionan estes programas.

Haciendo Tuneles con SSH

La tema de que has esperado...

- Puedes usar SSH para hacer servicios que no son seguros mas seguros usando los tuneles de SSH.
- Haciendo tuneles de servicios con SSH incluye autenticacion entre known_hosts, el uso de contraseñas, y intercambios de llaves privads y publicas.
- Usando SSH se puede hacer tuneles en forma indirecta usando una maquina entremedio.

Haciendo Tuneles con SSH cont.

El concepto basico se vea asi:

- Conectars de una maquina a otra maquina como `username`.
- Usa opciones de SSH para especificar en que puerto de TCP de la maquina remota que quieres mandar datos al puerto en tu maquina local.
- Su coneccion de SSH va a hacer un tunel para que los datos pueden viajar en forma segura entre la maquina remota y tu maquina.
- Hay varias opciones de que tienes que estar consciente:

Haciendo Tuneles con SSH cont.

Haciendo Tuneles – Un Ejemplo

Aqui es un ejemplo de hacer un tunel usando SSH bajo Linux:

```
ssh -C -f username@host.domain -L 1100:localhost:110 sleep 10000
```

Que esta pasando?

- La opcion '-C' especifica comprimir los datos. Bueno si estas usando un modem, malo si estas usando en red rapida.
- '-f' significa que el proceso duerma antes de ejecutar el comando listado (en este caso “sleep 10000”).
- '-L' manda el puerto por la izquierda, o el cliente (1100), al puerto por la derecha (110) o el lado remoto.

Haciendo Tuneles con SSH cont.

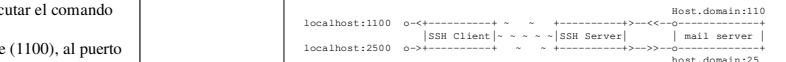
Haciendo Tuneles – Un Ejemplo cont.

Entonces, que hace este comando?

```
ssh -C -f username@host.domain -L 1100:localhost:110 sleep 10000
```

- Se hace un “tunel” de tu email por POP desde el puerto 110 en el lado remoto por el puerto 1100 en el lado local (tu maquina).
- Este proceso queda “dormida” (en el “background”) por 10000 segundos (todovia esta corriendo).
- Todo esto esta hecho bajo la autoridad entre Ud. (cliente) y el usuario@host.domain.

Diagrama* de un tunel de los servicios smtp y POP



*Thanks to http://www.ccs.neu.edu/groups/systems/howto/howto-sshtunnel.html

Haciendo Tuneles con SSH cont.

Haciendo Tuneles – Un Ejemplo cont.

Porque usar puertos como “1100” y “2500”?

- Puertos hasta 1024 solo estan bajo el control del usuario root.
- Si eres root puedes hacer un forward de 110 a 110, 25 a 25, y etc.
- Otros trucos de hacer tuneles con SSH incluyen haciendo tuneles por XWindows, IMAP, etc.
- Por el lado del cliente tienes que puntar tus programas a “localhost” - Por ejemplo, por POP/SMTP, su cliente de correo tiene que usar “localhost” en vez de host.domain (ej. “mail.host.com”).
- Si no eres root, y los puertos cambian, entonces tu cliente de correo tienes que poder de cambiar los puertos de SMTP y POP.
- **Ahora mostramos esto usando el cliente de correo Thunderbird de Mozilla bajo Linux ahora...**

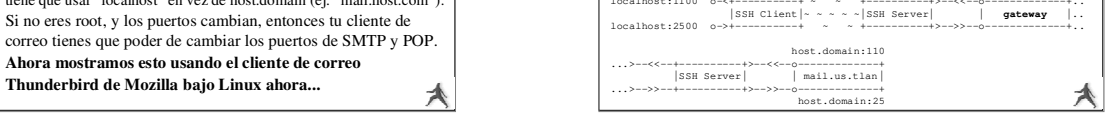
Haciendo Tuneles con SSH cont.

Un Ejemplo mas de Hacer Tuneles

Peudes usar SSH para hacer tuneles indirecto, o “Indirect Port Forwarding”

- Que harcer si el email de tu organizacion esta detras un firewall?
- Conectarse via una maquina intermediana (un puerto o gateway).
  Un ejemplo verdadero:

```
Ssh -C -f hallen@gateway.turbolinux.com -L
2500:mail.us.tlan:25 -L 1100:mail.us.tlan:110 /bin/sleep
10000
```



Haciendo Tuneles con SSH Conclusion

- Hacer tuneles te permite aceder servicios basicos como POP y IMAP en forma segura.
- Puede hacer un tunel de puertos de TCP usando SSH.
- Puede usar /etc/services para verificar que no estas usando un puerto ya definido.
- Solo root puede redefinir puertos abajo 1024.
- Puede hacer un tunel entre puertos directamente entre dos maquinas y en forma indirecta usando una maquina en el entremedio.

Conclusion SSH

SSH y SCP son dos herramientos excelentes para conectarse entre maquinas y para copiar datos en una forma segura.

Si puedes, recomendamos remover Telnet y FTP de tu sistema. O, solo permitir acceso de FTP usando el usuario “anonymous”

Puedes usar SSH para hacer tuneles entre puertos de TCP que normalmente no son seguros y que pasen tu informacion (usuarios, contraseñas, y datos de sesion) con datos no encifrado.

Recuerdas – Usar referencias por informacion mas detallada. Este incluye “`man ssh`” y “`man sshd`” por ejemplo.