Instalación de un proxy TOR2WEB para acceder a servicios ocultos desde Internet

enero 22, 2015 adastra Deja un comentario Go to comments

Seguramente muchos de los que leéis este blog y os interesan temas relacionados con el anonimato, TOR y cosas similares, conocéis el proyecto “TOR2WEB”, pero para los que no, basta con decir que es una plataforma creada por el equipo de TOR que permite que los servicios ocultos que se publican internamente en la web profunda de TOR se encuentren disponibles desde Internet sin necesidad de que el cliente tenga que arrancar una instancia de TOR en su máquina. Esto quiere decir que cualquier usuario corriente en internet, utilizando cualquier navegador web, podrá acceder a un servicio oculto muy fácilmente.

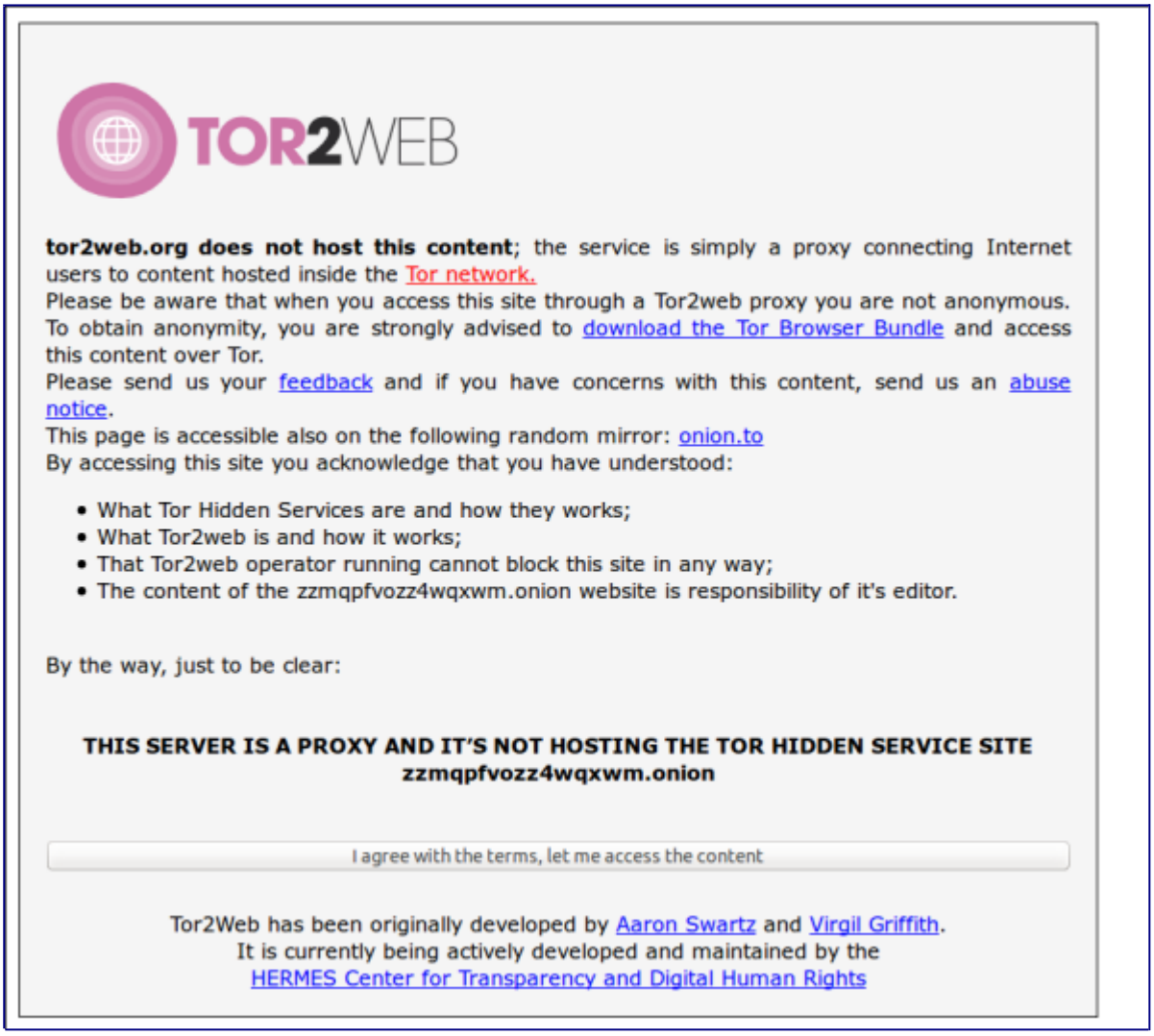
TOR2WEB está diseñado para funcionar como un proxy reverso, que solamente se encarga de enrutar todas las peticiones entrantes desde la “clear web” a la “deep web” de TOR, con lo cual los servicios ocultos siguen siendo ocultos, incluso para un proxy TOR2WEB. No obstante, el cliente que visita dichos contenidos no lo es, a menos claro, que utilice TOR para acceder a dicho proxy algo que que no tiene mucho sentido, ya que al utilizar TOR ya se tiene acceso a la web profunda de TOR y realmente no seria necesario utilizar TOR2WEB.

“tor2web.org” está conformado por voluntarios en todo el mundo que configuran y exponen un servicio de TOR2WEB para que cualquier usuario en Internet pueda utilizarlo. Dado que se trata de un proxy, no almacena los contenidos de los sitios ocultos**,** simplemente se encarga de enrutar las peticiones entre la web profunda de TOR y la “web clara”.

Acceder a un servicio oculto en la web profunda de TOR es simple, solamente hace falta sustituir la cadena “.onion” de la dirección del servicio oculto por “.tor2web.org”.

Por ejemplo, si la dirección onion de un servicio oculto es la siguiente: “ajio7mfsscpvgd23.onion” la dirección que se debe utilizar para acceder a dicho servicio utilizando TOR2WEB es: “ajio7mfsscpvgd23.tor2web.org”.

Lo primero que verá el usuario cuando navega hacia dicho sitio es una nota legal que indica que TOR2WEB es solamente un servicio de proxy que tira de la web profunda de TOR y que los contenidos que el usuario visualizará son responsabilidad del creador del servicio oculto al que se intenta acceder. Si el usuario está de acuerdo con los términos expuestos por el proxy, puede continuar y acceder a los contenidos del servicio oculto. La siguiente imagen enseña enseña dicha advertencia.



Por otro lado, la plataforma también almacena unas estadísticas básicas sobre el número de accesos que ha tenido el servicio oculto el día anterior, por cuestiones de privacidad y para no generar registros sobre el uso de un servicio oculto, solamente se pueden ver las estadísticas del día anterior y no se almacena nada sobre los datos de otros días. Para acceder a dicha información basta con entrar a la uri "/antanistaticmap/stats/yesterday". Por ejemplo: "ajio7mfsscpvgd23.tor2web.org/antanistaticmap/stats/yesterday".
Como se ha mencionado anteriormente, TOR2WEB es un servicio que se encuentra soportado por varios voluntarios en Internet, los cuales configuran una instancia del proxy de TOR2WEB con un dominio propio o con el dominio de TOR2WEB.

Aunque el procedimiento de instalación de un nodo de TOR2WEB es una tarea que se encuentra documentada y bastante bien explicada en la documentación oficial del proyecto, en este artículo explicaré en detalle cómo se puede instalar, configurar y arrancar un nodo de TOR2WEB.

Instalación y configuración de un nodo de TOR2WEB

Antes que nada, el proyecto se encuentra alojado en la siguiente URL: https://github.com/globaleaks/Tor2web-3.0 y se recomienda navegar un poco por cada una de las secciones de la documentación para tener una idea más global del funcionamiento de Tor2Web. En primer lugar es necesario instalar el proyecto y dado que se encuentra pensado principalmente para plataformas basadas en Debian, el mecanismo de instalación consiste adicionar un repositorio a la lista del sistema y posteriormente ejecutar el comando “apt-get” o “aptitute”. Existe un script de instalación que se encarga de realizar todas estas tareas de forma automática y que se recomienda utilizar.

```
>wget https://raw.githubusercontent.com/globaleaks/Tor2web-3.0/master/scripts/install.sh

>chmod +x install.sh

>./install.sh
```

Cuando se ejecuta el script pide permisos de administrador para instalar el programa utilizando “apt-get”. Después de realizar la instalación, automáticamente se crea el directorio “/home/tor2web/certs” que es donde se deberán crear los certificados y las claves para el proxy.
Se deben ejecutar los siguientes comandos dentro de dicho directorio.

```
>openssl genrsa -out tor2web-key.pem 4096

>openssl req -new -key tor2web-key.pem -out tor2web-csr.pem

>openssl x509 -req -days 365 -in tor2web-csr.pem -signkey tor2web-key.pem -out tor2web-intermediate.pem

>openssl dhparam -out tor2web-dh.pem 2048
```

Con los pasos anteriores se encuentra casi terminada la instalación, sin embargo antes de poder arrancar el servicio, es necesario crear un fichero de configuración en “/etc/tor2web.conf”. Dicho fichero no existe por defecto, pero después de instalar Tor2Web, se crea automáticamente un fichero de configuración que sirve de ejemplo en “/etc/tor2web.conf.example”. Las propiedades de configuración que incluye el fichero permiten activar o desactivar varias características interesantes de TOR2WEB, sin embargo, las siguientes son las mínimas que se deben de incluir en el fichero de configuración antes de iniciar el servicio.

nodename: Identificador único del servicio. Es bastante útil para identificar las trazas que se envían a los administradores de TOR2WEB.

datadir: Directorio donde se encuentran los ficheros necesarios para el correcto funcionamiento del programa, incluyendo los certificados creados anteriormente.

processes y ***requests_per_process***: Propiedades que permiten ejecutar el programa utilizando multiproceso. El valor recomendado para la propiedad "processes" es el número de cores del sistema +1.

basehost: Se trata del dominio base donde se ejecutará el proxy. TOR2WEB se puede desplegar en un dominio independiente al de tor2web y en tal caso, es necesario especificar dicho dominio en esta propiedad.

listen_port_http y listen_port_https: Puertos HTTP y HTTPS que serán utilizados por el proxy. Evidentemente los valores por defecto son 80 y 443 respectivamente.

sockshost y socksport: Son probablemente las propiedades más importantes, ya que permiten definir el host y el puerto en el que se encuentra en ejecución un proxy SOCKS de TOR. Es evidente la importancia de dichos valores, ya que TOR2WEB utilizará dicho proxy SOCKS para conectarse con la web profunda de TOR y devolver al usuario final los contenidos del servicio oculto que ha solicitado. En este sentido, para lanzar un nodo de TOR2WEB, es necesario tener una instancia de TOR levantada.

ssl_key, ssl_cert y ssl_dh: Se trata de propiedades que permiten especificar la ruta completa de los certificados y la clave privada que se han creado anteriormente para el servicio.

Con las propiedades anteriores se puede comenzar a utilizar TOR2WEB, sin embargo existen otras propiedades que habilitan características especiales del programa, por ejemplo la posibilidad de bloquear servicios ocultos con contenidos inapropiados, bloquear crawlers, sobre-escribir el fichero "robots.txt" y otras cosas que intentan "alejar" el contenido de servicios ocultos de buscadores como google.

Para conocer en mayor detalle todas las opciones de configuración disponibles, se recomienda revisar la documentación en la siguiente sección. https://github.com/globaleaks/Tor2web-3.0/wiki/Configuration-Guide
Para iniciar un nodo de TOR2WEB con la configuración mínima, se puede utilizar el siguiente fichero

```
[main]
nodename = AdastraTORY
datadir = /home/tor2web
processes = 5
requests_per_process = 100000
listen_port_http = 80
listen_port_https = 443
basehost = tor2web.org
sockshost = 127.0.0.1
socksport = 9150
ssl_key = /home/tor2web/certs/tor2web-key.pem
ssl_cert = /home/tor2web/certs/tor2web-intermediate.pem
ssl_dh = /home/tor2web/certs/tor2web-dh.pem
cipher_list = ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-
```

```
AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:DHE-DSS-
AES256-SHA:DHE-RSA-AES128-SHA:DES-CBC3-SHA
ssl_tofu_cache_size = 100
```

Con el fichero anterior ubicado en “/etc/tor2web.conf” ahora es posible iniciar el servicio como cualquier otro del sistema

```
>sudo /etc/init.d/tor2web start
Enabling Tor2web Apparmor Sandboxing
* Starting Tor2web tor2web…
* Starting tor daemon… [ OK ]
>
```

Con los pasos anteriores, ahora ya contamos con una instancia en ejecución de TOR2WEB en local y disponible para desplegar en algún servidor en Internet.

Saludos y Happy Hack!