Navegar gratis por una WiFi de pago mediante un túnel DNS con dns2tcp

Publicado por Dani Martinez on miércoles, 31 de agosto de 2016 Etiquetas: herramientas , linux , técnicas , tutoriales , wi-fi

Normalmente en una red WiFi de pago, cuando no estamos logeados en un portal cautivo con un usuario válido, tenemos el tráfico restringido.

Aún así y en la mayoría de los casos, podemos hacer un túnel y navegar a través del protocolo DNS debido a que el servidor DNS que automáticamente nos asigna el DHCP puede hacer consultas hacia Internet.

Podemos comprobar si podemos aprovechar esta característica en nuestra WiFi:

```
nslookup www.google.com
```

Si recibimos respuesta, podemos aprovechar esta vulnerabilidad y encapsular nuestro tráfico en DNS. Para llevar a cabo esta tarea, vamos a usar `dns2tcp`.

Prerrequisitos:

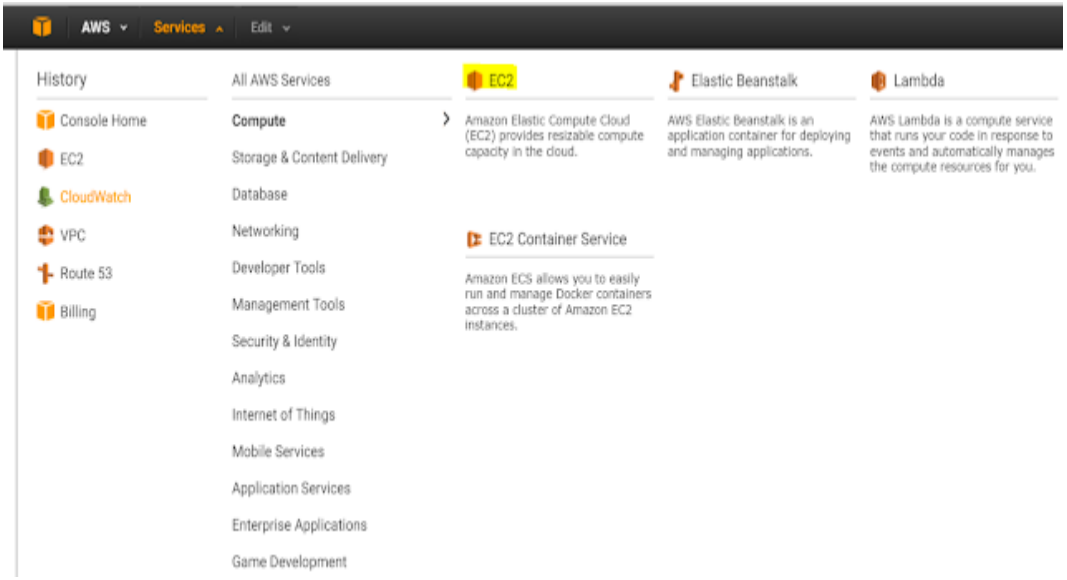
- Una máquina corriendo el servidor dns2tcp
- Un nombre de dominio.

Servidor dns2tcp

Podemos tener corriendo el servicio en local simplemente en nuestro ordenador, añadiendo una regla en el NAT para el puerto 53, pero voy a explicar como hacerlo en una máquina de Amazon AWS, que además es gratis durante un año.

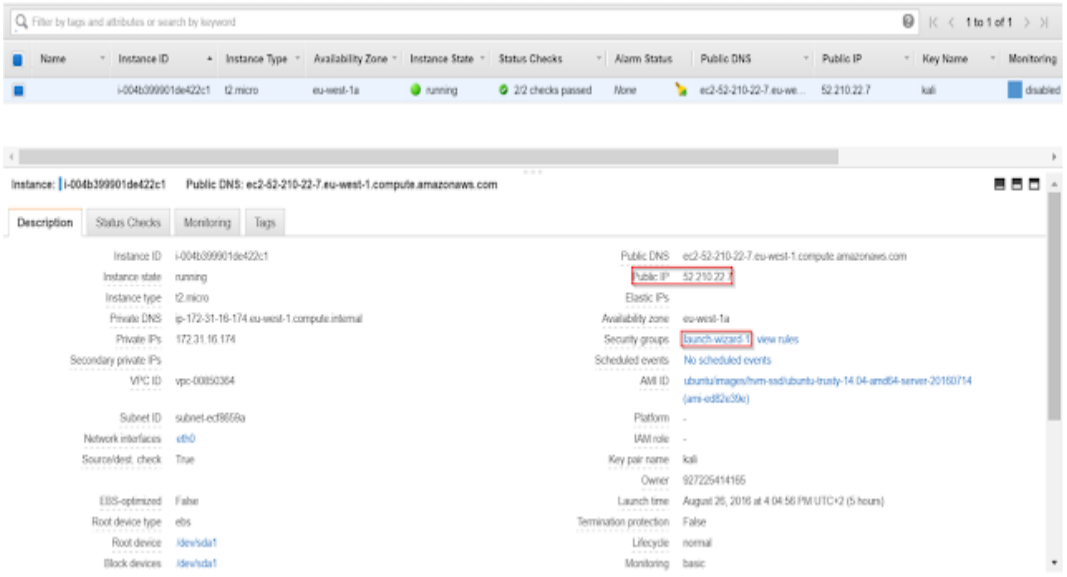
Vamos a: https://aws.amazon.com/ y buscamos el plan gratuito de hosting de una instancia EC2 durante un año.

Una vez nos logeamos, vamos a Services > Cloud Computing y elegimos EC2.

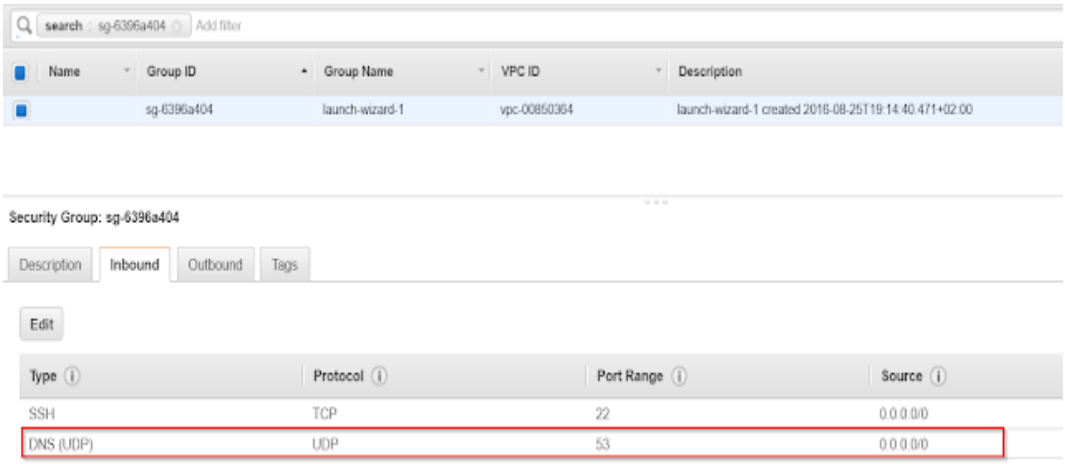


Después vamos al panel 'Instances' y damos a 'launch instance'. Yo he seleccionado Ubuntu Server de tipo t2.tiny. La máquina no es muy potente pero es gratis 750 horas al mes durante un año.

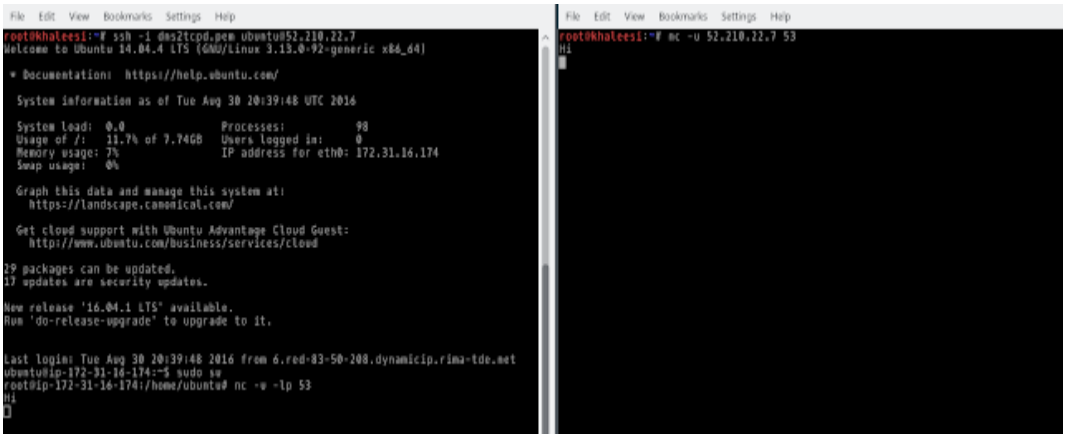
Cuando creamos la instancia nos dan una clave para conectarnos por ssh con la opcion -i.
Ahora que tenemos la instancia, podemos ver la información relevante:



Vemos nuestra ip pública y **Security Groups,** que es donde determinamos nuestras reglas de entrada y salida, ya que nuestra máquina tiene una ip privada.
Abrimos el puerto 53, dentro del Security Group que tenemos asignado a nuestra instancia:



Nos conectamos por ssh y comprobamos que tenemos el puerto 53 abierto al exterior:



Ahora descargamos el paquete dns2tcp:

```
$ apt-get install dns2tcp
```

Antes de explicar la configuración del servidor dns2tcp, vamos a ver la configuración del dominio, ya que así va a ser más fácil entenderlo.

Nombre de dominio

Vamos a la página que más nos guste que nos permita adquirir un nombre de dominio y administrar los DNS y adquirimos un dominio. En este caso he escogido **goddady.**
Una vez tenemos el dominio, vamos al apartado **administar dns** y necesitaremos:

- Un registro A apuntando a la máquina donde esta corriendo el servidor dns2tcp.
- Un registro NS para un subdominio apuntando a la maquina corriendo el servidor dns2tcp.

Tipo	Nombre	Valor	TTL
A	dns2tcpd	52.210.22.7	600 segundos
CNAME	email	email.secureserver.net	1 hora
CNAME	ftp	@	1 hora
CNAME	www	@	1 hora
MX	@	mailstore1.secureserver.net (Prioridad: 10)	1 hora
MX	@	smtp.secureserver.net (Prioridad: 0)	1 hora
NS	@	ns45.domaincontrol.com	1 hora
NS	@	ns46.domaincontrol.com	1 hora
NS	tunnel	dns2tcpd.ad0n.com	600 segundos

En mi caso, esos son los dos registros que he añadido para el dominio ad0n.com.
El registro NS tunnel dns2tpcd.ad0n.com tiene especial importancia, es una de las claves para que ésto funcione y quiere decir que cuando se haga una consulta DNS a tunnel.ad0n.com, el DNS autoritativo para ese subdominio, es decir, el que va a responder, será dns2tcpd.ad0n.com, osea, donde está alojado nuestro servidor DNS fake.

Podemos comprobar que lo tenemos todo bien configurado preguntando a un servidor autoritativo del dominio ad0n.com por el subdominio tunnel.ad0n.com:

```
$ dig @ns46.domaincontrol.com tunnel.ad0n.com
```

```
root@khaleesi:~# dig ns ad0n.com +short
ns46.domaincontrol.com.
ns45.domaincontrol.com.
root@khaleesi:~# dig @ns46.domaincontrol.com tunnel.ad0n.com

; <<>> DiG 9.10.3-P4-Debian <<>> @ns46.domaincontrol.com tunnel.ad0n.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21216
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;tunnel.ad0n.com.               IN      A

;; AUTHORITY SECTION:
tunnel.ad0n.com.        600     IN      NS      dns2tcpd.ad0n.com.

;; ADDITIONAL SECTION:
dns2tcpd.ad0n.com.      600     IN      A       52.210.22.7

;; Query time: 47 msec
;; SERVER: 208.109.255.23#53(208.109.255.23)
;; WHEN: Tue Aug 30 22:06:57 BST 2016
;; MSG SIZE  rcvd: 83
```

Pasamos a ver la configuración del cliente y servidor dns2tcp.

Cliente/Servidor dns2tcp

Servidor:

Abrimos el fichero `/etc/dns2tcpd.conf` y lo configuramos de la siguiente manera:

```
  GNU nano 2.2.6               File: /etc/dns2tcpd.conf

listen = 0.0.0.0
port = 53
# If you change this value, also change the USER variable in /etc/default/dns2tcpd
user = nobody
chroot = /tmp
domain = tunnel.ad0n.com
resources = ssh:127.0.0.1:22 , smtp:127.0.0.1:25
key = 1234
```

Escuchamos en 0.0.0.0 para que sea accesible desde el exterior y en domain ponemos el subdominio que va a delegar en nuestro servidor DNS.
El campo `key` es opcional pero servirá si no queremos que todo el mundo pueda usar nuestro servidor DNS sin clave.

Lanzamos el servidor:

```
$ dns2tcpd -F -d2 -f /etc/dns2tcpd.conf
```

```
root@ip-172-31-16-174:/home/ubuntu# dns2tcpd -F -d2 -f /etc/dns2tcpd.conf
21:23:48 : Debug options.c:97    Add resource ssh:127.0.0.1 port 22
21:23:48 : Debug options.c:97    Add resource smtp:127.0.0.1 port 25
21:23:48 : Debug socket.c:55     Listening on 0.0.0.0:53 for domain tunnel.ad0n.com
Starting Server v0.5.2...
21:23:48 : Debug main.c:132      Chroot to /tmp
21:23:48 : Debug main.c:142      Change to user nobody
```

Cliente

Comprobamos los servicios disponibles del servidor:

```
$ dns2tcpc -z tunnel.ad0n.com -k 1234
```

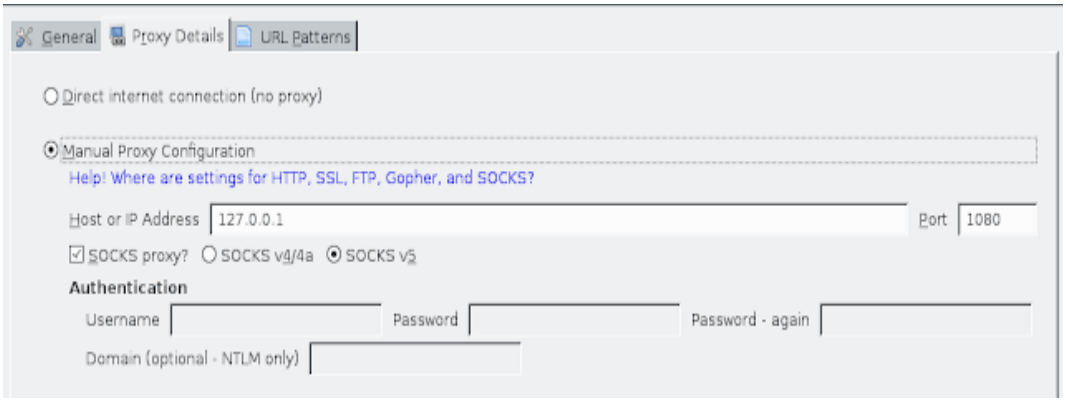
```
root@khaleesi:~# dns2tcpc -z tunnel.ad0n.com -k 1234
No DNS given, using 80.58.61.250 (first entry found in resolv.conf)
Available connection(s) :
        ssh
        smtp

Note : Compression SEEMS available !
```

Levantamos el túnel en el puerto 3333 solicitando el recurso ssh.

```
$ dns2tcpc -z tunnel.ad0n.com -k 1234 -l 3333 -r ssh
```

Nos conectamos por ssh al servidor y hacemos bind al puerto 1080, que será el puerto que usemos en el navegador para nuestro proxy socks.

```
$ ssh -i dns2tcpd.pem ubuntu@localhost -p 3333 -D 1080
```



Listo, ya podemos navegar a través de DNS.
La primera vez se tarda bastante en configurar todo, pero a partir de aquí, lo único que se necesitaría son estos dos comandos:

```
$ dns2tcpc -z tunnel.ad0n.com -k 1234 -l 3333 -r ssh
$ ssh -i dns2tcpd.pem ubuntu@localhost -p 3333 -D 1080
```

y configurar el proxy el navegador y a saltarse los portales cautivos en un momento ;).

11 comentarios :

1. *Anónimo*31 de agosto de 2016, 22:18

   Muy currado, thanks for the info! :D

   Responder

2. *Anónimo*31 de agosto de 2016, 22:27

   no tiene gracia. Si el portal bloquea las consulta al 53, permitiendo solamente sus dns (ej 8.8.8.8 8.8.4.4 y denegar el resto en el firewall) entonces bye bye al tunel dns2tcpc

   Responder
   Respuestas

   1. *Anónimo*24 de octubre de 2017, 16:39

      "... Podemos comprobar si podemos aprovechar esta característica en nuestra WiFi:

      nslookup www.google.com

      Si recibimos respuesta, podemos aprovechar esta vulnerabilidad y encapsular nuestro tráfico en DNS. Para llevar a cabo esta tarea, vamos a usar dns2tcp.... "

      primer apartado

      Responder

3. *Anónimo*31 de agosto de 2016, 22:34

   ja. bloquear el puerto de destino y chao bambino

   Responder

4. 

   *Pacheta*1 de septiembre de 2016, 14:24

   DE PUTA MADRE !

   Responder

5.

   *Maravento Studio*1 de septiembre de 2016, 19:51

   Muy bueno

   Responder

6. 

   *daniel flores*7 de septiembre de 2016, 1:36

   por vpn tambien se puede no? :v

   Responder

7. 

   *Joseph Caceres*23 de noviembre de 2017, 5:05

   puedes usar el puerto 8888 que tambien es escuchado por dns en algunos servidores dns publicos lol

   Responder

8. *Anónimo*11 de marzo de 2018, 17:06

   No funciona, trato de hacerlo con el portal cautivo de pfsense pero no me muestra las conexiones disponibles, de hecho para que las muestre tengo que desactivar el portal cautivo desde el pfsense y poner la sintaxis dns2tcpc -z tunnel.domain.com server.domain.com porque si no espcifico el dominio no funciona