CODDNS ANEXO

ANEXO DE SEGURIDAD PARA SERVIDORES

Proyecto Final ASIR
Fco de Borja Sánchez Soto
IES Tierno Galván (Parla)
Junio de 2015

PROTEGIENDO EL SERVIDOR

Introducción

En la guía oficial de CODDNS establecemos unas políticas de seguridad inexistentes.

Este anexo cubre esa carencia, ofreciendo un script basado en IPTABLES para proteger nuestro servidor.

También se indicarán los pasos a seguir para establecer una conexión SSH mediante el uso de pares de claves pública/ privada contra nuestro servidor.

IPTABLES como servicio

Crearemos un script teniendo en cuenta lo siguiente:

- Nuestro servidor debe resolver consultas DNS del exterior
    - Abriremos el puerto UDP 53 y UDP 953 como puerto de destino en entrada y origen en salida.
- Nuestro servidor debe poder resolver nombres para, por ejemplo, instalar paquetes con el gestor de aplicaciones.
    - Abriremos el puerto UDP 53 como puerto de destino en salida y origen en entrada
- Debemos proveer la interfaz web a los usuarios
    - Abriremos ambos puertos 443 y 80 como destino en entrada y como origen en salida.
- Debemos poder enviar correos electrónicos
    - Abriremos 25 y 587 como destino en salida y origen en entrada

Dejamos de manera opcional la posibilidad de configurar el servidor también como servidor de correo.

Para la funcionalidad actual aplicamos el script de protección iptables adjunto en la siguiente página.

```
#!/bin/sh
#
#  Script basado en IPTABLES para la proteccion de
# servidores con sistemas CODDNS
#
# Autor: Fco de Borja Sanchez
# http://coddns.org
#-------------------------------------------------
IPT="iptables"


function start() {
    echo "Protegiendo el sistema..."
    $IPT -P INPUT DROP
    $IPT -P OUTPUT DROP
    $IPT -P FORWARD DROP

    $IPT -A INPUT -i lo -j ACCEPT
    $IPT -A OUTPUT -o lo -j ACCEPT

    $IPT -A INPUT -m state --state ESTABLISHED,RELATED
    $IPT -A OUTPUT -m state --state ESTABLISHED,RELATED

    echo "  - Abriendo SSH"
    $IPT -A INPUT  -p TCP --dport 22 --sport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p TCP --sport 22 --dport 1024:65535 -j ACCEPT

    echo "  - Abriendo HTTP"
    $IPT -A INPUT  -p TCP --dport 80 --sport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p TCP --sport 80 --dport 1024:65535 -j ACCEPT
    # abrimos los dos sentidos -> instalacion de paquetes
    $IPT -A INPUT  -p TCP --sport 80 --dport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p TCP --dport 80 --sport 1024:65535 -j ACCEPT

    echo "  - Abriendo HTTPs"
    $IPT -A INPUT  -p TCP --dport 443 --sport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p TCP --sport 443 --dport 1024:65535 -j ACCEPT

    echo "  - Abriendo DNS"
    $IPT -A INPUT  -p UDP --dport 53 --sport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p UDP --sport 53 --dport 1024:65535 -j ACCEPT
    $IPT -A INPUT  -p UDP --dport 953 --sport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p UDP --sport 953 --dport 1024:65535 -j ACCEPT
    # abrimos los dos sentidos
    $IPT -A INPUT  -p UDP --sport 53 --dport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p UDP --dport 53 --sport 1024:65535 -j ACCEPT

    echo "  - Abriendo salidas para el correo"
#    echo "     - POP3"
#    $IPT -A INPUT  -p TCP --dport 110 --sport 1024:65535 -j ACCEPT
#    $IPT -A OUTPUT -p TCP --sport 110 --dport 1024:65535 -j ACCEPT
#    echo "     - POP3s"
#    $IPT -A INPUT  -p TCP --dport 995 --sport 1024:65535 -j ACCEPT
#    $IPT -A OUTPUT -p TCP --sport 995 --dport 1024:65535 -j ACCEPT
#    echo "     - IMAP"
#    $IPT -A INPUT  -p TCP --dport 143 --sport 1024:65535 -j ACCEPT
#    $IPT -A OUTPUT -p TCP --sport 143 --dport 1024:65535 -j ACCEPT
#    echo "     - SMTP entrante"
#    $IPT -A INPUT  -p TCP --dport 25 --sport 1024:65535 -j ACCEPT
#    $IPT -A OUTPUT -p TCP --sport 25 --dport 1024:65535 -j ACCEPT
    echo "     - SMTP saliente"
    $IPT -A INPUT  -p TCP --sport 25 --dport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p TCP --dport 25 --sport 1024:65535 -j ACCEPT
#    echo "     - SMTPs entrante"
#    $IPT -A INPUT  -p TCP --dport 587 --sport 1024:65535 -j ACCEPT
#    $IPT -A OUTPUT -p TCP --sport 587 --dport 1024:65535 -j ACCEPT
    echo "     - SMTPs saliente"
    $IPT -A INPUT  -p TCP --sport 587 --dport 1024:65535 -j ACCEPT
    $IPT -A OUTPUT -p TCP --dport 587 --sport 1024:65535 -j ACCEPT


    echo " HECHO!"
}
```

```
function stop() {
    echo "Desprotegiendo el sistema..."
    $IPT -P INPUT ACCEPT
    $IPT -P OUTPUT ACCEPT
    $IPT -P FORWARD ACCEPT

    $IPT -F INPUT
    $IPT -F OUTPUT
    $IPT -F FORWARD
    $IPT -t mangle -F
    $IPT -t nat -F
    echo " HECHO"
}

case $1 in

start)
    # clear possible previous rules
    stop
    # apply new ones
    start
;;

stop)
    # clear rules
    stop
;;

restart)
    # clear possible previous rules
    stop
    # apply new ones
    start
;;

status)
    # show status
    iptables -L
;;
*)
    echo "Uso: $0 start|stop|restart"
;;
esac
```

Accediendo a nuestro servidor con par clave pública/ privada

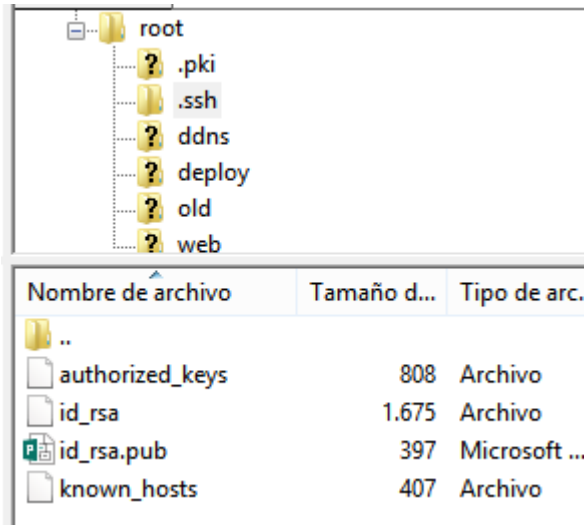
ACCEDIENDO DESDE WINDOWS (PUTTY)

Empezamos creando una clave en el servidor de CODDNS y copiándola contra el mismo servidor:

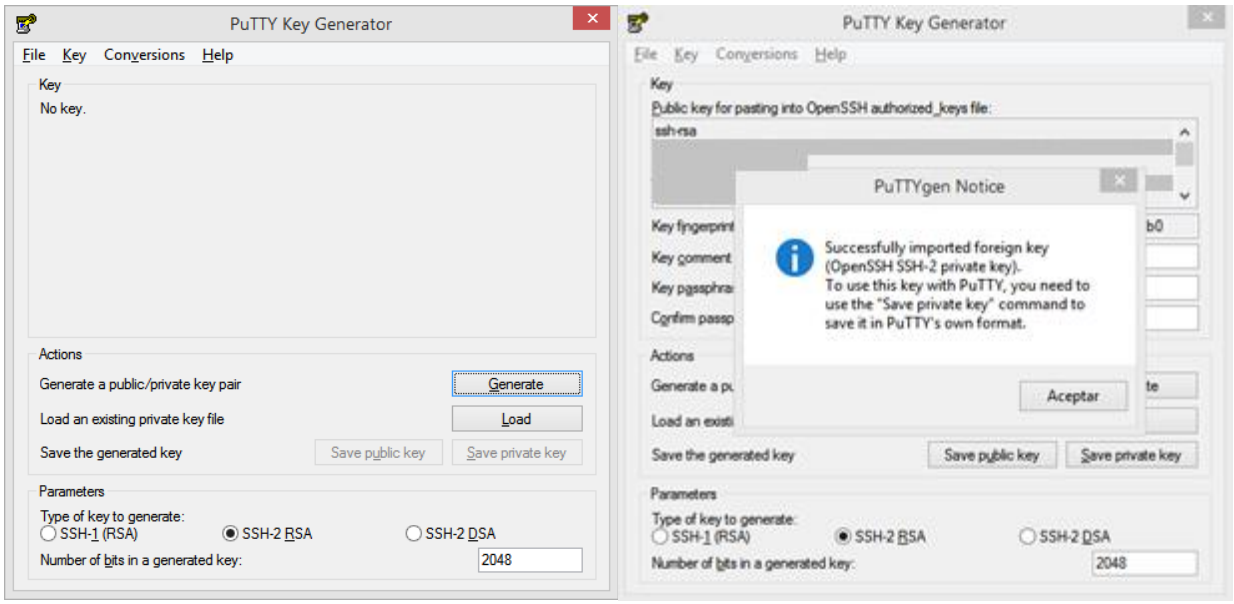
```
ssh-keygen
ssh-copy-id root@coddns.org
```

Se almacenará en el directorio ~/.ssh/id_rsa

Descargamos el par de claves vía Filezilla, por ejemplo a nuestro equipo Windows.



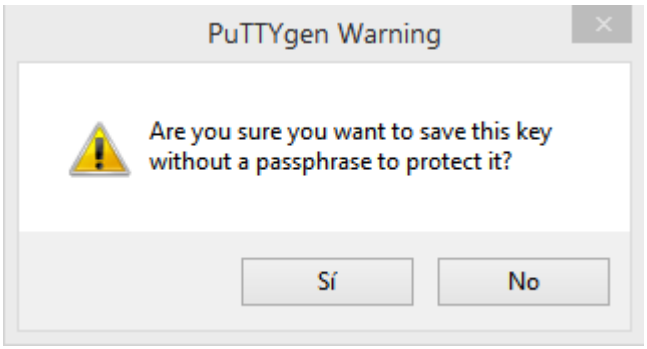
Utilizaremos puttygen para generar el fichero .ppk que nos autorice un acceso directo a nuestro servidor, cargando el fichero id_rsa previamente descargado (clave privada):



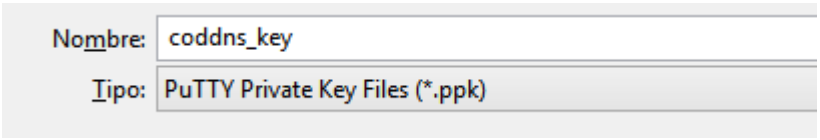
El mensaje nos indica que la carga de la clave privada ha sido un éxito.

Seleccionaremos “Save private key” para guardar nuestra clave en un formato que putty entienda:

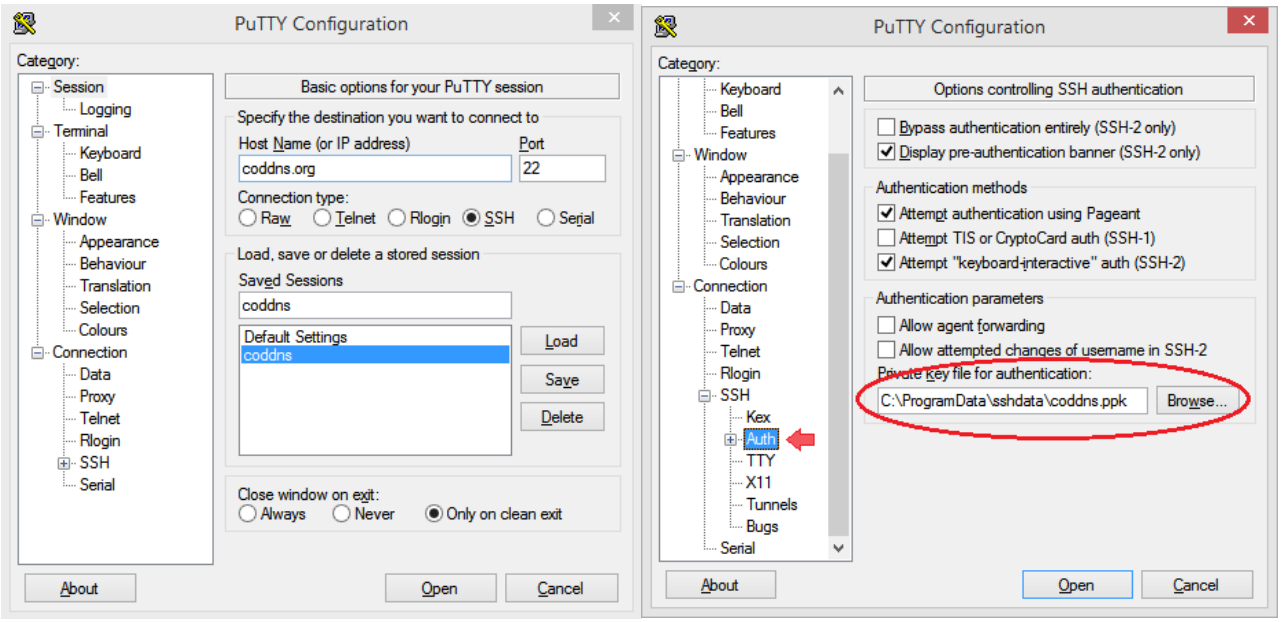
Nos avisará de lo siguiente:



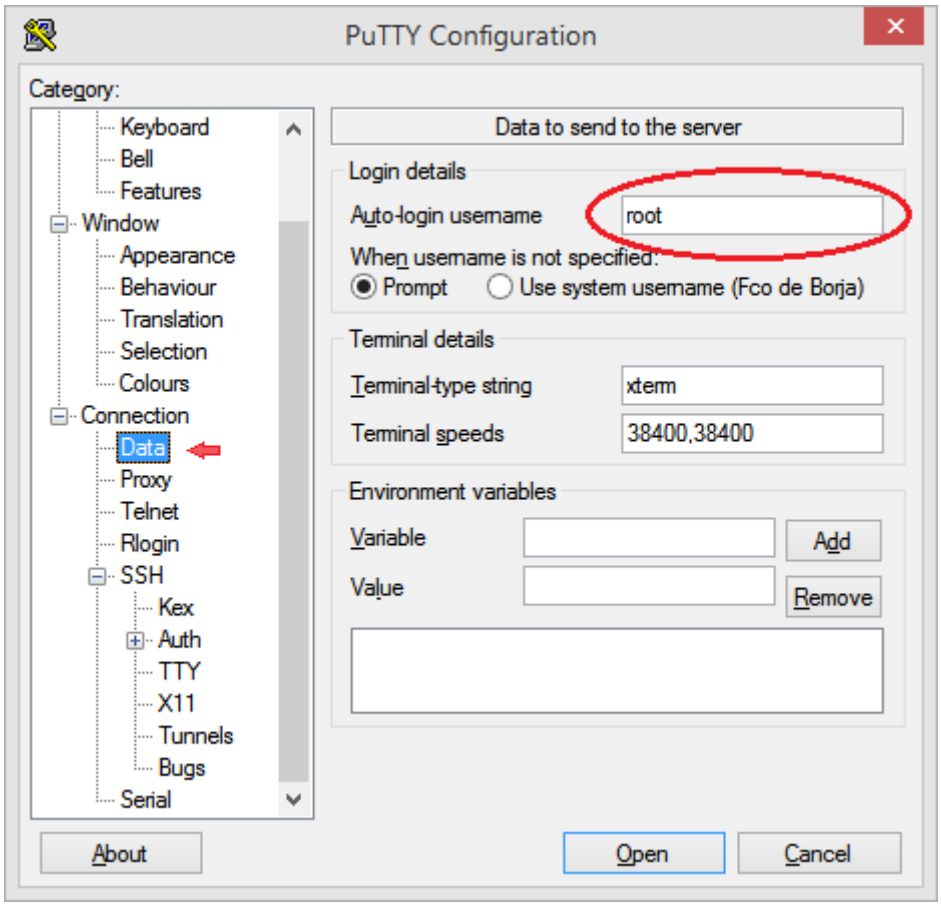
Responderemos que sí, dado que la gracia de conectar usando el par de claves es no tener que poner una contraseña, aunque seremos vulnerables si alguien sin permiso accede a nuestros ficheros.



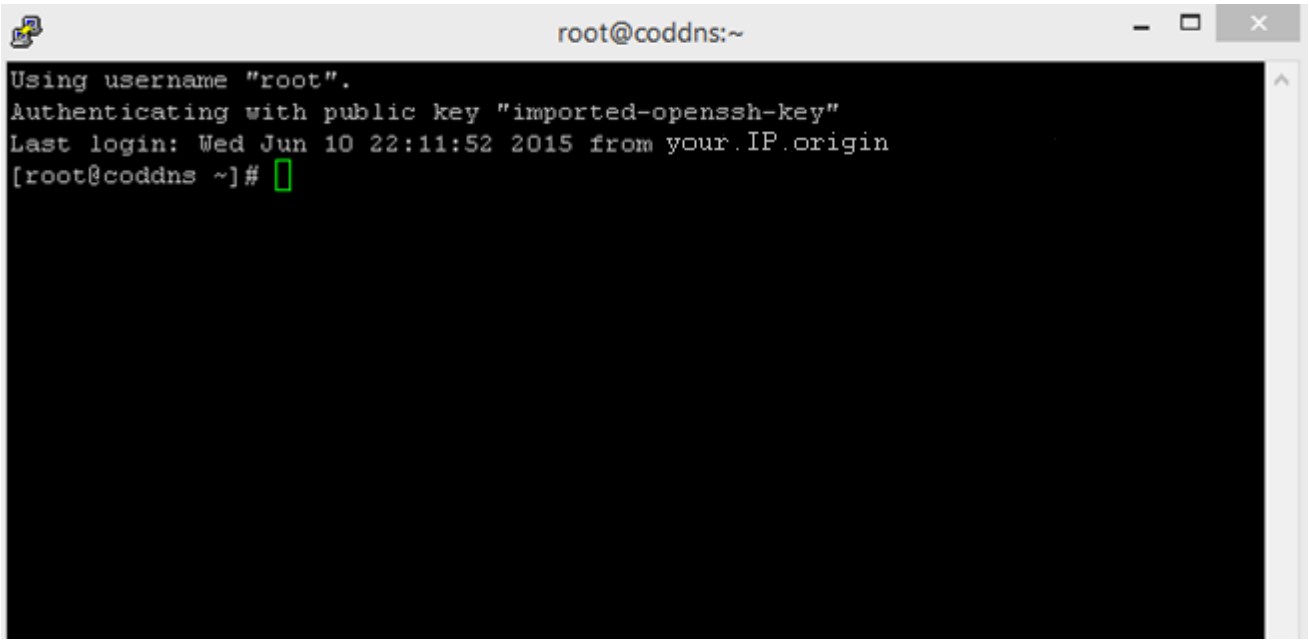
Una vez tenemos nuestro fichero .ppk, iremos a putty y configuraremos los datos necesarios:



Podemos agregar también la opción de autologin como root indicándolo en el lugar preciso:



¡Y listo! Ya tenemos acceso directo a nuestro servidor:



ACCEDIENDO DESDE LINUX

Comparado con la forma de hacer las cosas en Windows, esta es realmente un paseo.

Simplemente seguimos los pasos que se describen a continuación:

```
# crear una clave
ssh-keygen
# copiar dicha clave al servidor
ssh-copy-id root@coddns.org
# introducir la contraseña una única vez, ya tenemos el cliente
#  autorizado en el servidor

ssh root@coddns.org
```

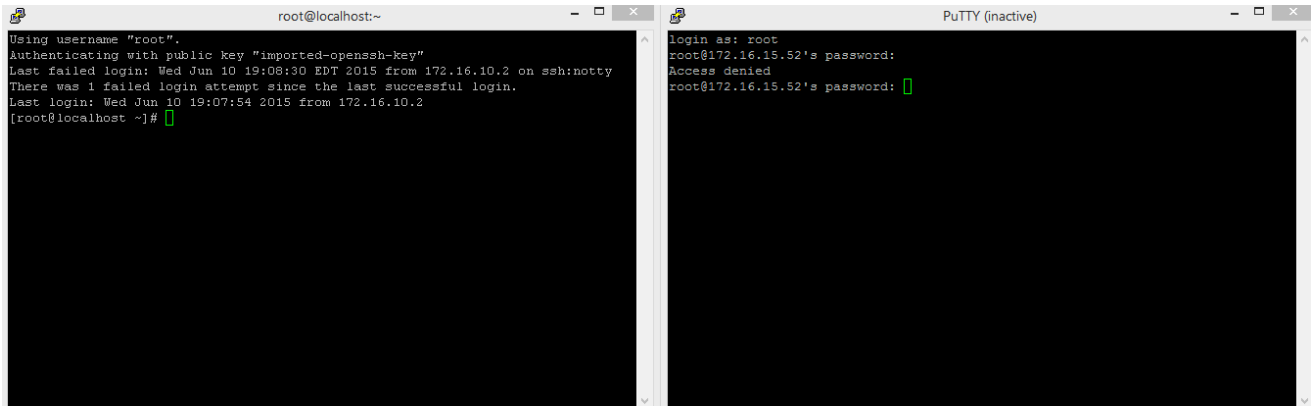
Completado.

Si queremos dotar de mayor seguridad a nuestro servidor, podemos deshabilitar la opción de login por uso de contraseña en las opciones de sshd_config:

Editamos el fichero /etc/ssh/sshd_config, modificando la línea PermitRootLogin de YES a without-password

```
PermitRootLogin without-password
```

Una vez hecho esto (primero en un entorno virtual de prueba jeje) comprobamos que funciona correctamente



A la izquierda tenemos un acceso con clave compartida, y a la derecha un acceso denegado por intento de conexión mediante introducción de contraseña.