Universidad Nacional Autónoma de México

DGTIC

PBSC 8

Proyecto Módulo 3

Túnel ICMP

Octavio Domínguez Salgado

Rodrigo Augusto Ortiz Ramón

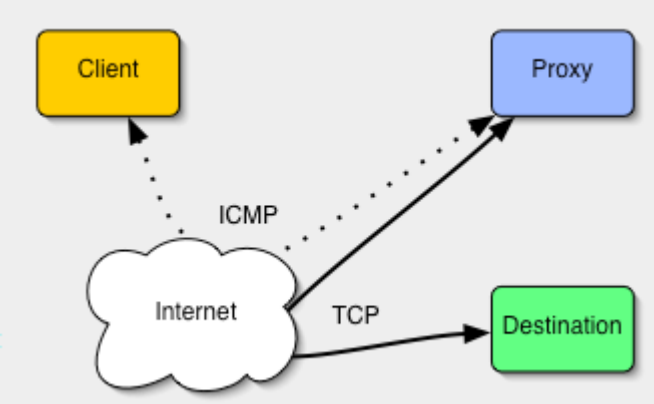
Diego Valverde Rodríguez

14/05/2014

Introducción

Ptunnel es una aplicación que permite conexiones TCP por medio de un túnel a un host remoto usando el protocolo ICMP de solicitud de eco y de los paquetes de respuesta, comúnmente conocidas como las solicitudes de ping y respuestas. A primera vista, esto puede parecer una cosa bastante inútil, pero en realidad puede ser muy útil en algunos casos.

El ptunnel puede ser usado para evitar la detección del contenido que el paquete de icmp por parte de un sistema detector de intrusos, cabe mencionar que si el contenido (payload) es muy grande, el IDS podría sospechar que se trata de un paquete malicioso, por tanto es importante cuidar la división de un archivo o mensaje para que vaya en partes iguales y con cierto tamaño para que el detector de intrusos no sospeche.



Protocolos

ICMP (Protocolo de mensajes de control de Internet) es un protocolo que permite administrar información relacionada con errores de los equipos en red. Si se tienen en cuenta los escasos controles que lleva a cabo el protocolo IP, ICMP no permite corregir los errores sino que los notifica a los protocolos de capas cercanas.

Medios usados

Los medios utilizados para el desarrollo de los programas se usó el framework de Python versión 2.7, el módulo de pycripto versión 2.6.1, 2 sistemas operativos

Windows 7 virtualizados en VMware, para programar se usó netbeans con el plugin de Python, geany y para controlar las versiones del desarrollo se usó git hub.

Algoritmos de cifrado

Se cifró y descifró mediante el algoritmo de AES en Python, ya que su implementación es muy simple y existe mucha documentación, solo hay que instalar un biblioteca llamada pycripto que proporciona un marco agradable para la creación de prototipos y experimentación con algoritmos criptográficos, gracias a los algoritmos de clave pública que se implementan fácilmente

Desarrollo

Primero se buscó la información necesaria para entender cómo resolver la problemática de creación del túnel icmp, una de las primeras opciones para programar el túnel fue usar Python, ciertamente buscando más acerca del tema, se optó por usar ese lenguaje ya que permitía crea los tuneles que servirían para enviar los paquetes.

Se lograron encontrar cachos de código que permitieron unirse para construir la aplicación requerida.

Primero se encontró un código que permitía crear los paquetes icmp con sus diferentes campos, realizar el encapsulado y el envió del paquete, a continuación se muestra parte del código encontrado.

Conclusiones

Justificación de las acciones tomadas

Se escogió Python como lenguaje de programación por su gran flexibilidad, su documentación bien explicada y que tiene la opción de poder crear tunneles por

medio de icmp, en cuanto a versión de Python que se escogió (2.7) es porque es más compatibles con la aplicación del tunnel y con el módulo de pycrypto

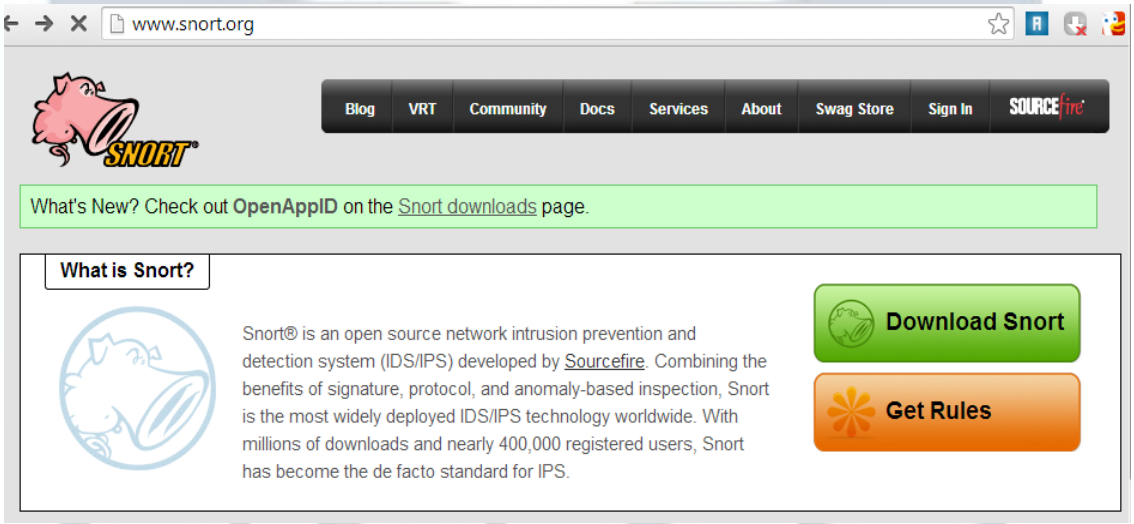
Manual de usuario

Manual técnico

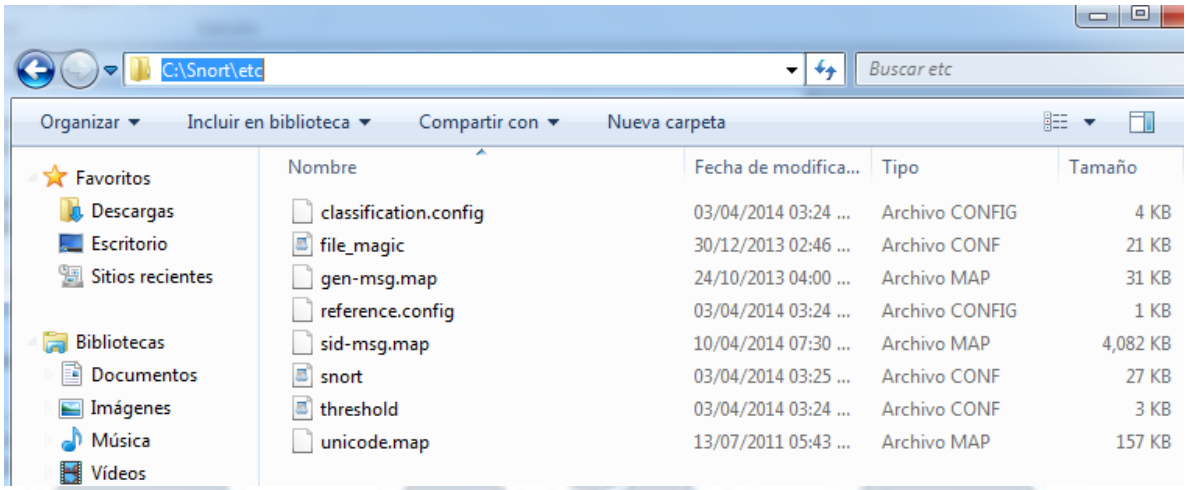
Anexo 1

IDS

Se descarga Snort y las correspondientes reglas para conformar nuestro IDS.



Después de la instalación se edita el archivo de configuración : snort.conf




```
##################################################
# Step #1: Set the network variables.  For more information,
README.variables
##################################################

# Setup the network addresses you are protecting
#ipvar HOME_NET any
var HOME_NET 192.168.150.0/24
```

Se especifica la variable que contendrá la red a la que se aplicará el IDS

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an
absolute path,
# such as:  c:\Snort\rules
var RULE_PATH c:\Snort\rules
var SO_RULE_PATH c:\Snort\so_rules
var PREPROC_RULE_PATH c:\Snort\preproc_rules
```

Se especifican las rutas que contienen los rules del IDS dentro de nuestro sistema de archivos.

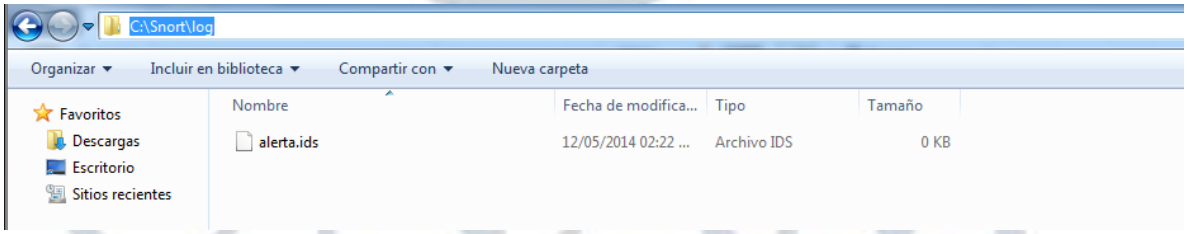
```
# Step #4: Configure dynamic loaded libraries.
# For more information, see Snort Manual, Configuring Snort -
Dynamic Modules
######################################################

# path to dynamic preprocessor libraries
dynamicpreprocessor directory C:\Snort\lib
\snort_dynamicpreprocessor

# path to base preprocessor engine
dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll

# path to dynamic rules libraries
#dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

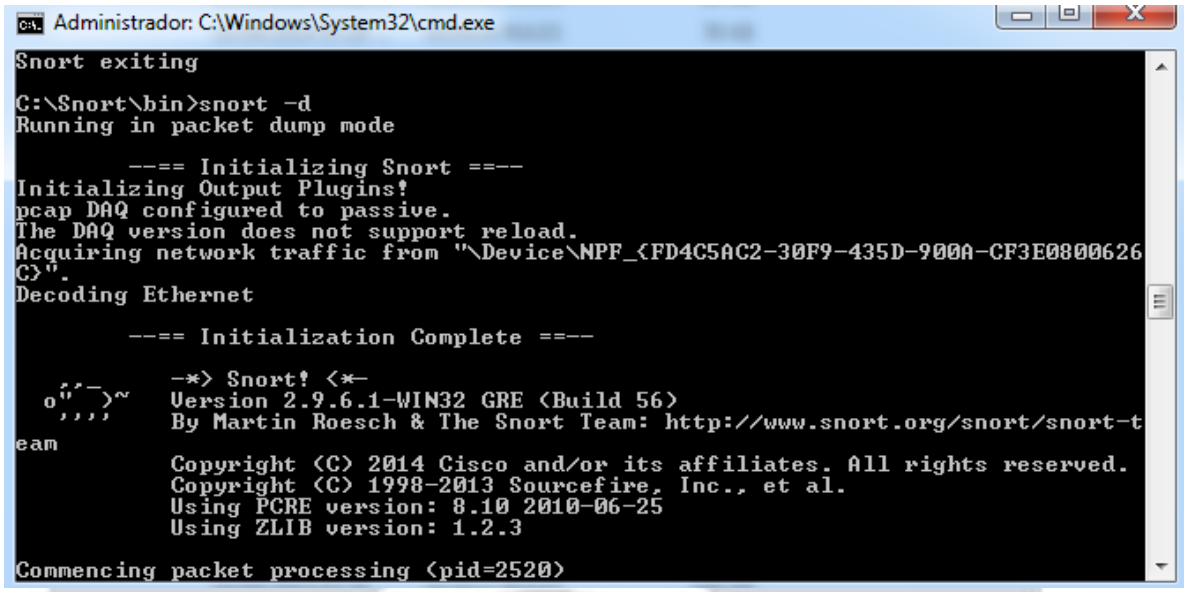
Se configuran las bibliotecas dinámicas

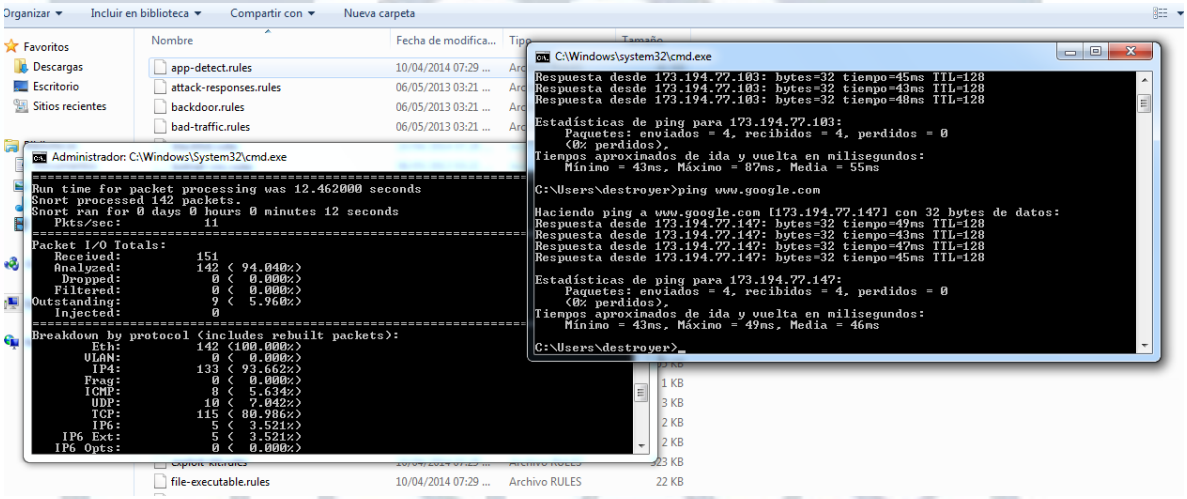
```
# metadata reference data.  do not modify these lines
include C:\Snort\etc\classification.config
include C:\Snort\etc\reference.config
```

```
# pcap
# output log_tcpdump: tcpdump.log
output alert_fast: alerta.ids
```

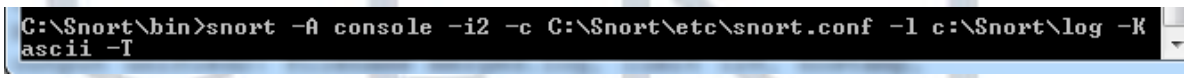
Se crea archivo IDS en el log de Snort







Se ingresa la siguiente línea desde el CMD para comprobar la correcta configuración del IDS.



De aparecer errores se corrigen en el archivo de configuración: sort.conf. Sólo comentando las líneas que se identifican como los causantes del programa en el mensaje que aparece después de ejecutar el comando anterior.

Referencias

https://pypi.python.org/pypi/pycrypto/2.6.1

http://stuff.mit.edu/afs/sipb/user/golem/tmp/ptunnel-0.61.orig/web/

http://es.kioskea.net/contents/265-el-protocolo-icmp

http://pastebin.com/JLD6grb2

http://pastebin.com/gH3zzHdQ

http://www.snort.org/snort-downloads?