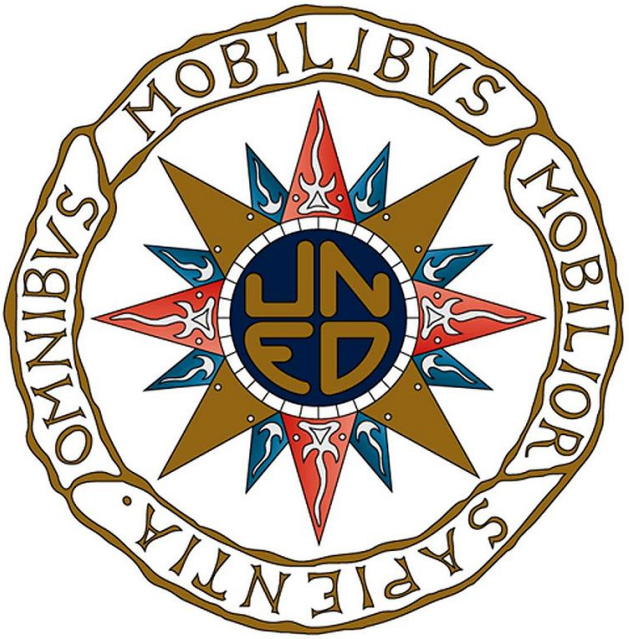
UNED

MÁSTER UNIVERSITARIO EN CIBERSEGURIDAD



Trabajo Fin de Máster

Anonimato y Pentesting con Raspberry Pi

Daniel Martínez Luengo

2021

Máster Universitario en Ciberseguridad

UNED

Trabajo de fin de máster

Anonimato y Pentesting con Raspberry Pi

Autor: Daniel Martínez Luengo

Directores: Antonio Robles Gómez

y

Miguel Romero Hortelano

Curso 2020-2021

Convocatoria junio

Principio de Intercambio de Locard

«Il est impossible au malfaiteur d'agir avec l'intensit que suppose l'action criminelle sans laisser des traces de son passage»,

Que traducido significa:

«Es imposible que un criminal actúe, especialmente en la tensión de la acción criminal, sin dejar rastros de su presencia».

Edmond Locard mantuvo que el contacto con otra persona o elemento de la escena de un crimen, supone un intercambio de materiales físicos, por el que el criminal deja evidencias en la escena y toma elementos de esta que «se lleva consigo».

Dedicatoria

Dedico este trabajo a mi familia, sin ella no habría sido posible, tanto por su apoyo como por sus continuos ánimos. Especialmente a mi mujer, Laura, que siempre me dejaba tiempo libre para dedicarme a mis estudios.

Resumen

Este trabajo se compone de varias partes. La principal se trata de configurar un dispositivo de bolsillo tipo Raspberry Pi en modo routing de tal manera que nos permita conectarnos a Internet de forma anónima a través de la red Tor (The Onion Router). Esto es útil para poder navegar por Internet en países con fuertes medidas restrictivas de libertad de expresión (censura). Como objetivos secundarios, nos centraremos en probar si existe el anonimato al utilizar la red Tor con otros servicios diferentes a los que se usan en visitas a páginas Web como pueden ser FTP, TELNET, etc. Para ello, crearemos varios laboratorios en un entorno controlado empleando herramientas de escaneo y enumeración de puertos y servicios y herramientas de monitorización de dichos escáneres. También, usaremos el router para acceder a la Dark Web y visitar distintas páginas.

Summary

This work consists of several parts. The main part is about configuring a Raspberry Pi pocket device (in routing mode) in such a way that it allows us to connect to the Internet anonymously through the Tor network (The Onion Router). This is useful for surfing the Internet in countries with strong restrictive measures on freedom of expression (censorship). As secondary objectives, we will focus on testing whether anonymity exists when using the Tor network with services other than those used for visiting websites, such as FTP, TELNET, etc. To do this, we will create several labs in a controlled environment using scanning tools and port and service enumeration and monitoring tools for these scanners. We will also use the router to access the Dark Web and visit different pages.

Palabras clave

Raspberry Pi, Tor, anonimato, Deep Web, Pentesting, Honeypot, Kali.

Key words

Raspberry Pi, Tor, Anonymity, Deep Web, Pentesting, Honeypot, Kali.

Índice

Disclaimer

Este trabajo está creado con fines educativos e informativos. No nos hacemos responsables del mal uso que puedan dar al mismo. Las IPs y muchos de los enlaces mostrados en el trabajo son dinámicos o han sido enmascarados.

1. Introducción

A lo largo de la vida de lo que conocemos como Internet (red de redes), la seguridad ha ido evolucionando hasta lo que es ahora en nuestros días. Hace un par de décadas los ordenadores no tenían antivirus; se enviaba todo el tráfico por la red sin ningún cifrado ni confidencialidad ni autenticación; no se usaban firewalls, etc. Todo inimaginable en los tiempos actuales. En estos días, se aplican todas esas medidas y aún así el anonimato queda bastante lejos. Gobiernos con fuertes restricciones bloquean las comunicaciones y espían a sus ciudadanos y a su propia cúpula política, gobiernos unidos en coalición se espían entre ellos para obtener ventajas con respecto a contratos internacionales de cifras multimillonarias (La Vanguardia). Todos estamos ante el ojo del gran hermano, a veces el ojo es un gobierno, a veces el ojo es una multinacional, a veces el ojo es un grupo de hackers, pero cada vez es más complicado tener privacidad. Por estas razones, las empresas y particulares hacen uso de redes anónimas o proveedores de VPN para mantener sus datos protegidos. Aunque existen servicios gratuitos de VPN como Windscribe o redes similares a Tor como Freenet o I2P, en lo que respecta a este trabajo se refiere, la red que usamos para disfrutar de privacidad es la red Tor (Tor Project Inc.). Por lo general, esta red se usa con el navegador Tor Browser (Tor Project Lnc), pero modificaremos Tor para que no solo sirva para enviar tráfico anónimo a través del navegador, sino que todo el tráfico se reenvía de manera anónima. En este trabajo, configuraremos un dispositivo de bolsillo Raspberry Pi capaz de dar anonimato a otras personas que lo necesiten y no dispongan de ningún tipo de conocimiento en la materia y sin que deban instalarse nada, simplemente conectándose al punto de acceso del Router creado. Aprovecharemos también la oportunidad del dispositivo para averiguar cómo funciona la red Tor, navegando en la Surface Web de forma anónima en modo Out-proxy y en la Dark Web en modo In-proxy. También, analizaremos si con dicho dispositivo se logra un anonimato usando otros servicios ajenos a la navegación, como pueden ser TELNET, FTP, etc. Para comprobarlo, realizaremos Pentesting y monitorización del tráfico y analizaremos las evidencias obtenidas.

1.1 Contexto y justificación del Trabajo.

Al igual que otros trabajos que hemos realizado, este en concreto viene de la necesidad de investigar y verificar la seguridad de nuestros sistemas.

En nuestro día a día nos preocupa si nuestra red wifi es vulnerable al ataque de intrusos por la vulnerabilidad "*Krack Attack*". También, nos preocupamos de nuestros procesadores a ver si son afectados por la vulnerabilidad conocida como "*Spectre*". Por si eso no fuera suficiente, cada vez aparecen vulnerabilidades más sofisticadas relacionadas con el kernel del sistema operativo, chips con puertas traseras ya insertadas por hardware por el fabricante en nuestra electrónica de red, Webcams, televisores, móviles, u otros dispositivos con vulnerabilidades que permiten ser usados para denegaciones de servicio distribuidas. Y muchos casos más.

La motivación de investigar y verificar el anonimato de la red Tor, surge de dos necesidades reales. Por una parte, la necesidad de anonimato en países con fuertes medidas de control, incluso con penas de cárcel, simplemente por disponer de libertad de expresión y querer usar redes sociales. Con este dispositivo no son necesarios conocimientos informáticos para poder navegar de forma anónima, tanto en la Surface Web como en la Dark Web, puesto que dispone de un punto wifi al que conectarse sin ningún requerimiento más por parte del usuario. La segunda necesidad comenzó tras recibir una y otra vez ataques de todo tipo durante mucho tiempo, siempre encubiertos por Proxies, VPNs, red Tor, etc. Muchos de esos ataques se intentaron bloquear enviando correos a direcciones de uso abusivo a los ISP de varias personas y empresas que, sin saberlo, estaban compartiendo su conexión a una Botnet o utilizando la red Tor. Pero los ISPs no pueden bloquear algo anónimo, de tal manera que no es el legítimo dueño el que está realizando esas actividades, sino que es alguien oculto en los sistemas de anonimización ya nombrados. Este trabajo permitirá comprobar si el sujeto que hace un mal uso de una red Tor, podría perder el anonimato al usar otros servicios de escaneo y ataque de vulnerabilidades y si es posible des-anonimizar en algunos casos concretos.

1.2 Objetivos del Trabajo.

El objetivo principal es configurar un dispositivo portátil Raspberry Pi, capaz de realizar las funciones de router enviando el tráfico a través de la red Tor de forma anónima. Dicho router, tendrá la posibilidad de conectarse mediante un RJ45 a otro router/switch que proporcione Internet a través de DHCP; o de un módem 3G; todo ello de forma automatizada. Para la conexión al router, proporcionaremos un servicio wifi como Hotspot, de esta manera, cualquier dispositivo se podrá conectar a la red Tor y navegar en la Darknet.

Los objetivos secundarios son los siguientes:

- Comprobar si la conexión es anónima o si por el contrario al realizar Pentesting o conexiones envía información de la IP original.
- Analizar navegación In-proxy (Dark Web) y Out-proxy (Surface Web).
- Configurar Raspberry como dispositivo Pentesting emulando las primeras fases del Hacking ético.
- Configurar Raspberry como dispositivo Honeypot (DinoTools) y con monitorización de red, para recoger los ataques recibidos por la otra Raspberry.

1.3 Enfoque y método seguido.

Para realizar este trabajo se han seguido varios métodos. Para la parte del desarrollo de los objetivos y para la parte de resolución de problemas, hemos usado un enfoque ingenieril (Web ciencia). El método ingenieril aplicado consiste en definir el problema, imaginar el producto, planificar y diseñar, probar el prototipo, mejorar el producto y obtenerlo. Si el resultado se puede mejorar aún más, volver a la fase de definir el problema, de tal manera que el proceso sea cíclico hasta alcanzar el objetivo. Otro método utilizado es el visto en la asignatura "*Hacking Ético*" impartida en el Máster Universitario en Ciberseguridad y de la certificación CEH (Certified Ethical Hacking) (EC-Council) para aplicar hacking ético, aunque esta metodología es muy similar a otras. Las fases son las siguientes: Reconocimiento, escaneo, acceso, mantener acceso y cubrir huellas, aunque en este trabajo solo usamos las fases de Escaneo y acceso, ya que las otras no eran necesarias para comprobar el anonimato. También, se ha utilizado la metodología vista en la asignatura "*Auditoría*

y Monitorización de la Seguridad" impartida en el mismo máster comentado para la parte de monitorización y Honeypot. Para la parte legal, se ha recurrido a la parte de la asignatura de "*Ciberilícitos*" también impartida en dicho máster.

- Metodología ingenieril:

El primer paso será definir nuestras necesidades que son las siguientes: Un router portátil para conectar a la red Tor, un dispositivo capaz de recoger las conexiones recibidas, un dispositivo capaz de lanzar conexiones de distintos servicios y poder acceder a la red Tor de manera Out-proxy simplemente para disponer de anonimato, e In-proxy para poder acceder a la Darknet (Test de velocidad).

Para imaginar el producto, revisamos el estado del arte de los distintos proyectos relacionados con el mismo que hayan sido producidos por la Raspberry Pi y que cumplan las necesidades comentadas. En esta fase decidimos que Raspberry Pi era nuestro candidato como dispositivo. Para ver su rendimiento y adaptabilidad en cada rol (router, Pentester y Honeypot) se adquirieron tres modelos diferentes en función de su memoria y procesamiento y se eligió la mejor configuración en función de sus características. Ver estado del arte.

Realizamos un diseño en función del producto imaginado y con él la planificación en función de las fechas, marcando hitos importantes. Ver planificación.

A medida que vamos obteniendo parte del producto pasamos a la fase de pruebas y continuamos con la creación del siguiente producto. Comenzamos con el objetivo principal y el que más tiempo nos ocuparía, ya que tiene muchas acciones a realizar. Cuando terminamos parte de su configuración, pasamos a la siguiente fase para asegurarnos de que cumple con los requisitos de los laboratorios. Posteriormente, realizamos este mismo proceso con la Raspberry Pi de monitorización y luego con la Raspberry Pi de Pentesting. Ver entorno de pruebas.

Cuando comprobamos que nuestros productos satisfacen nuestros objetivos, volvemos a la fase inicial e incluimos mejoras en los mismos o características que nos aporten calidad. Para comprobar los objetivos, las pruebas que realizamos son las de conectarnos de forma anónima e In-proxy al Honeypot desde un dispositivo móvil y desde un portátil, saliendo a la red Tor con el router conectado tanto por

RJ45 como por modem. Posteriormente, cuando logramos este objetivo, configuramos los otros dos, primero el Honeypot y posteriormente el pentester (con Kali). Una vez terminada la fase anterior, pasamos de nuevo al entorno de pruebas. Se establece la monitorización del tráfico en el Honeypot y lanzamos escaneos y ataques de red con la Raspberry Kali. Durante todo el proceso recogemos los datos y documentamos todo. Ver laboratorios.

- Metodología Hacking ético:

En la metodología para la parte de pruebas de Pentesting, se seleccionan las fases 2 y 3. La fase 1 reconocimiento no es necesaria porque ya sabemos la IP a la que vamos a lanzar la batería de escaneos y ataques y también sabemos que no existen más IPs, usuarios, direcciones de correo en ninguna otra parte. Nos centramos en las fases 2 y 3 porque son las que más pueden ayudarnos a comprobar el anonimato. La fase 2 consiste en el escaneo y enumeración. De esta manera, encontraremos los servicios en escucha y lanzaremos sobre ellos los ataques. La fase 3 tampoco era necesaria, pero la hemos añadido para intentar lograr obtener más evidencias de anonimato. Para realizar el hacking ético se barajaron y probaron varias herramientas. Finalmente se usó la distribución de Kali Linux ARM y las herramientas Nmap, Nikto, Rapidscan para la parte de escaneo y Armitage para la parte de ataque.

- Metodología monitorización:

En la monitorización existen buenas prácticas que indican que no por disponer de más información tendremos mejores resultados, sino que es mejor realizar un análisis previo y centrarse en recoger lo que nos interesa de verdad. Por esa razón, analizamos nuestro diseño para encontrar nuestras necesidades principales que eran dos, poder publicar una elevada cantidad de servicios a Internet y recoger las conexiones en tiempo real sin tener que revisar logs, sino que pudiéramos ver de forma interactiva y gráfica dichas conexiones. Esto nos ayudó a encontrar las herramientas necesarias, Dionaea como Honeypot y NTopng como monitor de red.

1.4 Breve sumario de productos obtenidos.

Los productos obtenidos podrán dar un servicio independiente cada uno, por esa razón, podemos decir que obtendremos tres productos físicos con un rol diferenciado entre sí. Uno hará de enrutador a la red Tor, otro hará de equipo de test de penetración para hacking ético y otro hará de Honeypot, monitorización y análisis. También, se entregará una memoria que incluirá el montaje, los resultados de las pruebas y las conclusiones obtenidas. A parte de eso, un breve estudio de la red Tor, su funcionamiento y su contenido.

Resumen de los productos obtenidos:

1. Raspberry Pi 4 8GB: Honeypot de tamaño reducido.
2. Raspberry Pi 4 2GB: Router proveedor anonimato con conexión 3G/Ethernet.
3. Raspberry Pi 3: Kali Linux para Pentesting.
4. Memoria, resultados y conclusiones.
5. Estudio de la red Tor y la Dark web.

En la siguiente figura, podemos ver las tres Raspberry Pi utilizadas.

Figura 1: Las tres Raspberry utilizadas

En la siguiente figura podemos ver un esquema general de lo obtenido.

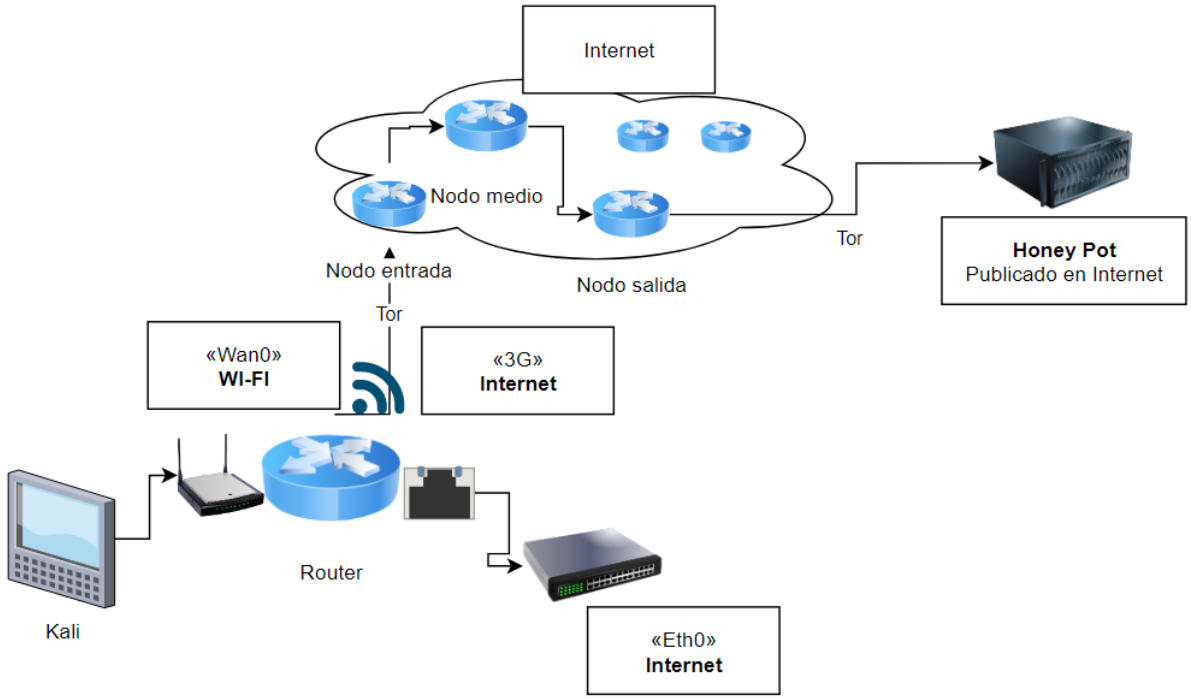


Figura 2: Esquema final del TFM

1.5 Breve descripción de los otros capítulos de la memoria.

En los próximos capítulos hablaremos de los siguientes temas:

Capítulo 2. Capacidades de las Raspberry y la red Tor. Estado del arte.

Capítulo 3. Montaje del entorno. Aquí comentaremos cómo instalamos los distintos sistemas operativos de cada uno.

Capítulo 4. Navegación en Tor. Realizaremos un análisis de nuestra experiencia navegando en la Deep web, contenido y riesgos encontrados.

Capítulo 5. Laboratorio. Aquí mostraremos los laboratorios realizados para llegar a nuestros objetivos.

Capítulo 6. Planificación y presupuesto del proyecto.

Capítulo 7. Conclusiones obtenidas y líneas futuras. Hablaremos de los resultados obtenidos en las pruebas, problemas encontrados y las conclusiones finales acerca de los objetivos. Por último, las posibles líneas de investigación futuras.

2. Elementos hardware y software utilizados

2.1 Estado del arte y elección.

En el mercado existen varios productos similares en prestaciones a Raspberry Pi y Tor. A parte de Raspberry Pi, los dispositivos que podrían ser compatibles con este trabajo son los siguientes: NVIDIA Jetson Nano, ASUS Tinker Board S, Rock64 Media Board, ODROID-XU4, Banana Pi BPI M3 o NanoPi M4 (Rodrigo). Muchos de ellos son más potentes, pero también más caros y con menos comunidad dando soporte. Con respecto a las redes anónimas, existen varias alternativas a Tor, algunas más modernas y mejor optimizadas. Las dos principales son I2P y Freenet (Díaz e Incibe), aunque hace pocos años han aparecido otras alternativas que buscan hacer competencia a Tor, como puede ser Loopix (Crespo).

Se ha decidido usar Raspberry por su compatibilidad con muchos dispositivos a la hora de interconectar un modem 3G/4G, y también por la gran cantidad de manuales creados por la comunidad de usuarios a nivel mundial. Muchas de las herramientas necesarias para las funciones realizadas se han probado por la comunidad, lo que garantizaba el éxito de gran parte de ellas. También, permite realizar diversidad de servicios como consola, enrutador, servidor de impresión, Smart TV, ordenador personal, firewall, Honeypot, cámara de vigilancia y controlador para robótica. Lo que permite darle un segundo uso al finalizar este trabajo.

Aunque la selección de usar la red Tor estaba decidida desde antes de comenzar este proyecto, para realizar el anonimato, se han valorado distintos productos gratuitos como I2P o Freenet, sin embargo, para este trabajo se necesitaban ciertos conocimientos ya desplegados en la comunidad, así como ciertas compatibilidades que proporciona Tor. Freenet se descartó porque usa los recursos como datastore, lo que puede provocar problemas legales al no saber qué contenido, ni quien lo está almacenado, ya que cada vez se almacena más contenido ilegal (Echeverri e Incibe). También, se valoraron otros productos como VPNs de pago o gratuitas (Guy Fawkes y Vpn Mentor) que sin duda ofrecen el mismo servicio y en muchos casos mejor, pero uno de nuestros objetivos es revisar el grado de anonimato de la red Tor. Algunas de estas VPNs son gratis, pero no garantizan lo que se hace con el tráfico que viaja por su infraestructura. El uso de

una VPN de pago, a emplear en sustitución de Tor, se anotó como líneas futuras al ser interesante utilizar este mismo router, pero configurado con una red segura y confiable. Otra opción que se valoró es Tails (Tails non-profit organization), un sistema operativo en el que ya viene integrado Tor y que todo el tráfico va direccionado por la red Tor, pero no se encontró una versión para ARM. Por último, comentar que ya existe un producto llamado Onion Pi (Lady Ada y Adafruit) que realiza funciones similares a nuestro router, pero que también se ha descartado debido a que no cumple el requisito de la conexión 3G/4G, ni otros requisitos del proyecto como el enrutamiento desde Ethernet y wifi.

2.2 Raspberry Pi.

La Raspberry Pi es un dispositivo de bolsillo, de tamaño similar a una tarjeta de crédito y con una altura de un par de centímetros. Este dispositivo se compone de microprocesador, memoria, tarjeta de red, tarjeta wifi, Bluetooth, entrada de sonido, entradas USB, puerto HDMI y entrada de alimentación. También dispone de unos pines llamados "*GPIO*" (Fundación Raspberry Pi) capaces de interactuar con otros circuitos eléctricos/electrónicos y son programables con el lenguaje Python a través de su sistema operativo. Esto lo convierte en un dispositivo multifuncional. Ver líneas futuras.

A continuación, vamos a presentar unas breves características técnicas de los dispositivos Raspberry Pi y porque se usará cada uno para un rol determinado, enrutador, atacante o Honeypot. En este caso hay dos modelos, el 3 B y el 4 B. Ver Anexo A.

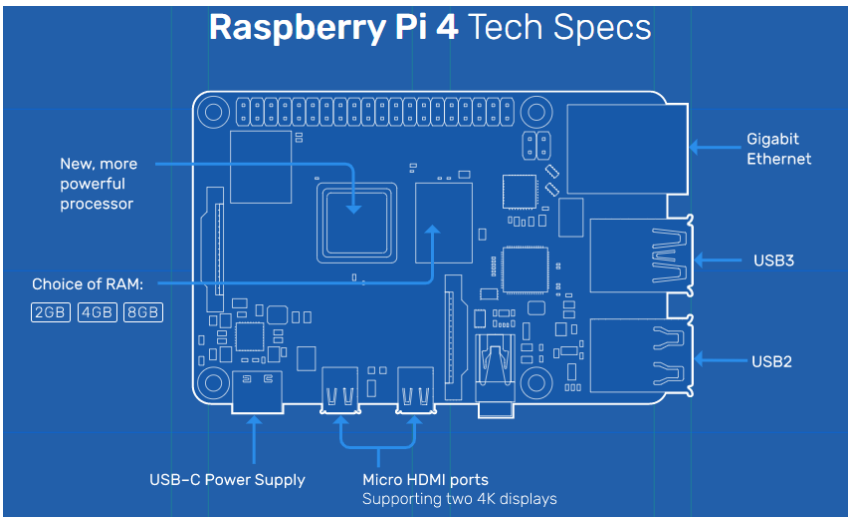


Figura 3: Hardware Raspberry Pi 4 modelo B. (Fundación Raspberry Pi)

El modelo Raspberry Pi 4 está compuesto por un procesador ARM con cuatro núcleos a 1,5 GHz de velocidad y memoria a LPDDR4. Para este trabajo, disponemos de una Raspberry Pi 4 con 2 GB de memoria y otra con 8 GB. Dispone de entrada Gigabit Ethernet. También dispone de Wireless IEEE 802.11ac que soporta 2,4 GHz y 5 GHz que viene con módulo Bluetooth incorporado. Para más detalles, Ver Anexo A.

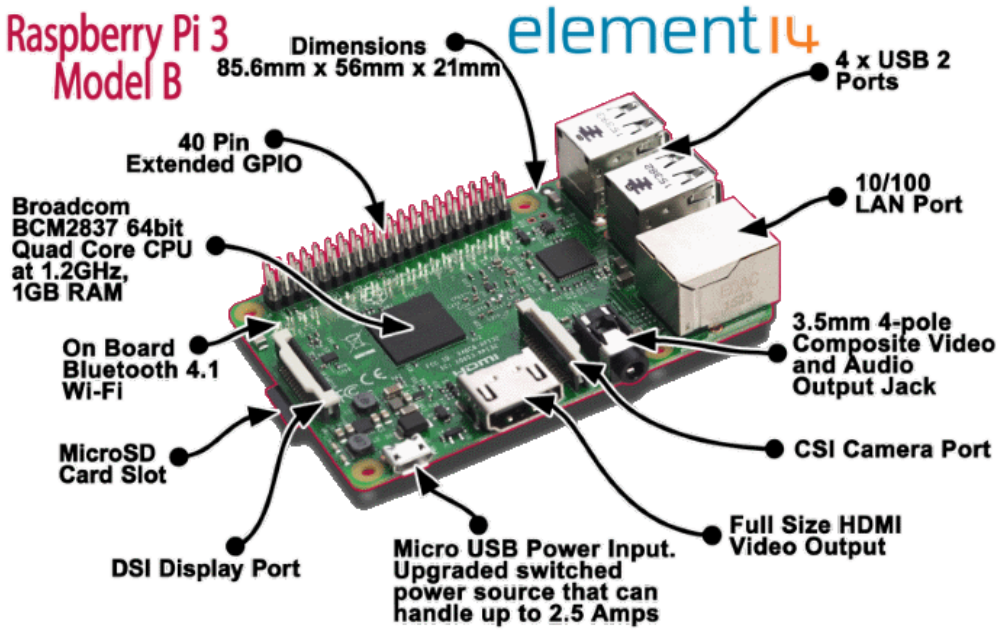


Figura 4: Hardware Raspberry Pi 3 modelo B. (Comunidad Element14)

En el modelo 3, encontramos que las características son más limitadas. Tenemos un procesador ARM con cuatro núcleos y una velocidad de 1,2 GHz. De memoria disponemos de 1 GB LPDDR2. La tarjeta de red es ethernet a 100 Mbits/s.

El módulo WIFI soporta 2.4GHz IEEE 802.11.b/g/n y Bluetooth 4.1 (Fundación Raspberry Pi).

En total, disponemos de tres dispositivos, uno de ellos con gran capacidad de memoria al tener 8 GB, otro algo más reducido con 2 GB, pero con igual procesamiento y características de red y otro más limitado en todo, con menos memoria, menos procesador y menos capacidades de red.

Para realizar una distribución de tareas, necesitamos saber los requisitos técnicos de cada uno de los roles. Después de varias pruebas, vemos que el rol principal de enrutador que se encargará de proporcionar el anonimato y el acceso a la red a otros dispositivos no requiere mucha memoria, pero sí mejores velocidades de red. Por esta razón, el dispositivo encargado del enrutamiento será la Raspberry Pi 4 de 2GB. Para la tarea de rol de atacante y navegación en la Deep Web, se seleccionó la Raspberry Pi 3 B, ya que no se requieren grandes recursos a menos que se utilicen herramientas más potentes como OpenVas, Nessus, Acunetix, etc. Por último, y dado que nos interesa recoger el tráfico de los ataques recibidos en nuestro Honeypot, éste tendrá que tener gran cantidad de memoria para poder publicar más servicios y poder ejecutar más herramientas de análisis y monitorización.

2.3 Tor Project.

En la década de los 90, tres personas del laboratorio de investigación naval de estados unidos (NRL), en respuesta a la pregunta de si había alguna manera de no revelar la identidad de quien hablaba con quién en Internet, se les ocurrió el despliegue de los primeros diseños y prototipos de enrutamiento de cebolla, así, protegerían la información de los departamentos de inteligencia y el tráfico que enviasen a través de Internet. De esa idea, décadas después y gracias al estudiante Roger Dingledine, del instituto de tecnología de Massachusetts (MIT), apareció el proyecto Tor (The Onion Routing). La idea del enrutamiento en cebolla es el enrutamiento en capas. Se trata de enrutar el tráfico a través de diferentes caminos y en cada paso, cifrarlo de tal manera que el resto de los nodos no sepa por los que pasa ni de donde procede.

Es importante en este punto comentar que, en la navegación Out-proxy, existen tres tipos de routers en Tor, el de entrada, el de medio (middle) y el de salida. Asignar el rol de router middle es lo normal y no tiene complejidad, recibimos la información del router de entrada y se la enviamos al de salida. Sin embargo, el router de entrada tiene la responsable labor de saber la IP del que hace uso de la red, por esta razón es el rol más difícil y con más requisitos a la hora de que un nodo se establezca como tal. Para poder ser nodo salida también requiere de unos requisitos, ya que aunque no mostrará la IP de entrada, si que sacará el tráfico a Internet sin cifrar al ser el último router, con lo que si hay contraseñas e información sensible enviadas en claro, se podrá obtener sin ninguna restricción (J. Menéndez y Una al Día. Hispasec) (CristianRus4 y Xataca) (CCN-CERT).

Cuando pasaron los años y Tor fue ganando popularidad y subvenciones, se decidió desarrollar un navegador para que el uso de la red fuese más sencillo. El navegador está basado en Firefox y se llama Tor Browser. El uso del mismo no es necesario para usar la red Tor, pero facilita mucho el trabajo si lo que se pretende solo es navegar. Existen dos tipos de navegación en la red Tor llamadas Out-proxy e In-proxy. El primer tipo se usa para navegar por Internet de forma anónima y el segundo para navegar dentro de la red Tor usando los enlaces .onion. A esta red solo se puede acceder a través de Tor. En ambos casos, se usan lo que se conoce como circuitos, que es la creación del enrutamiento del tráfico a través de tres routers.

2.4 TorBrowser, Tails, socats, Anonym8, Redsocks, proxychains, TorGhost.

Ya hemos visto para qué sirve y cómo funciona Tor, básicamente para darnos la posibilidad de disfrutar del anonimato en la navegación redireccionando el tráfico a un puerto de Tor. Para ello, lo más sencillo es usar Tor Browser. Se trata de un navegador creado por la fundación Tor Project y basado en el código fuente del navegador firefox el cual se ha modificado para que pueda funcionar con Tor. Esto facilita mucho el uso de Tor por parte de usuarios inexpertos. El problema es que solo redirecciona el tráfico de navegación, el resto de tráfico queda sin anonimizar.

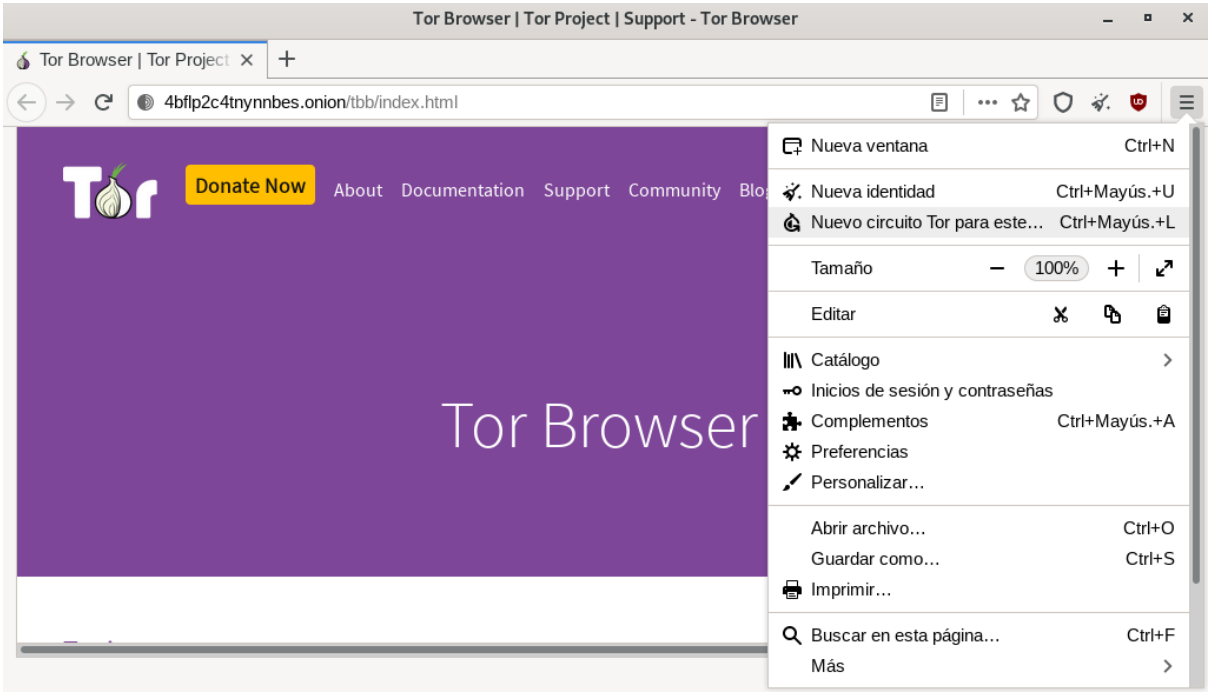


Figura 5: Navegador Tor Browser

Para navegar por Internet de forma anónima se puede utilizar "*Tor Browser*", aunque hay que configurar ciertos parámetros (como no usar "*cookies*") para obtener más anonimato. Por otro lado, tenemos la problemática de que no todo es navegación, también disponemos de resolución de nombres, conexiones FTP, Telnet, SSH, etc, que no se envía por el navegador, sino a través de la conexión normal. Para realizar este tipo de conexiones con otros puertos de manera anónima, existen varias herramientas que se encargan de redireccionar todas las peticiones de salida a través de Tor. Aunque finalmente se decidió realizar el redireccionamiento de todos los puertos usando "*Iptables*", las herramientas analizadas nos ayudaron a entender su funcionamiento. Vamos a comentar algunas de ellas:

Tails (The Amnesic Incognito Live System) (Tails non-profit organization), es un sistema operativo que usa Tor para producir anonimato. Usa una distribución Linux Debian, pero también permite instalarse en otros sistemas. Viene con Tor Browser activado y también redirecciona todo el tráfico de otros puertos usando Tor.

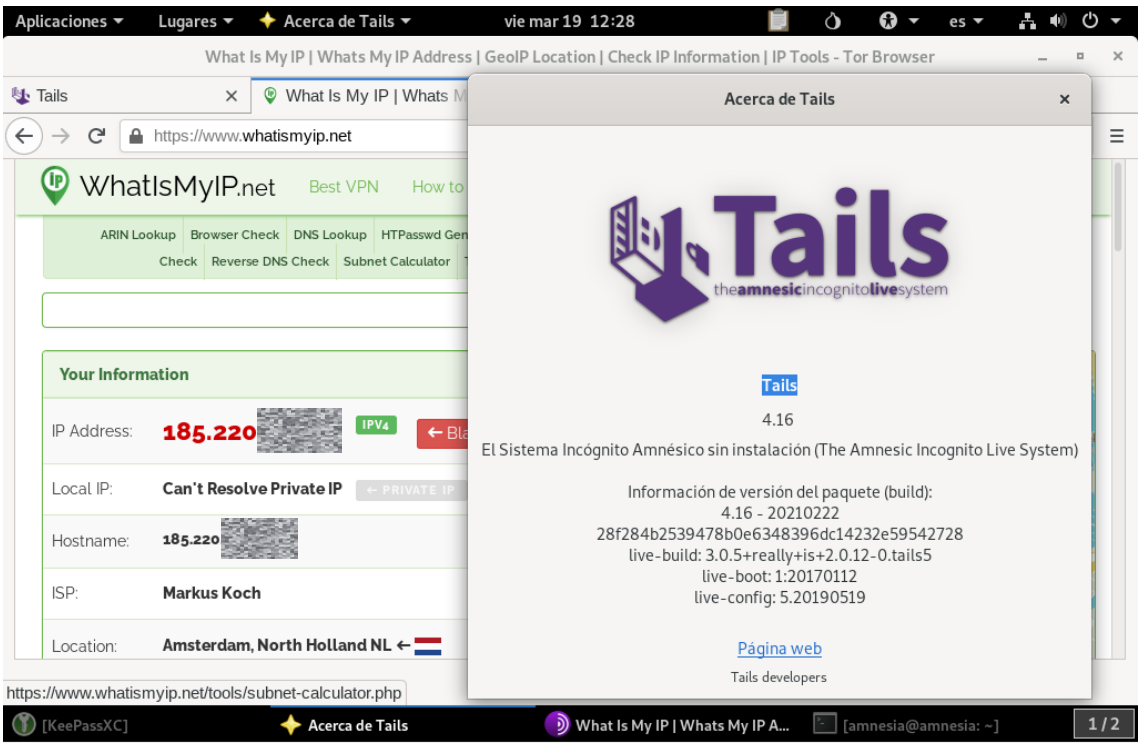


Figura 6: Sistema operativo Tails (Tails non-profit organization)

En la siguiente figura, podemos ver que, si actualizamos el sistema operativo, accede a direcciones .onion de la red Tor. También vemos uno de los problemas de la red Tor, la lentitud en la velocidad de descarga.

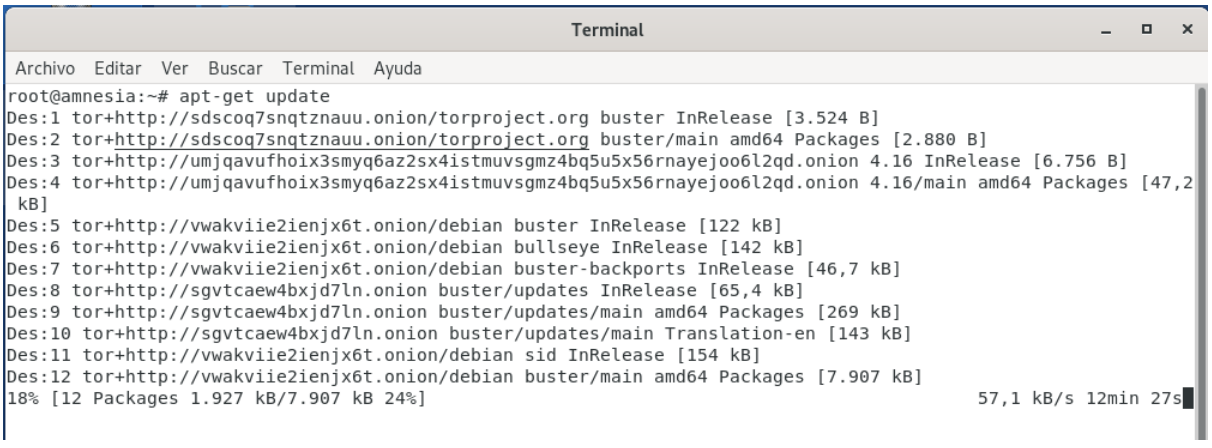


Figura 7: Actualizaciones con Tails a la red Tor

Socats: Socats (SOcket CAT) (Dest-unreach.org), es una herramienta de retransmisión multipropósito. Se puede utilizar para muchas tareas de ciberseguridad, pero una de ellas, la relacionada con Tor, es que se puede realizar "*port forwarding*" y podemos redireccionar, por ejemplo, el puerto 22 al puerto que

usemos de Tor, de esa manera, muestra salida a través de SSH se haría por Tor de forma anónima.

Anonym8: (Hiroshiman) Esta herramienta sirve para crear un proxy a la red Tor. Permite distintas funcionalidades y dar más anonimato cambiando la Mac de la tarjeta, uso de UDP, etc.

RedSocks: (Evdokimov) Esta herramienta permite redirigir todas las conexiones TCP a un proxy o a SOCKS4-5 usando las "*Iptables*" de Linux . Esto permite junto con Tor, redireccionar todo el tráfico y de esa manera, aumentar el anonimato debido a que ningún paquete irá fuera de la red Tor.

Proxychains: Al igual que las anteriores herramientas comentadas, proxychains son, como su nombre indica, Proxies encadenados (Hamsik). Esto en ciberseguridad tiene varias utilidades como pivotar, pero en tema de anonimato también se usa para redirigir el tráfico de una aplicación para que funcione en la red Tor. No solo hace de proxy, sino que es capaz encapsular el tráfico a la red Tor.

TorGhostNG: TorghostNG (Luzthedev et al.) es un script de Python que usa "*Iptables*" para redirigir todo el tráfico que quiera salir a Internet a través de la red Tor. Es muy sencillo de utilizar, pero a la vez muy potente a la hora de lograr el anonimato.

Durante el desarrollo de este trabajo se han investigado y probado varias de ellas llegando a la conclusión que lo más idóneo es usar "*TorGhostNG*" si lo que se quiere es automatizar el proceso, sin embargo, para que todo fuese compatible y aprender cómo funciona exactamente el proceso, hemos configurado todo el tráfico de manera manual mediante *"Iptables"* (lo mismo que hace "*Torghost*").

2.5 Surface Web, Deep Web y Dark Web.

En relación a la navegación anónima con Tor, aparecen a menudo tres conceptos: Surface Web, Deep Web y Dark Web. Vamos a aclarar cada uno de ellos.

Surface Web (Test de velocidad): Es todo el contenido que suele estar indexado por los buscadores de Internet como Google, Duckduckgo, Yahoo, Bing, etc. En dichos buscadores podemos encontrar todo tipo de contenido, desde

periódicos, hasta venta de entradas, películas en streaming, manuales, etc. Aquí podemos navegar de forma normal con el navegador de nuestro sistema operativo, ya sea con el móvil, Tablet, PC o portátil. O podemos navegar de forma anónima con Tor o software similar, usando un proxy gratuito o de pago, usando una VPN gratuita (VPN del navegador Opera) o de pago, etc.

Deep Web (Test de velocidad): Es el resto de contenido que no se ha indexado a través de los buscadores, sin embargo, está publicado en Internet, ya sea a propósito o por desconocimiento del propietario. Aquí podemos encontrar servicios publicados, documentos, ficheros de backup, cámaras de seguridad y multitud de servicios más. Por esa razón, se habla de que más del 90% del tráfico se encuentra ahí. Con la página Shodan, podemos realizar búsquedas de servicios no indexados. Este servicio no se dedica a indexar como otros buscadores, sino que se dedica a escanear servicios levantados y permite realizar búsquedas de los mismos a través de su portal Web o directamente mediante el uso de APIs.

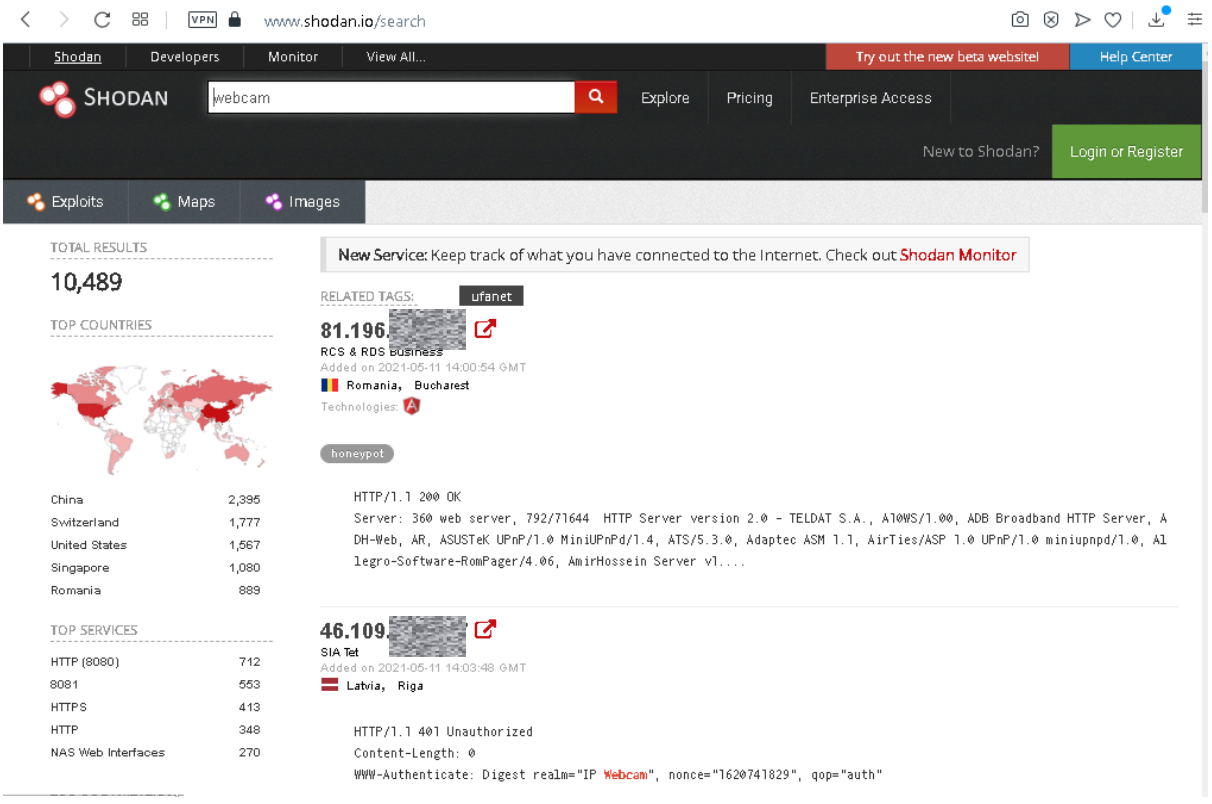


Figura 8: Ejemplo encontrado en la Deep Web mediante Shodan

Por último, hablaremos de la **Dark Web** (Test de velocidad). Para poder acceder a las páginas web publicadas en la Darknet, es necesario usar herramientas

especiales. La herramienta Tor se puede usar para tener anonimato en la Surface Web (Out-proxy) o para navegar por la Darknet y entrar en las Web ocultas. Tor dispone de un navegador llamado Tor Browser mediante el cual es posible resolver dominios .onion que son los usados en la Darknet de Tor. Cuando usamos Tor para navegar a sitios .onion hablamos de navegación In-proxy de Tor. También podemos usar el Router configurado en este trabajo para navegar In-proxy sin necesidad de instalar nada, simplemente conectándonos a su punto de acceso wifi.

La siguiente figura ilustra sobre el tipo de contenido de cada Web.

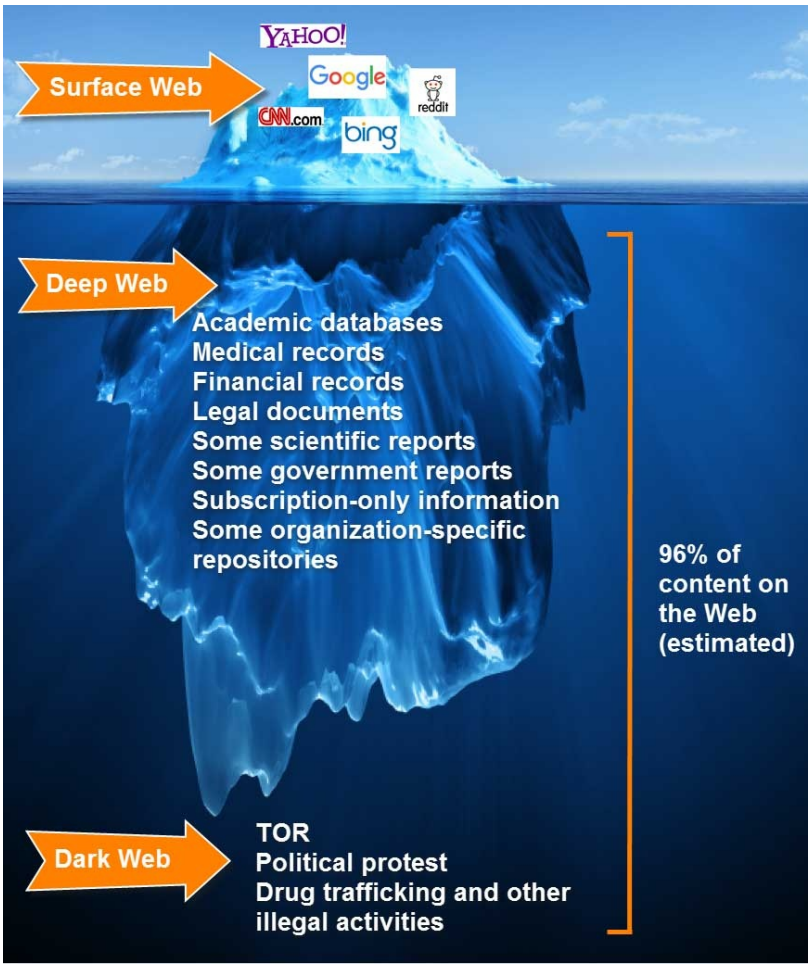


Figura 9: Iceberg Surface, Deep and Dark Web. (Test de velocidad)

2.6 Análisis de los Honeypot.

Durante el análisis, encontramos múltiples opciones de Honeypot. Disponemos de la imagen preconfigurada Honeeepi. Se probó, pero es del 2016 y hay cierto tipo de cosas que no funcionan muy bien y las herramientas que incluye y nos interesan, las podemos instalar nosotros para no perder el control. Se probó

esta imagen y también otras herramientas instalándose en Raspbian. Pasamos a describir cada una de ellas:

Tango honeypot intelligence with Splunk (Aplura): Año 2015-2016, aunque hace tres años se actualizó. Es un Honeypot que utiliza Cowrie y una versión Enterprise de Splunk. Se descartó por evitar tema de licenciamiento.

Honeyd (Datasoft): Año 2015-2016, aunque tiene una web en la que aparece una versión de hace cuatro años. Esta herramienta simula host virtuales lo que la convierte en candidata ideal para emular una empresa completa con su pool de IPs, etc. Se descartó por no ser el objetivo de este trabajo.

Honeyview: Esta herramienta sirve para mostrar de forma gráfica los resultados obtenidos de Honeyd. También se descartó porque no utilizamos Honeyd.

DShield (Guy Bruneau y SANS ISC): Honeypot de la empresa SAN. Se descartó porque envía los resultados a una web externa. Se anotó para futuros proyectos.

TPOT (Telekom-security): Proyecto que incluye diversos Honeypots junto con herramientas NSM. Se probó en nuestra Raspberry Pi y funcionó, pero se descartó por la cantidad de herramientas que no íbamos a utilizar. Se decidió emplear herramientas similares a las que usa, pero solo las necesarias.

Kippo (Tamminen y Desaster): Honeypot orientado a recibir ataques de fuerza bruta del cliente SSH. Es capaz de realizar interacción con los ataques. Se descartó al ser de alta interacción, ya que nosotros buscamos una de baja interacción simplemente para registrar los accesos, no ataques de fuerza bruta de usuarios, ni interacción con la consola SSH.

Honeytrap (Honeytrap): Al igual que el anterior, sirve como Honeypot de servicio SSH.

Honeypy (Px Mx y Foospidy): Honeypot instalable en Python que utiliza Dockers para emular los servicios.

Cowrie: Emula servicios de SSH y Telnet. No nos funcionó en Raspberry así que lo descartamos.

Glastopf (MushMush): Simulador de servicios Web, se podría usar para el tema de revisión de cookies, etc y para obtener más información del atacante y de los métodos utilizados. Tampoco nos funcionó en Raspberry.

Conpot (MushMush): Interesante emulador de servicios industriales. No aplicaba a este trabajo, pero no se descarta para futuros proyectos.

Sweetsecurity (Smith y TravisFSmith): Más que un Honeypot se podría decir que es un NSM con Bro, ElasticSearch, Kibana, etc. Se podría utilizar para recoger datos y ataques, pero nos dio errores en algunas partes de la instalación.

Dionaea (DinoTools): Potente Honeypot capaz de emular bastantes servicios. Al final nos decantamos por este al ser un Honeypot de baja interacción y como ya se comentó, es lo que nos interesa en combinación con NTopng para registrar el tráfico.

NTopng (NTop Company): Para ver el flujo de las conexiones entrantes y salientes hay diversas herramientas. Se optó por NTopng al ser compatible con Debian y Dionaea y permitir ver en tiempo real las conexiones realizadas al dispositivo Honeypot.

Wireshark (Wireshark Foundation): Es una herramienta muy versátil de análisis de tráfico de red. La usaremos para poder investigar el tráfico a más bajo nivel. Permite ver el tráfico generado en cada trama y su contenido a nivel de detalle. Solo se usará en casos puntuales para confirmar evidencias y para almacenar el tráfico. Al almacenar el tráfico en un fichero pcap, lo podremos diseccionar más adelante desde cualquier otro dispositivo. Ver líneas futuras.

Con la herramienta "*Dionaea*" podremos recoger los test de intrusión realizados por la Raspberry Kali, ya que nos simulará una serie de servicios. Con la herramienta "*NTopng*" podremos ver las IPs de origen para comprobar que todo el tráfico llega enmascarado por la red Tor. Con la herramienta "*Wireshark*" podremos analizar las evidencias más a fondo y guardar el tráfico capturado.

3. Montaje del entorno de pruebas

Para el montaje del entorno de pruebas, se van a dividir las tareas de modo independiente, de tal manera que se pueda reproducir alguno de los productos resultantes de este trabajo de forma independiente. En los siguientes puntos, veremos el montaje de las imágenes de los sistemas operativos de los distintos dispositivos, su configuración y las pruebas realizadas.

3.1 Instalación de imágenes.

Para cada dispositivo hay que descargar una imagen en concreto. Raspbian viene añadida en la herramienta Imager para crearla y pasarla a la tarjeta micro SD, sin embargo, otras como Kali Linux (Kali org), hay que descargarla del propio proveedor. Vamos a comentar la instalación de las imágenes, para ver el proceso en más detalle lo reflejaremos en los Anexos. La aplicación para instalar las imágenes en Raspberry Pi es Raspberry Pi Imager.

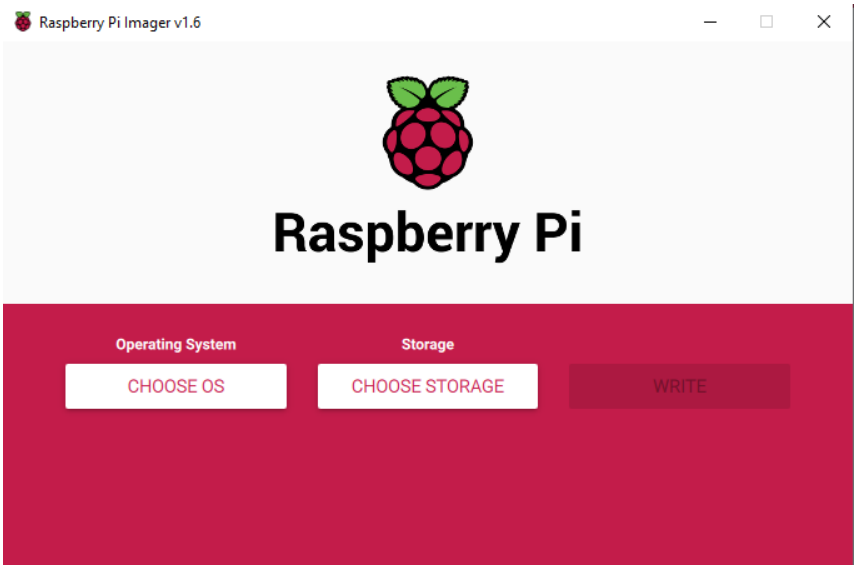


Figura 10: Raspberry Pi Imager v1.6.

3.1.1 Instalación de imagenes.

Para instalar la imagen, tenemos que descargar la última versión del software de la página Web del proveedor y utilizar la herramienta "*Pi Imager*" para

grabarla en la SD o usar la herramienta "*Pi Imager*" y seleccionar el sistema operativo Raspbian que viene incluido. Como podemos ver en la siguiente figura, tenemos bastantes opciones de descarga a parte de Raspbian.

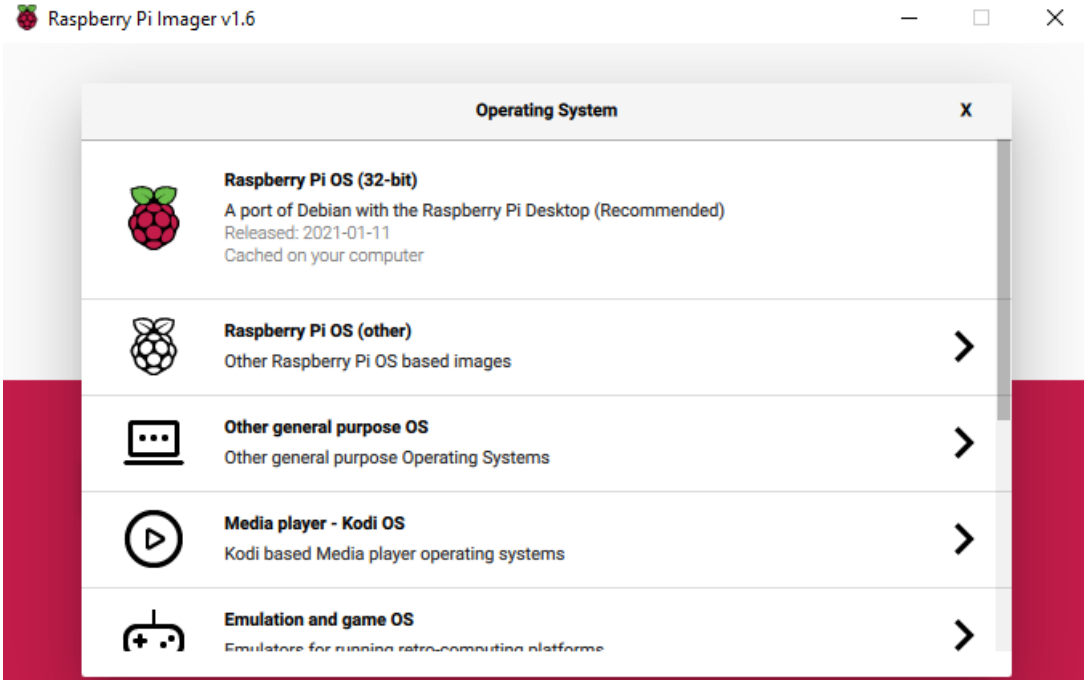


Figura 11: Menú aplicación Imager

Si queremos instalar otra imagen que no aparezca en la aplicación, vamos a la última opción "*Use Custom*" y seleccionamos el archivo .img que corresponda.

La imagen Raspbian, la instalaremos sin entorno gráfico en nuestra Raspberry Pi 4 de 2GB. El proceso lo podemos ver en el anexo B.

Este mismo proceso lo realizamos en las otras dos Raspberry. En la Raspberry Pi 4 de 8Gb instalaremos el Honeypot. Para ello, tenemos varias opciones, seleccionar una imagen de Honeypot ya preconfigurada de Internet y pasarla a nuestra tarjeta SD o podemos instalar una distribución de Raspbian e instalar el Honeypot y herramientas de monitorización (elegimos esta última).

Para la parte de Pentesting, nos descargaremos una imagen Kali Linux de la Web oficial de Offensive Security que dispone de imágenes para chips ARM específicamente compiladas para Raspberry Pi modelos 2, 3 y 4.

El proceso es el ya comentado excepto porque hay que elegir la opción "*Use Custom*" y seleccionar la imagen que corresponda. Para más detalles, ver Anexo B.

3.1.2 Configuración de imágenes.

La configuración de las imágenes consiste en unos primeros pasos iguales para todas las Raspberry y otros específicos para cada una en concreto. Los pasos comunes al Honeypot y Pentesting son los que podemos ver al final del anexo B. Instala y configura XRDP para acceder de forma remota e instala el software de grabación de pantalla simplescreenrecorder para poder documentar gráficamente todas las actuaciones. En el dispositivo Router, los pasos son diferentes al carecer de entorno gráfico para optimizar recursos.

Los pasos específicos se dividen por dispositivos. En el Router, tenemos que realizar cuatro pasos importantes: configurar nuestro punto de acceso, configurar nuestro servicio Tor, configurar nuestro modem 3G y por último configurar el enrutamiento en función de si salimos por 3G o por Ethernet (ver Anexo B). En el dispositivo, que hará de Honeypot, tendremos dos pasos, uno será instalar Dionaea, NTopng y Wireshark y el otro que será configurar nuestro router doméstico para poner la IP de la Raspberry en la DMZ y realizar port forwarding de los servicios publicados (ver Anexo D). Por último, en el dispositivo pentester, solo tendremos que instalar la imagen Kali, actualizarla e instalar todas las herramientas necesarias. Toda la batería de pruebas contra nuestro Honeypot y nuestra navegación a la red Tor será con dicha imagen (Ver Laboratorios).

Dispositivo punto de acceso + Tor + Módem 3G/Eth:

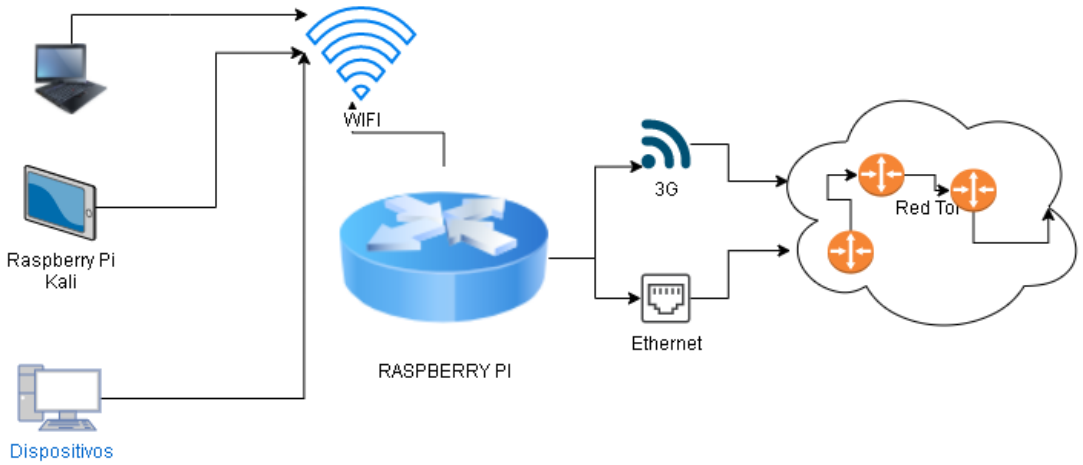


Figura 12: Esquema básico de red Router Raspator

En el dispositivo router utilizamos la antena wifi como Hotspot (punto de acceso wifi) para que los clientes se conecten a nuestra Raspberry Pi y se les asigne una IP por DHCP. Nuestro router utilizará la salida a Internet en función de lo que encuentre conectado, nuestro modem 3G o nuestra red Ethernet. Para más detalles de la instalación, ver anexos C, D y E.

Dispositivo Honeypot + Ntopng.

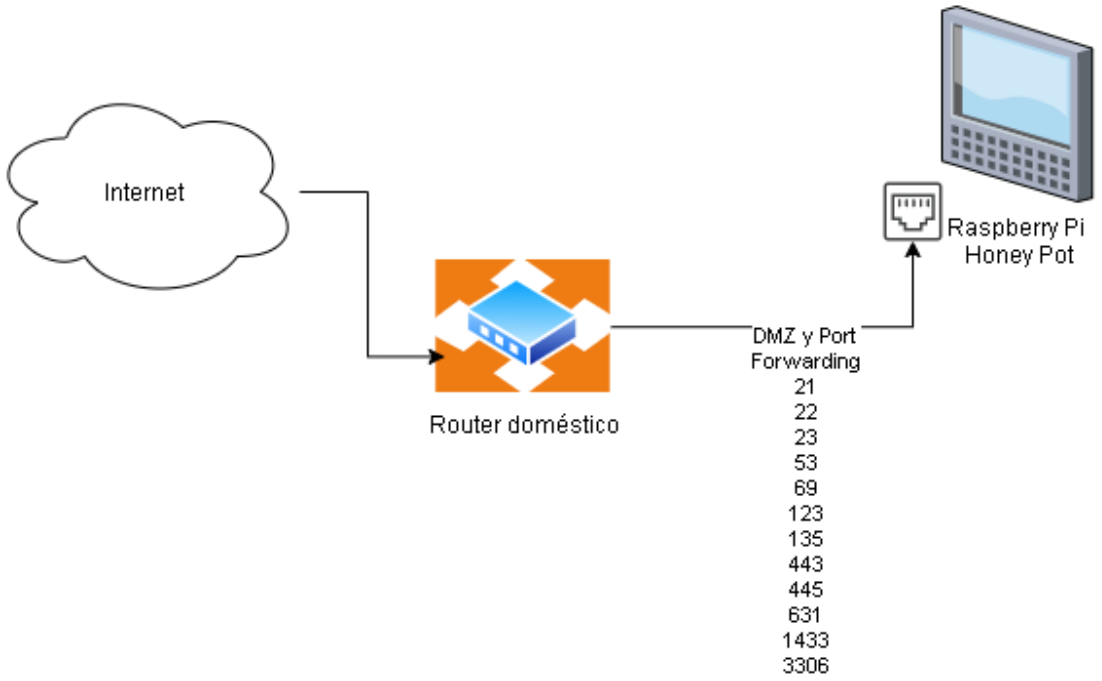


Figura 13: Esquema básico de red Honeypot

En el dispositivo Honeypot redireccionamos todo el tráfico de Internet de los puertos que emula con el servicio Honeypot de Dionaea para que le lleguen directos a la Raspberry Pi. Con el software NTopng monitorizamos las peticiones y su procedencia. Para ver más detalles de su instalación, ver Anexo D.

Dispositivo Pentesting con Kali:

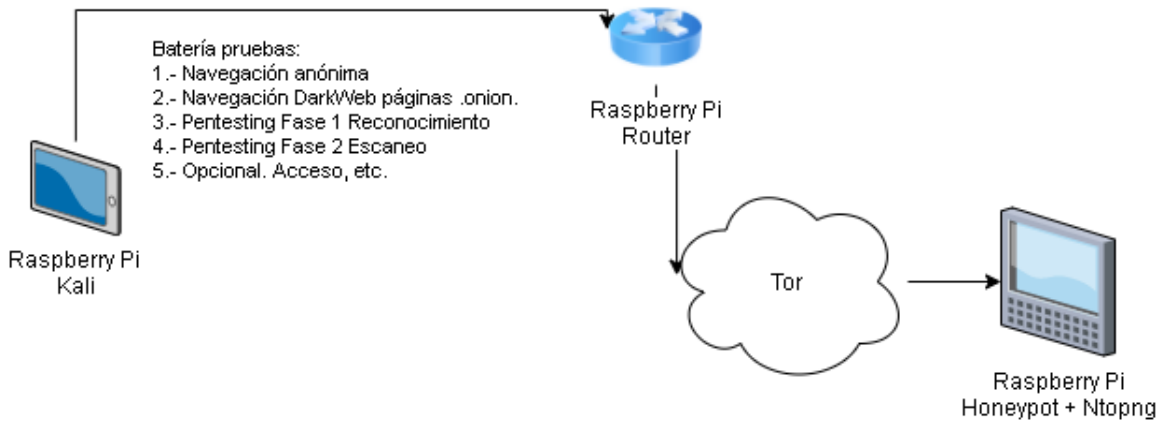


Figura 14: Esquema básico de red Pentester

Desde el dispositivo pentester o kali realizamos todos los laboratorios necesarios (ver Anexo 3). Los dos primeros laboratorios, son para dar a conocer la navegación anónima conseguida con el dispositivo Router (Out-proxy e In-proxy) y las demás son para analizar el anonimato de lanzar dos de las fases de un test de penetración contra el otro dispositivo Honeypot. Para este trabajo, en la parte de hacking ético, las interesantes son las tres primeras fases, Reconocimiento, Escaneo y Acceso. Reconocimiento no lo vamos a realizar porque ya conocemos la IP contra la que lanzar los escaneos y ataques. Opcionalmente podemos usar otras fases como mantener acceso, borrar huellas, etc, aunque no es el objetivo de este trabajo, esto es debido a que el Honeypot está actualizado y sin vulnerabilidades, pero se refleja cómo trabajo de investigación en líneas futuras. Ver líneas futuras.

3.2 Monitorización.

Los sistemas Honeypot y monitorización mediante NSM o SIEM, son herramientas que se utilizan en empresas e industria para aumentar su capacidad de proteger de ataques normales y ataques persistentes, detectar anomalías en la red, etc. También, sirven para alertar de dichos ataques persistentes y ser capaz de eliminarlos de forma segura, anticiparse a futuros ataques, desviar la atención de los atacantes y todo ello de la manera más optimizada y económica posible. Este tipo de herramientas es administrado por lo que se conoce como "*Blue Team*", y son un

grupo de personas con distintas habilidades y roles capaces de instalar, configurar, monitorizar y gestionar las herramientas nombradas.

En lo concerniente al Honeypot, este se utiliza sobre todo para anticiparse a futuros ataques y para desviar la atención de los atacantes. Se intenta que los atacantes se centren en los equipos configurados para ello en vez de la infraestructura real. Dionaea es un Honeypot de baja interacción lo que implica que, a menos que se modifique, sería detectado muy rápidamente. Para la parte de monitorización, existen herramientas gratuitas muy potentes capaces de detectar ataques de red, escaneos o incluso disponen de inteligencia artificial para correlacionar eventos de varios sistemas (SIEM) y detectar y avisar al "*Blue Team*". Aunque dichas herramientas cada vez son son más sofisticadas, siempre es necesaria la labor humana para su puesta en marcha, configuración, mantenimiento e interpretación. En el caso de NTopng, es una herramienta de monitorización capaz de detectar y alertar de ciertos ataques de red, pero en un entorno de producción se suelen usar otras más completas. Sin embargo, para este trabajo, tanto Dionaea como NTopng fueron las acordes para los dispositivos Raspberry.

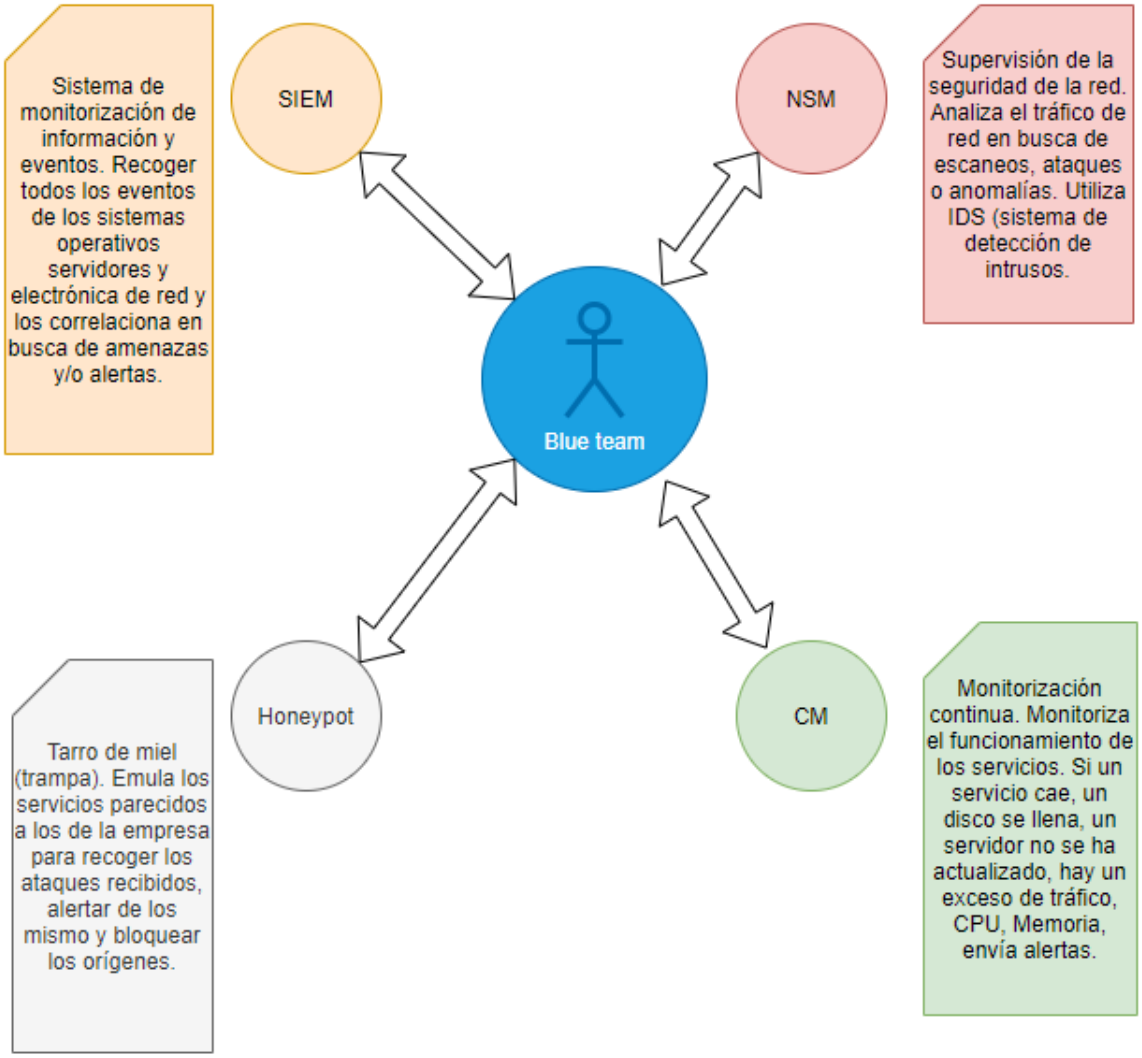


Figura 15: Esquema herramientas "*Blue Team*"

3.3 Pentesting.

Para el laboratorio de pruebas de escaneo y acceso, se ha utilizado el tipo de Pentesting de caja negra. Cuando nos referimos a hacking ético, hay varios tipos de auditorías que se dividen en tres categorías según los conocimientos que se tengan sobre la infraestructura o las acciones que el mismo cliente contrate.

Auditoría de caja blanca: El auditor o pentester conoce gran parte o toda la infraestructura de red y sistemas del cliente y en algunos casos tiene acceso a las aplicaciones y su código fuente. También, suele estar directamente conectado a la

parte de red interna como un usuario más para poder hacer la auditoría más en profundidad.

Auditoría de caja negra: El auditor solo conoce los datos que le de el cliente y normalmente son reducidos (simulando así lo que sería un ataque real). Suele ser el nombre de la empresa y puede que el del dominio. A partir de ahí, el trabajo del pentester es recoger toda la información posible y continuar con las fases del hacking ético comentadas en los siguientes párrafos.

Auditoría de caja gris: Esta auditoría es un punto intermedio entre las dos anteriores. El auditor conoce parte de la infraestructura de red o sistemas y algunos servicios, pero el resto tendrá que averiguarlo realizando las siguientes fases.

Vamos a enumerar las fases de la metodología aprendida durante el Máster universitario en Ciberseguridad y la del CEH, esta metodología divide el hacking ético en las siguientes fases:

- Fase 1 – Reconocimiento (Reconnaissance):

En esta fase de reconocimiento, se investiga al cliente (víctima) para ver qué tipo de información valiosa podemos obtener: usuarios, dominios, cuentas de correo e IPs. En resumen, obtener el máximo de información posible.

- Fase 2 – Escaneo (Scanning):

En esta fase, al disponer de distintos objetivos (IPs, Dominios, cuentas de correo, usuarios) se procede a realizar el escaneo. Se puede realizar un escaneo de IPs y de puertos con Nmap, se puede buscar información de los dominios y subdominios en Internet, páginas de archivo de antiguas versiones de sus páginas web, etc. También es posible enviar correos de ingeniería social a cuentas de correo para ver la predisposición de los usuarios a ser engañados con correos manipulados.

- Fase 3 – Obtener Acceso (Gaining Access):

En la fase de escaneo ya hemos obtenido información suficiente para saber que es vulnerable a uno o varios ataques, sino, habría que volver a la fase 1 y 2 hasta encontrarla. Como ya tenemos una o varias vulnerabilidades explotables, podemos probar dichas vulnerabilidades en un entorno de laboratorio para no dar la voz de alarma; si vemos que son explotables y no dañinas (no producen pérdida de

servicio); las lanzamos al cliente. Si obtenemos acceso o accesos, pasamos a la siguiente fase.

- Fase 4 – Mantener Acceso (Maintaining Access):

Entrar dentro de los sistemas no es el fin. En esta fase posiblemente haya que repetir las fases 1, 2 y 3 en busca de nuevos objetivos, pero antes, es necesario mantener el acceso para no perderlo. Para ello, debemos crear sesiones de Meterpreter o Shells reversibles, de tal manera que si se reinicia el dispositivo no perdamos la conexión y siempre esté disponible para nosotros (nosotros somos los auditores contratados, ya que si no fuera el caso, ya habríamos quebrantado varias leyes y seríamos juzgados por lo penal -cárcel-). Una vez hecha la parte de mantener el acceso, tenemos que hacer lo que se conoce como "*desplazamientos laterales*" o "*pivoting*", para ello, volvemos a las Fases 1, 2 y 3, pero esta vez dentro de la infraestructura de red del cliente, lo cual será más sencillo y nos dará más información, pero debemos tener cuidado de no hacer saltar las alarmas ni dejar los sistemas inestables.

- Fase 5 – Limpiar Huellas (Clearing Tracks):

Esta fase se realizaría si el Pentesting no es de tipo "*Hacking ético*", sino un Pentesting no autorizado. Al no estar autorizado para realizarlo, el atacante no quiere que se sepa que estuvo allí y borra ficheros logs allá donde estuvo y por donde pasó: sistemas operativos, firewalls o routers. En un hacking ético, esta fase sería sustituida por la de entrega de informe de vulnerabilidades. Son dos, uno técnico de las vulnerabilidades y de cómo solventarlas y otro menos técnico para los CEOs de la empresa.

El esquema seguido para el escaneo de puertos y búsqueda de vulnerabilidades sería el siguiente:

Como ya conocemos la IP (fase de reconocimiento), pasamos a la fase de escaneo y enumeración para ver los servicios disponibles e intentar averiguar la máxima información de los mismos. Por ejemplo: dispone del puerto 443 abierto, en ese caso, lanzamos un script de Nmap para averiguar más información sobre dicho puerto; dispone del puerto 80 y el servicio está proporcionado por Apache, IIS, etc, realizamos una búsqueda de la vulnerabilidad de XSS, o de si se puede aplicar SQL Injection, etc.

Cuando tenemos la máxima cantidad de información posible, podemos pasar a la fase de acceso o ataque. Para ello, si hemos detectado que es factible aplicar Cross Site Scripting (XSS), preparamos un exploit para intentar obtener una Shell (Consola o terminal a la máquina víctima). Si por el contrario vemos que utiliza una versión de Apache antigua que dispone de un exploit capaz de generar una Shell, intentaremos subir un fichero de tipo "*phpshell"* al propio servidor Apache, o aprovecharemos cualquier otra vulnerabilidad. Cada servicio encontrado es un punto de ataque, pero solo se aprovecharán los que veamos que son factibles.

Si se llega a tener éxito con cualquier de los probados, el siguiente paso es mantener acceso. Es importante mantener acceso debido a que muchas de estas vulnerabilidades vuelven el sistema inestable y este puede perder conexión, reiniciarse o que lo reinicie algún administrador. Mantener acceso es la siguiente fase y consiste en migrar el proceso a otro más estable que se esté ejecutando en la máquina, instalar el acceso como servicio de tal manera que se ejecute al arrancar, intentar deshabilitar las herramientas de seguridad que se encuentren en dicha máquina, antivirus, firewall, antispyware, UAC, etc, crear reglas en el router o instalar puertas traseras.

La última fase "*limpiar huellas*", se suele usar con ataques persistentes o para ataques que no se quieran descubrir por la víctima en un corto espacio de tiempo. Se ha descubierto que los hackers que realizan espionaje industrial o cualquier otro tipo de hacking no ético prefieren los ataques persistentes, esto se debe a que dan más beneficios, tanto económicos como dañinos. Esta fase de limpiar huellas no se suele hacer cuando se contrata una empresa de seguridad para hacer una auditoría de pentesting o hacking ético, en vez de ello lo que se entrega son dos informes. Uno con las vulnerabilidades encontradas, las explotadas, como remediarlas, como se tuvo acceso y desde donde. Este informe es más técnico y sobretodo orientado a los administradores de red de la empresa. Por otra parte, se entrega otro informe más formal y menos técnico, orientado a los directores o responsables de la compañía. Los dos son igual de importantes y complementarios. Con el primero se consigue solucionar el problema y con el segundo se consiguen los recursos y la concienciación necesaria por parte de la empresa.

Como ya comentamos, para este trabajo, solo realizamos la metodología del escaneo y enumeración y acceso para generar la máxima cantidad posible de

paquetes de red en busca de algún indicio de nuestra IP original, no la enmascarada por Tor.

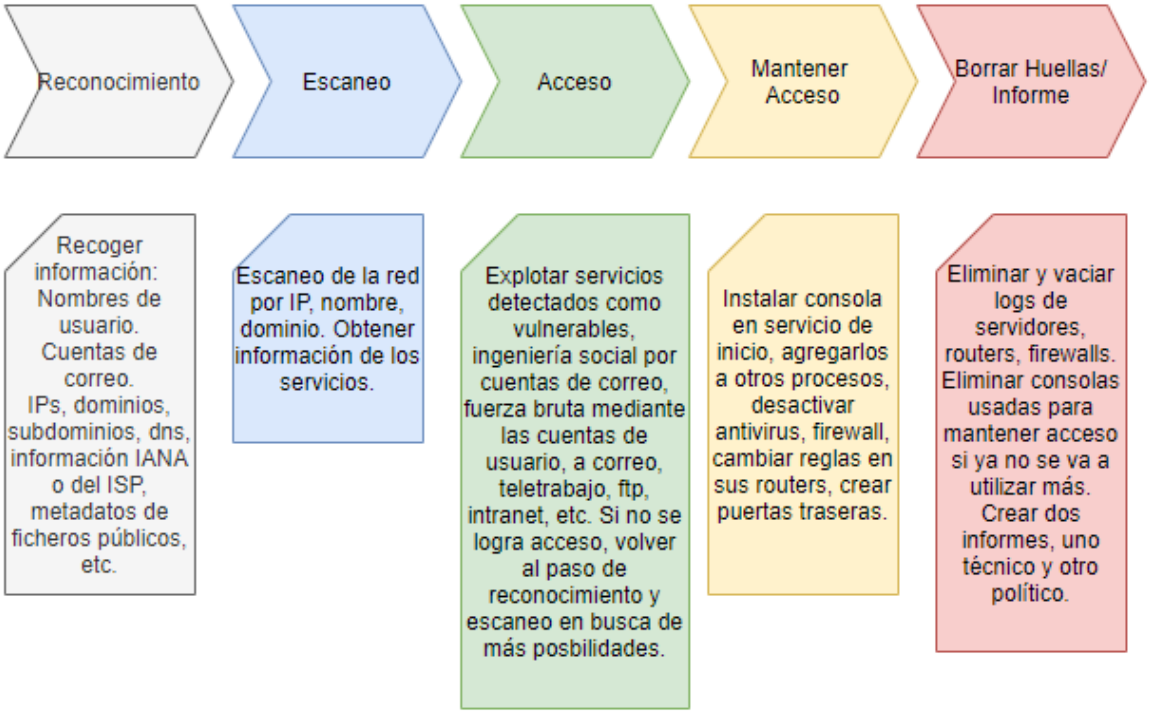


Figura 16: Esquema hacking ético

4. Navegación en TOR

4.1 Navegación anónima con Tor.

Como ya hemos comentado en apartados anteriores, tenemos multitud de opciones de navegar de forma anónima. En el caso de Tor, debemos descargar el navegador Tor Browser para nuestro Windows, Linux, Mac o Android. Una vez descargado, lo abrimos y ya estará listo para navegar de forma casi anónima. Si no se usan herramientas como TorghostNG o las comentadas, ciertos servicios tienen que ir a través de nuestra salida a Internet sin cifrar y sin pasar por Tor. También, hay cookies necesarias para que funcionen ciertas páginas y otros parámetros que no podremos anonimizar. Si usamos nuestro router, no será necesario Tor Browser, simplemente nos conectaremos al Hotspot y ya podremos navegar de forma anónima.

Nota importante: Un reciente estudio indica que más del 25% de los nodos de salida Tor espiaron a los usuarios (J. Menéndez y Una al Día. Hispasec). Esto significa que hay que tener mucho cuidado con la información que se envía a través de esta red y hay que intentar que siempre vaya cifrada.

4.2 Navegación Out-proxy.

Resumamos el proceso de conexión Tor "*Out-proxy*" para navegar de forma anónima por la Surface Web. Lo que mostramos es lo que ocurre cuando ejecutamos el servicio de Tor o abrimos el navegador Tor Browser.

Todo el proceso es transparente al usuario:

Paso 1: Nos conectamos al nodo autoridad y descargamos la lista de nodos disponibles por los que podemos pasar.

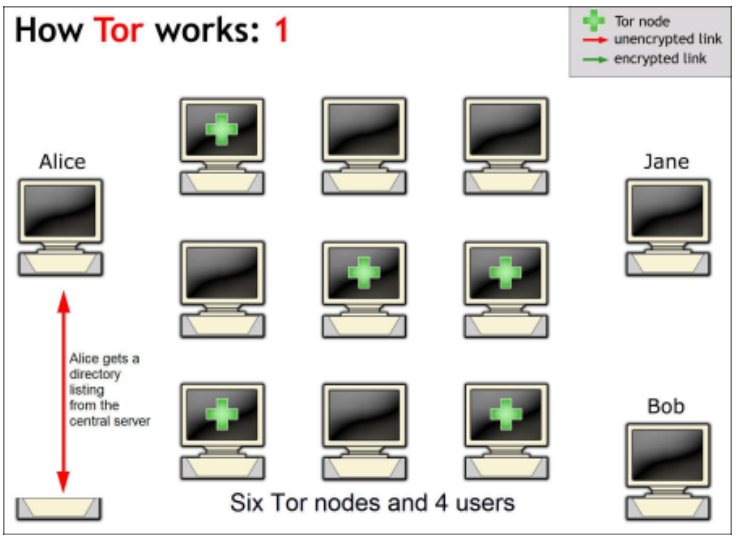


Figura 17: Cómo trabaja Tor 1. (Tor Project)

Paso 2: A menos que se especifique en el archivo de configuración torrc, se eligen tres nodos al azar, uno de entrada, uno medio y otro de salida (para ser entrada o salida hay que cumplir unos requisitos y no vale cualquiera, tienen que estar declarados como tal por Tor Project). Nos descargamos las claves públicas de los tres nodos y el mensaje en claro se cifra con el nodo de salida, el resultado se cifra con el nodo medio y el resultado con el nodo de entrada, de esta manera, solo el nodo de salida verá lo que hay dentro cuando se lo envíe el nodo medio, el nodo medio verá los datos cifrados y el nodo de entrada podrá ver la cabecera del origen, la IP real del emisor, pero no el contenido del mensaje, ya que también estará cifrado.

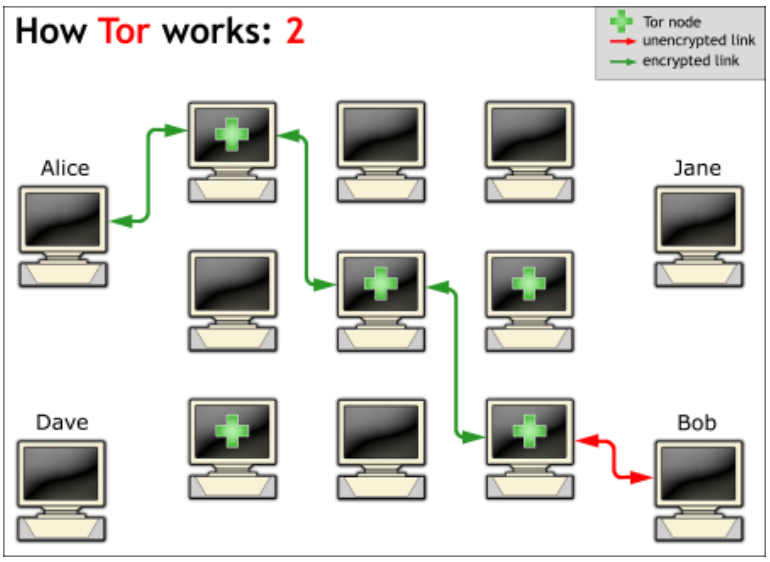


Figura 18: Cómo trabaja Tor 2. (Tor Project)

Por defecto, pasados 10 minutos, ese proceso se vuelve a repetir eligiendo diferentes nodos y creando otro nuevo circuito.

La siguiente figura muestra los pasos de cifrado por capas. Como el router de entrada tiene su clave privada, podrá retirar esa capa y pasárselo al nodo medio, como el nodo medio tiene la clave privada de la siguiente capa podrá quitarla y pasársela al siguiente nodo, por último, el nodo de salida tiene su clave privada y podrá retirarla y enviar el mensaje sin cifrar a la IP destino.

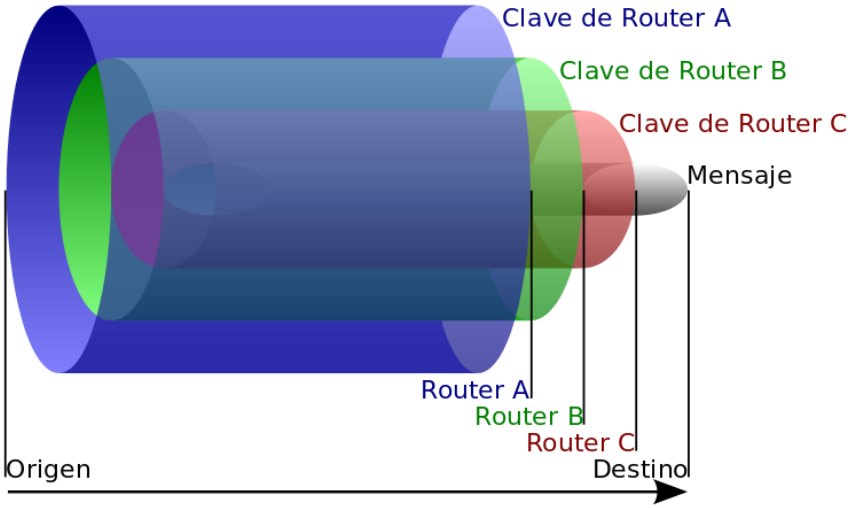


Figura 19: Cifrado de la red Tor. (Tor Project)

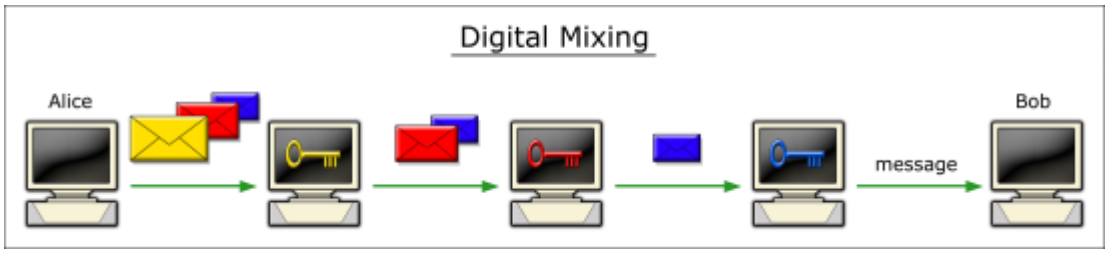


Figura 20: Cifrado de la red Tor. (Tor Project)

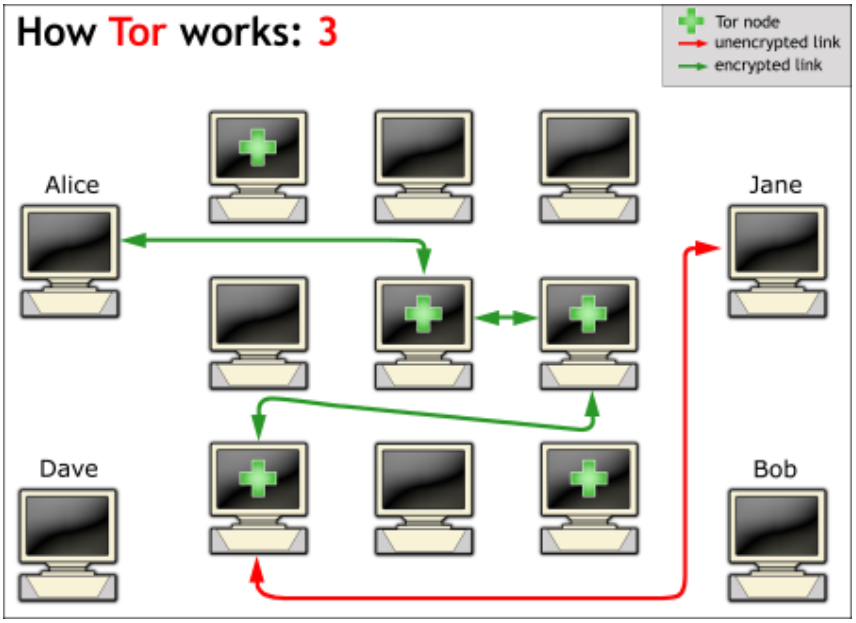


Figura 21: Cómo trabaja Tor 3. (Tor Project)

Antes de continuar, debemos comentar que este protocolo se puede modificar mediante el fichero de configuración torrc en el que podemos añadir/retirar configuración, seleccionar los nodos bridge, de entrada, medio o salida. También el país de ubicación, compartir y publicar puertos y múltiples opciones más. También, el propio Tor incluye algunos comandos de resolución DNS, ver el estado de la red, etc. Para ver el tipo de cifrado, ir al Anexo E: Cifrado conexión OP contra OR. Para ver opciones de configuración, ir al Anexo F: Archivo de configuración Torrc.

4.3 Navegación In-proxy de Tor.

El funcionamiento de la red Tor In-proxy es diferente a Out-proxy. Los dominios .onion no se resuelven con los DNS de Internet, sino que lo hacen de una manera diferente. En la red Tor, para poder navegar In-proxy, es necesario que haya publicados servicios Web. Dichos servicios se publican mediante lo que se conoce como Hidden services.

Si se quiere publicar un servicio web como Hidden service, el paso inicial de la instalación en el servidor es igual a instalar un IIS o un Apache, lo que cambia es cómo se accede al servicio y la configuración en la carpeta Tor. En el archivo torrc hay que añadir las dos siguientes líneas, la primera es donde se almacenará la

configuración del servicio, la segunda es el puerto externo que se compartirá y el interno al que se redirigirá.

HiddenServiceDir /Library/Tor/var/lib/tor/hidden_service/

HiddenServicePort 80 127.0.0.1:8080

Dentro de la carpeta "*hidden_service*", encontramos dos archivos de configuración, "*private_key*" en el que estará la clave privada y "*hostname*" en el que se encontrará la dirección .onion de nuestro servicio web. La dirección .onion, se genera a partir de pasar la función hash Sha1 a la clave pública generada a partir de la clave privada, se toma la primera mitad y se codifica en Base32. De esta manera, la dirección .onion ocupará 16 caracteres usando de la a-z y del 2-7. (Castillo) (Tor Project Inc).

Hasta ahora, esta parte es la que describe cómo compartir un hidden service (una página web .onion, por ejemplo) en la red Tor. Pero ahora vamos a hablar del funcionamiento y de cómo es capaz de comunicarse el cliente con el servidor sin saber dónde se encuentra cada uno.

Hemos hablado de los nodos "*relé*" que son de entrada o nodo guarda, el nodo medio o nodo middle y el nodo de salida, imprescindibles para lograr el anonimato en la navegación a través de Tor. También existe otro nodo llamado nodo Bridge el cual no muestra su IP en el listado público de Tor y que se usa para poder acceder a Tor desde países que tengan restricciones o bloqueos a esos nodos. Al no estar publicada la IP del nodo bridge es más difícil bloquear dicho acceso. Existen varias maneras de encontrar un nodo bridge: mediante lista de correo, con un buscador o con una cuenta de Twitter.

Ahora debemos añadir cuatro tipos de nodo más. Algunos de ellos pueden estar siendo otro tipo de nodo para otros circuitos al mismo tiempo. Los **nodos de directorio** (son 9 en total), el nodo que hará de **punto de introducción**, el nodo que hará de **base de datos** de los hidden services (los servicios ocultos .onion) y los **nodos de encuentro** o Rendezvous point.

Nodos de directorio: A partir de 2019, existen 9 nodos de directorio. Cada nodo está controlado por una organización diferente y cada hora deben llegar a un consenso para establecer el listado de nodos "*relé*". Los nodos de directorio contienen la información de los **nodos "*relé*" listados públicamente en la red Tor**.

Los nodos directorio son esenciales para todo el funcionamiento de la red, especialmente para los servicios ocultos comentados anteriormente (hidden services), también son los que deciden quienes son los nodos de entrada y de salida. Junto con los nodos directorio se encuentran otros nodos conocidos como **nodos caché** que ayudan en la labor de distribuir el listado de nodos, de esta manera no se ven desbordados.

Nodo punto de introducción: Estos nodos son elegidos al azar y consultados primero por el servidor que tiene el hidden service para ver si están disponibles para realizar esa labor. En este tipo de nodos se almacena la descripción del servicio oculto junto con la clave pública y el listado de IPs de los otros puntos de introducción que lo albergan. Esa información se envía junto con la dirección .onion al nodo base de datos.

Nodo base de datos: Estos nodos almacenan una base de datos de hashes distribuida. De tal manera que cuando nos conectamos a Tor, el nodo directorio le envía al cliente conectado la dirección de uno o varios de estos nodos para que pueda consultar los servicios .onion y este nodo le pase el listado de los puntos de introducción, ya que como hemos comentado, el cliente no conocerá la IP del servidor y viceversa.

Nodo de encuentro: El nodo de encuentro es el encargado de establecer la conexión entre el cliente y el servidor. Es un nodo neutral interpuesto entre los dos nodos (el cliente y el servidor) de tal manera que no se conozcan. A parte de eso, el nodo de encuentro tiene un circuito contra el cliente y otro circuito Tor contra el servidor.

En el apartado anterior Navegación Out-proxy, hemos visto a grandes rasgos cómo se realiza la conexión a la red Tor. Ahora vamos a resumir cómo funciona la conexión entre un dispositivo conectado a la red Tor y un servidor que disponga de un hidden service.

El servidor realiza una petición a varios puntos de introducción preguntándoles si quieren serlo, si están de acuerdo, se les envía alguna información para que puedan reconocerlo (clave pública, descripción y lista de puntos de introducción donde se encuentra). Después se carga el hash de ese dominio .onion junto con las IPs de los puntos de introducción en el nodo que

contenga la base de datos. En ese momento, ya está publicado el hidden service de ese servidor. Sería lo equivalente a publicar un registro DNS a nivel global de un dominio.

El cliente ya está en la red Tor, entra en Surface Internet, abre un buscador y encuentra una web .onion a la que quiere acceder. Al abrir este dominio .onion, se envía una petición de dicho dominio a cualquier nodo que tenga la base de datos distribuida de los hashes. Este nodo le devolverá una lista con varios puntos de introducción. El cliente accede a uno de ellos y le dice que quiere verse con ese dominio .onion en un punto de encuentro elegido también al azar. El servidor recibe la petición y la acepta o la descarta. El cliente le dice al punto de encuentro que envíe al punto de introducción una clave de un solo uso y su dirección IP para organizar la cita. El punto de introducción envía dicha petición al servidor y este decide si aceptarla o no. Si la acepta, a partir de ahora, el cliente y el servidor se comunicarán mediante el punto de encuentro. En este apartado, vemos que el cliente tiene un circuito Tor y el servidor otro, lo que garantiza la privacidad de ambos. (Julian) (Fuentes Iglesias)

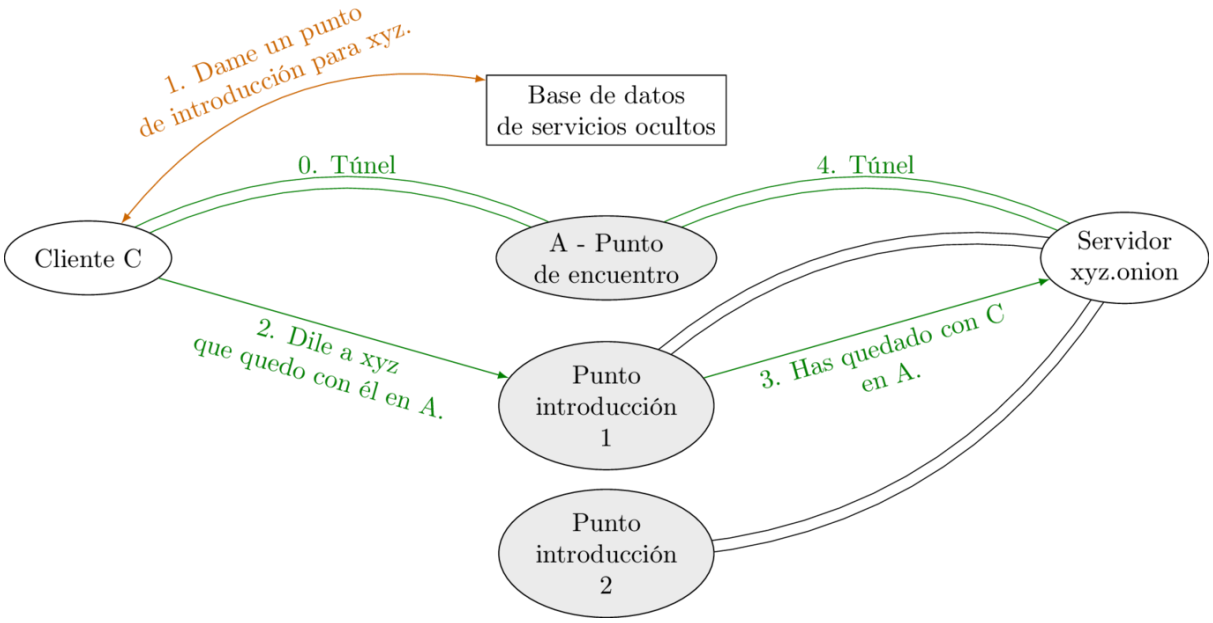


Figura 22: Resumen navegación In-proxy (Julian)

Más adelante, hablaremos del tema legal, pero adelantamos que, aunque usar Tor no es ilegal, sí que lo es cierto comportamiento que hagamos del mismo, con lo que hay que tener cuidado sobre el contenido al que accedemos. Se

recomienda no entrar en páginas de sexo, drogas, armas o terrorismo. Sin saberlo, puede ser que estemos incumpliendo alguna ley.

Para poder usar este tipo de navegación, necesitamos una página de índices .onion a la cual recurrir para encontrar lo que buscamos. La siguiente Web es una página de la Surface Web donde podemos encontrar varios enlaces .onion. Como podemos comprobar, los post más recientes son del 2017. También detectamos que hay algunas páginas obsoletas y otros enlaces no funcionan.

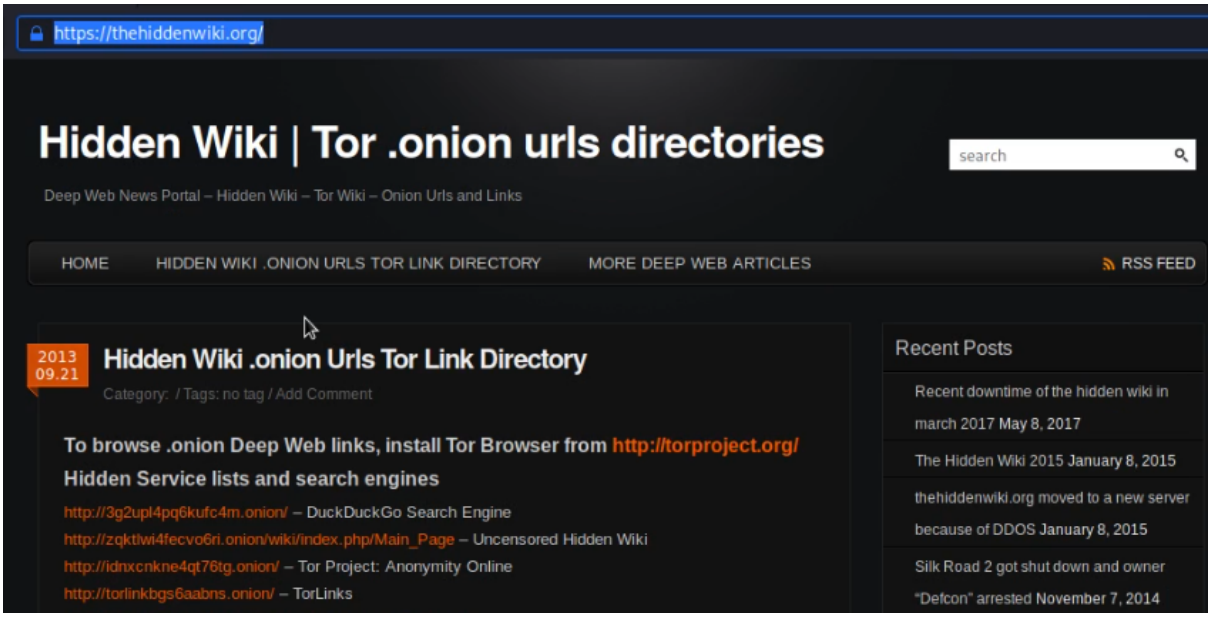


Figura 23: Página de índices .onion.

Una vez encontrada una página de índices que nos sirva, podemos abrir el contenido, por ejemplo, abrimos una web de índices muy conocida en la red Tor, "*The Hidden Wiki*". También hay dos buscadores muy conocidos llamados "*Not evil*" y "*Torch*", los cuales permiten buscar por contenido.

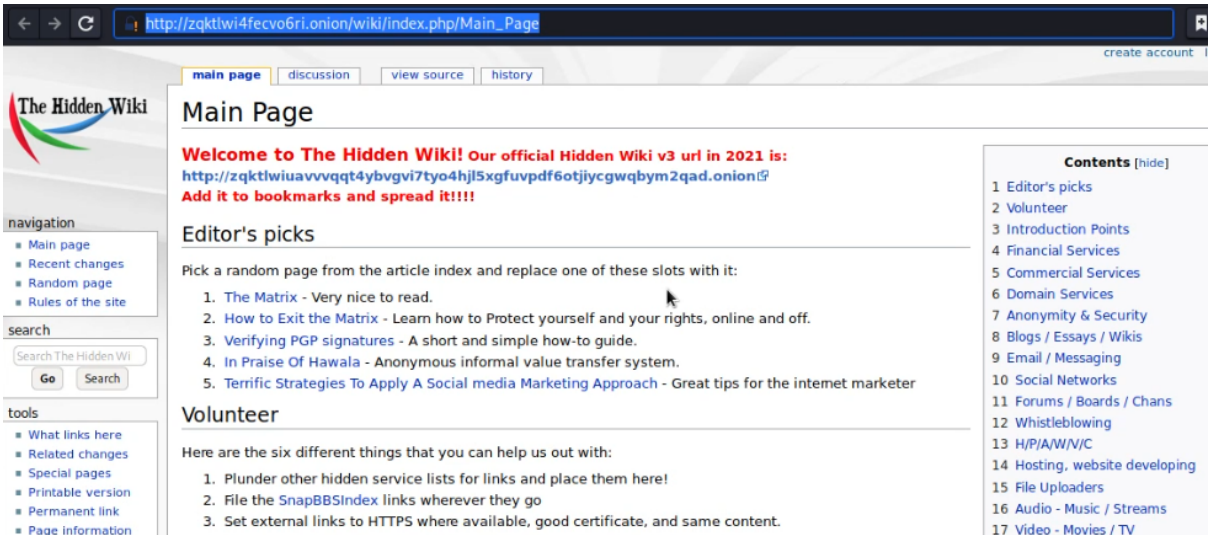


Figura 24: Índices páginas .onion red Tor

Algunas páginas web son fraudulentas o se usan para otros fines poco éticos como la venta de objetos robados o venta de armas ilegales.

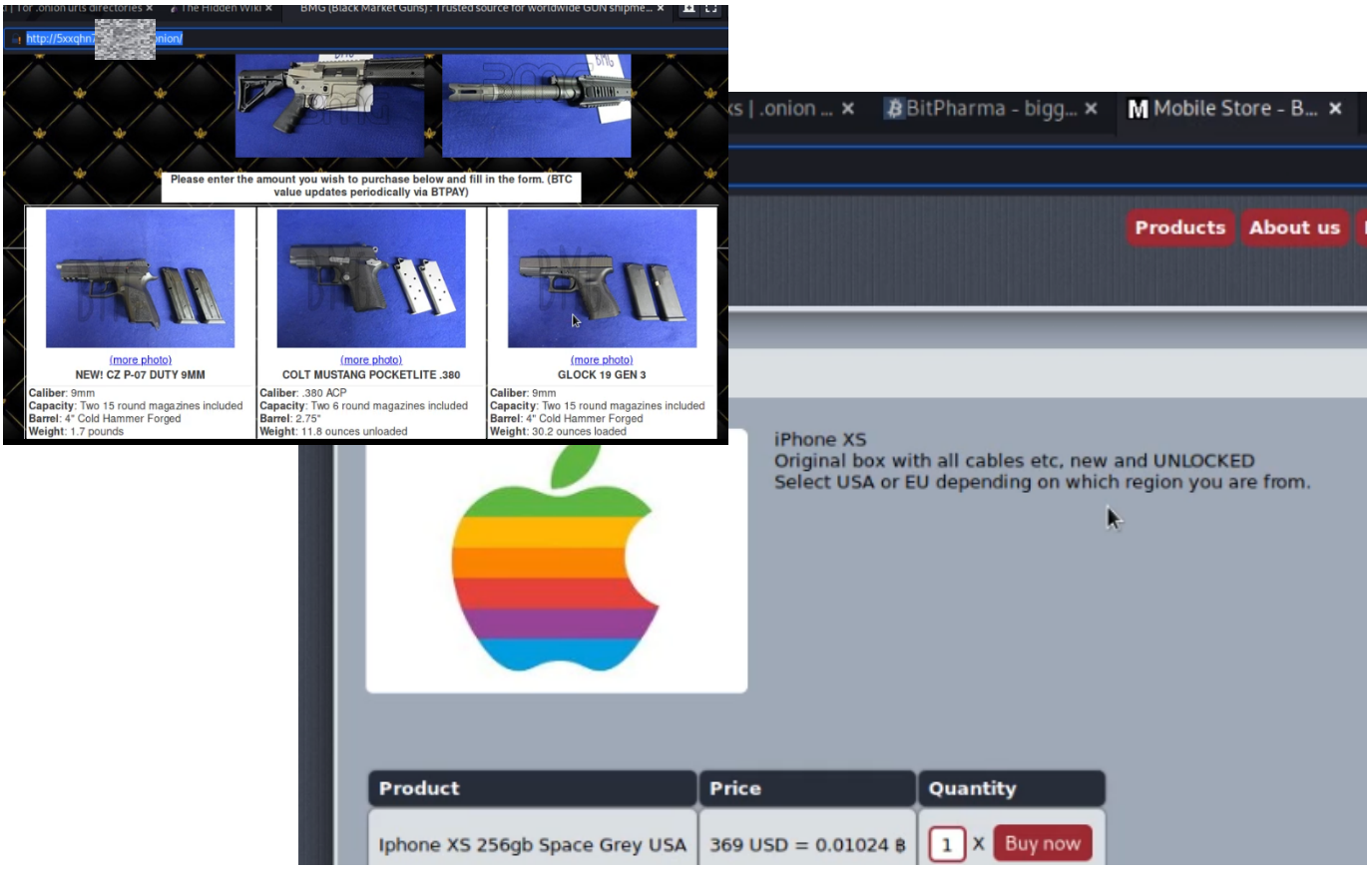


Figura 25: Venta de móviles y armas en la red Tor

Si nos conectamos con el móvil a nuestro Router, no necesitamos instalar Tor Browser para poder navegar en la Dark Web. En las siguientes capturas de móvil, podemos ver cómo estamos conectados a la Red Tor, como accedemos a una página de índices .onion y finalmente encontramos una Web de venta de servicios hacking.

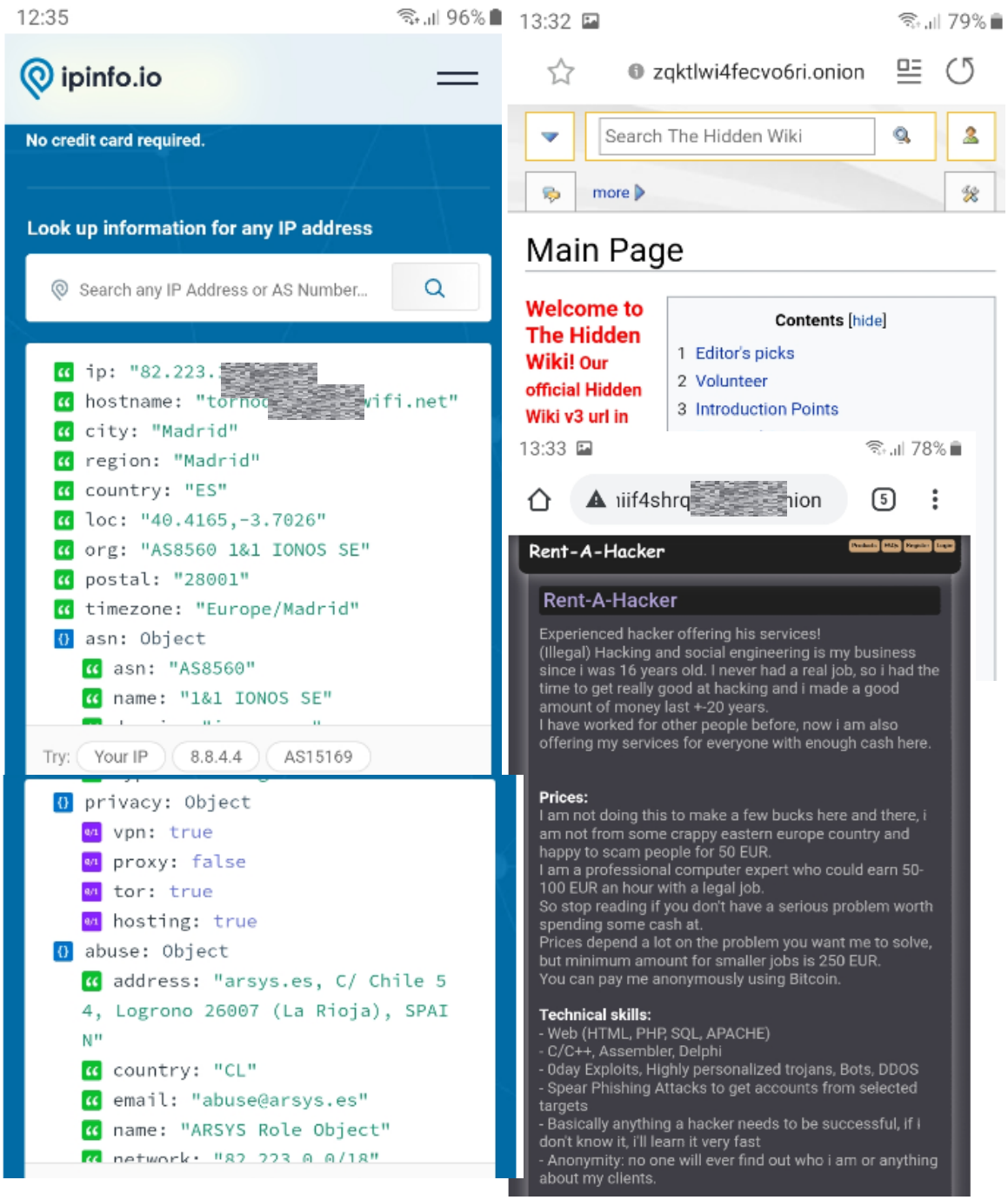


Figura 26: Captura móvil acceso red Tor

4.4 La ley con respecto a Tor.

Como hemos comentado, el uso de Tor no es ilegal en España, sin embargo, cierto contenido si lo es solo por hecho de tenerlo almacenado en nuestro PC o haber accedido al mismo. Este contenido es de carácter pedófilo y terrorista y puede ser juzgado por lo penal. Otros contenidos también son ilegales como drogas, armas, etc. Puede que no estén penados por el mero hecho de acceder a verlos en Tor, pero sí de tenencia si se adquieren, como pueden ser drogas, armas o software malicioso. En la mayoría de los casos, como mínimo, pueden ser acusados de delito menor con sanciones administrativas, aunque eso depende del uso que se les de o de la cantidad adquirida. Otros delitos como la intrusión (hacking) o delitos contra la propiedad intelectual también va por lo penal. (CJ Click Jurídico.) (Gil Gil y Hernández Berlinches 178-179)

En la siguiente tabla podemos ver la relación de castigos penales en relación al delito cometido si se usa Tor para fines poco éticos:

Descripción del delito	Artículo	Castigo
Intrusismo informático	Artículo 197 bis del 2015	Dos años o multa de 18 meses
Delito contra la propiedad intelectual	Artículo 270.2 del 2015	Pena de prisión de seis meses a cuatro años y multa de doce a veinticuatro meses
Capacitarse en actos terrorista	Artículo 575	De dos a cinco años
Material pedófilo	Artículo 189.1	De uno a cinco años

Tabla 1: Posibles delitos y su castigo (Gil Gil y Hernández Berlinches 178-179)

En la siguiente figura podemos ver que con algo de navegación ya se accede a páginas que pueden generar un problema legal como son venta de drogas, armas o alquiler de los servicios de un hacker.

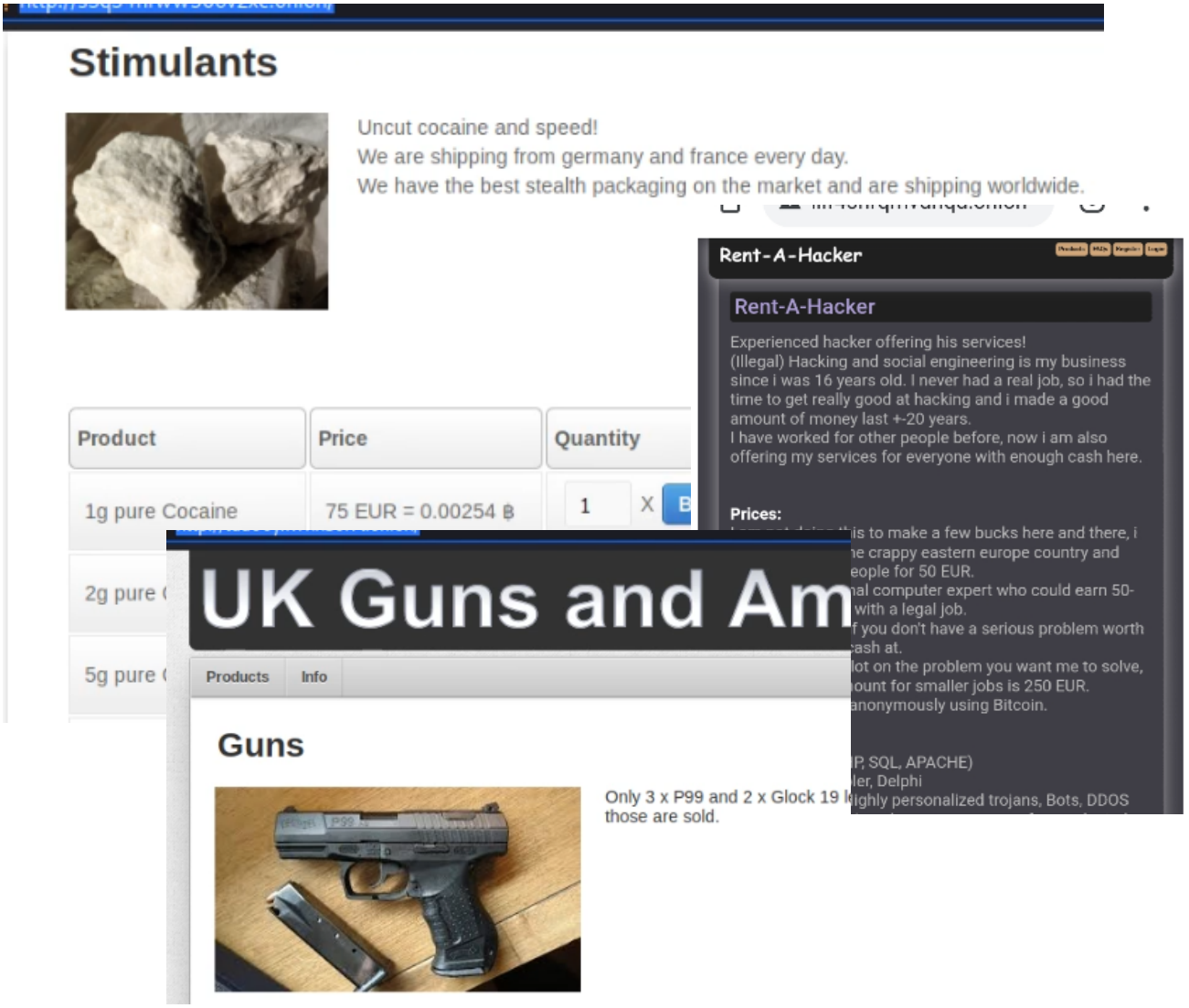


Figura 27: Páginas .onion de dudosa legalidad en Tor

5. Laboratorios

5.1 Acceso anónimo a la Surface web.

El primer laboratorio que realizamos al completar la instalación de nuestro router Tor es acceder de forma anónima (Out-proxy). Para ello, simplemente nos conectamos al punto wifi "*Raspator*" y la conexión a Internet automáticamente irá a través de la red Tor.

Nota: Raspator fue el nombre que se nos ocurrió durante la práctica, pero debemos reseñar que, durante la fase de investigación, se encontró un proyecto en GitHub similar que realiza algo parecido a crear un router con conexión a la red Tor, aunque el proceso es distinto (automatizado mediante script) y no contempla ciertas características con el objetivo de este TFM como el proceso de automatización y conexión mediante 3G. (Fajardo y Jayfajardo)

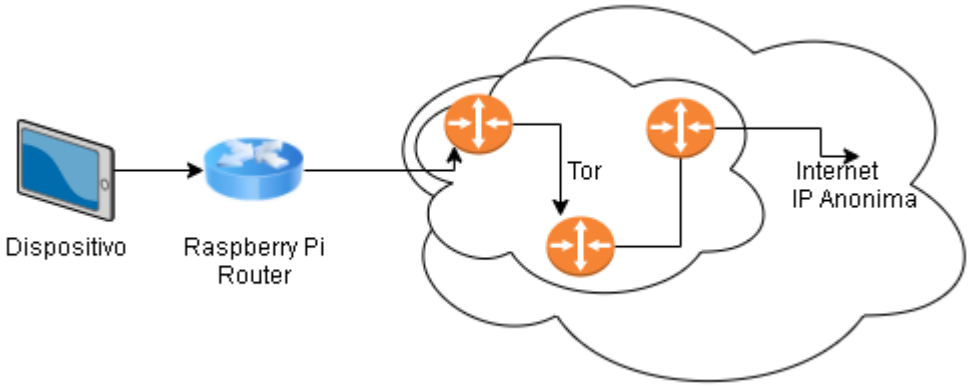


Figura 28: Esquema básico de red acceso Surface web con Tor

Comprobamos nuestro anonimato entrando en los enlaces https://hidemy.name/es/anonymity-cheker y https://proxy-cheker.net/es/privacy. Allí veremos que, efectivamente, nuestra conexión viene de un proxy Tor.

En la web de hidemy nos identifica que estamos usando Tor. Esto se debe a que dispone del listado de IPs de los nodos de Tor.

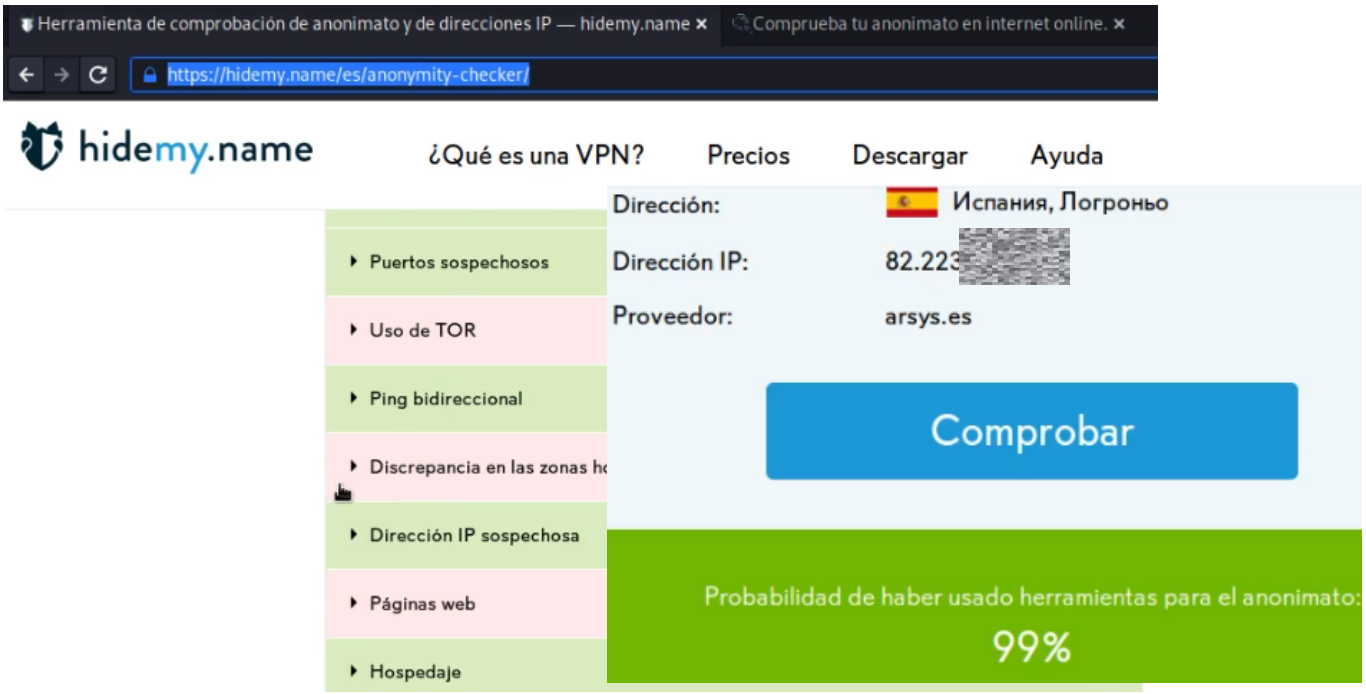


Figura 29: IP nodo salida y resultado de anonimato. Uso de Tor

5.2 Acceso y navegación en la Dark web.

En esta ocasión, el acceso a la red será dentro de la red Tor usando los enlaces .onion (In-proxy). Para ello, accedemos a una página de índices. Por ejemplo, la ya comentada "*The hidden Wiki*". La diferencia con respecto a navegar de forma normal, es que ahora estamos navegando dentro de la red Tor con sus dominios .onion, dominios que solo son posibles de resolver si estás usando el protocolo Tor.

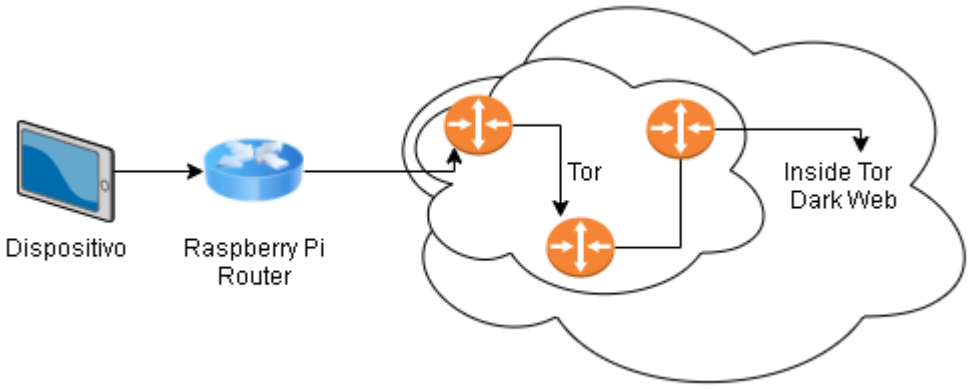


Figura 30: Esquema básico de red acceso red Tor

En la siguiente figura podemos ver una web .onion de enlaces a otras web Tor de tipo comercial y no comercial:

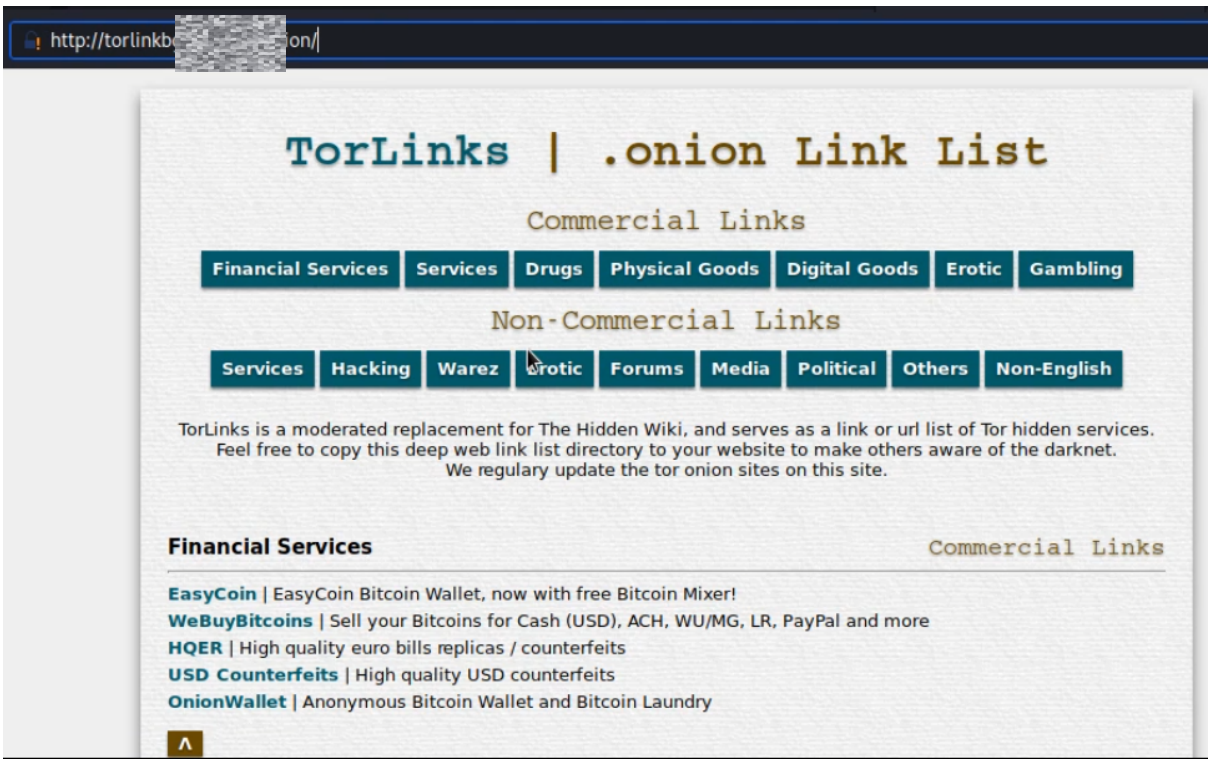


Figura 31: Ejemplo de navegación dentro de Tor

5.3 Hacking ético: Test de penetración.

Una vez completado con éxito los laboratorios anteriores, realizamos una serie de test de penetración para enumerar los servicios descubiertos en el Honeypot previamente configurado.

Existen varios tipos de escaneos de enumeración. Unos más agresivos que otros y que van a dejar más o menos huella. Siempre que realizamos un escaneo con una herramienta automatizada, quedarán logs de dicho escaneo. Si el escaneo de puertos o servicios es muy afinado, quedarán menos o incluso se entremezclarán con el tráfico legítimo. Si por el contrario se escanean puertos e IPs de manera continua, sin pausas y utilizando todo el ancho de banda y recursos, una gran cantidad de filas de nuestra intervención aparecerán en los logs del servidor/router contra el que lo lancemos, incluso hará saltar las alarmas de los IDS/IPS del cliente.

En este trabajo se intenta precisamente dar la voz de alarma y que nuestro escaneo aparezca en nuestro Honeypot de tal manera que podamos identificar la IP de origen con respecto de los demás ataques que nos realicen.

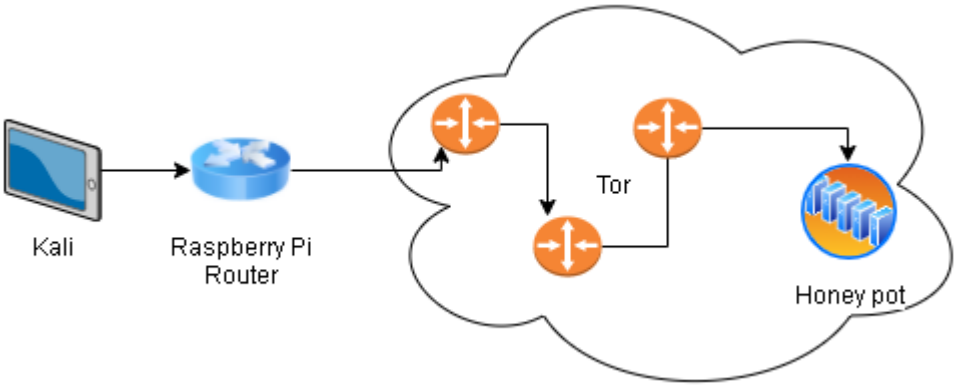


Figura 32: Esquema básico de red recorrido por el test de penetración

Como hemos comentado, la fase de un test de penetración que nos interesa para este trabajo es la de escaneo, dicha fase viene relacionada con la fase anterior de reconocimiento en la que habríamos averiguado la IP a la que pasar los escaneos. En la fase de escaneo, también hay que seguir un orden para no realizar pérdida o degradación de servicio. Para afinar más la búsqueda de vulnerabilidades, nos centraremos en los servicios que son más explotables (aunque cualquiera podría serlo si es vulnerable a un exploit): servicios web, servicios SMB, etc. Por eso usaremos un primer escaneo con Nmap, luego usaremos herramientas de búsqueda de vulnerabilidades de servicios Web como NIKTO, RAPIDSCAN y posteriormente un escaneo SMB con scripts de NMAP, etc.

5.3.1 Escaneo con Nmap.

La ejecución de esta herramienta contra servidores no autorizados puede conllevar penas de cárcel o sentencias administrativas.

Nmap: (Insecure.Com LLC) (Byte Mind) Es una herramienta de auditoría de seguridad y de descubrimiento de red. También se usa para monitorizar e inventariar la red. No solo permite escanear la red en busca de puertos abiertos, sino que dispone de múltiples tipos de escaneo de tal manera que es capaz de averiguar el sistema operativo, detectar filtrado de puertos (presencia de firewalls), utilizar una

máquina zombi, modificar el tipo de escaneo para evadir firewalls, usar señuelos, scripts preparados para servicios y un largo etc.

Los comandos y resultados son muy extensos como para añadirlos incluso como anexo, pero hemos añadido unas capturas de pantalla de resultados significativos de los comandos (Byte Mind).

Los dos primeros escaneos que lanzaremos conllevan considerablemente tiempo y dejan mucha huella, pero suelen dar buenos resultados. Recordemos que Tor no permite usar el protocolo UDP por eso solo hablaremos de escaneos TCP. Se trata de "*connect* scan" y "*syn scan*". **Connect scan** abre una conexión contra el puerto y la cierra pasando por todo el proceso de conexión de TCP (three-way handshake), sin embargo, **syn scan,** abre la conexión, pero la deja abierta con lo que solo realiza parte de dicho proceso TCP. *"Syn Scan"* y "*Connect Scan"* dieron resultados similares (excepto el FTP, solo mostrado con connect Scan). Para ejecutarlos solo hay que cambiar el flag -sS y -sT respectivamente. -A para que lance scripts y -p para indicar los puertos.

El escaneo de puertos muestra todos abiertos, sin embargo, esto es debido a que el Honeypot estaba en la DMZ y aparecen muchos filtrados por el propio firewall del router doméstico. La lista de servicios abiertos vulnerables es la siguiente: 21, 22, 23, 53, 69, 80, 123, 135, 443, 445, 631, 1433, 3306, aunque en la siguiente tabla solo mostramos algunos de ellos y las líneas de ataque mediante las cuales sería posible el acceso.

Servicio	Línea de ataque	Puerto
MYSQL	Enumeración de usuarios mediante fuerza bruta. netadmin:netadmin, etc.	1443
SMB	Puertos abiertos con carpetas compartidas. ADMIN$, C$, etc.	445
HTTP	Cross site scripting. XSS	80
HTTP	Apache 2.4.38	80
SSH	OpenSSH 7.9 p1 Raspbian	22
FTP	Synology Diskstation NAS	21

Tabla 2: Tabla resultado de los servicios escaneados

En la siguiente figura vemos que el servicio ssh usa OpenSSH 7.9p1

kali@kali:~$ sudo nmap -A -sS
-p21,22,23,53,69,80,123,135,443,445,631,1433,3306 37.11.X.X

```
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Synology DiskStation NAS ftpd
22/tcp   open  ssh           OpenSSH 7.9p1 Raspbian 10+deb10u2+rpt1 (protocol 2.0)
| ssh-hostkey:
|   2048 63:28:e8:82:6b:95:98:20:25:1f:40:7d:d3:0f:43:dd (RSA)
|   256 2a:2a:5d:1b:a1:e2:82:f1:e5:92:b9:b8:9f:93:ba:b0 (ECDSA)
|_  256 01:ba:6a:18:6a:98:3e:9c:6d:40:a1:11:e2:0d:42:51 (ED25519)
```

Figura 33: Resultado Nmap syn scan

En la siguiente figura vemos que tiene el servicio FTP Synology y que está habilitado para el acceso anónimo.

kali@kali:~$ sudo nmap -A -sT
-p21,22,23,53,69,80,123,135,443,445,631,1433,3306 37.11.X.X

```
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Synology DiskStation NAS ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
22/tcp   open  ssh           OpenSSH 7.9p1 Raspbian 10+deb10u2+rpt1 (protocol 2.0)
| ssh-hostkey:
|   2048 63:28:e8:82:6b:95:98:20:25:1f:40:7d:d3:0f:43:dd (RSA)
|   256 2a:2a:5d:1b:a1:e2:82:f1:e5:92:b9:b8:9f:93:ba:b0 (ECDSA)
|_  256 01:ba:6a:18:6a:98:3e:9c:6d:40:a1:11:e2:0d:42:51 (ED25519)
```

Figura 34: Resultado Nmap connect scan

En la siguiente figura vemos que tiene activo el servicio Apache con httpd 2.4.38. y nos muestra la IP interna de la máquina.

```
80/tcp   open  http          Apache httpd 2.4.38
|_http-server-header: Apache/2.4.38 (Raspbian)
|_http-title: Did not follow redirect to https://192.168.1.120
123/tcp  open  ntp?
135/tcp  open  msrpc?
443/tcp  open  ssl/http      Apache httpd 2.4.38
|_http-server-header: Apache/2.4.38 (Raspbian)
|_http-title: 400 Bad Request
| ssl-cert: Subject: stateOrProvinceName=OR/countryName=US
| Not valid before: 2021-05-09T17:21:58
|_Not valid after:  2022-05-09T17:21:58
|_ssl-date: ERROR: Script execution failed (use -d to debug)
| tls-alpn:
|_  http/1.1
```

Figura 35: Resultado Nmap

Aunque muestra errores, vemos que tiene abierto el servicio MYSQL.

```
3306/tcp open  mysql?
|_mysql-info: ERROR: Script execution failed (use -d to debug)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_ssl-date: ERROR: Script execution failed (use -d to debug)
|_sslv2: ERROR: Script execution failed (use -d to debug)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
```

Figura 36: Resultado Nmap connect scan MYSQL

Una vez obtenido el listado de servicios disponibles, podemos entrar más profundidad usando los propios scripts de Nmap.

Los siguientes scripts sirven para analizar **MSSQL** y **MYSQL**.

En la siguiente figura, podemos ver que nos dice la versión del producto SQL server 2000 SP1, ya está desfasado y es vulnerable a múltiples exploits.

$sudo nmap --script ms-sql-info -p1433 37.11.XX

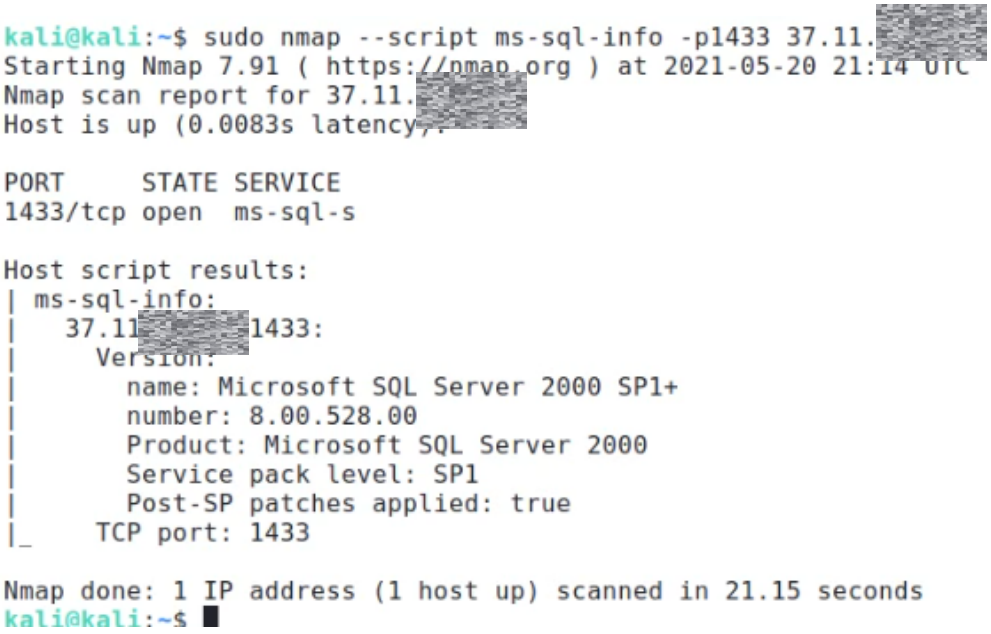


Figura 37: Resultado Nmap con script ms-sql-info

En la siguiente figura, vemos que el script de ataque de fuerza bruta nos enumera ciertos usuarios que están activos en la base de datos.

$sudo nmap --script ms-sql-brute -p1433 37.11.XX

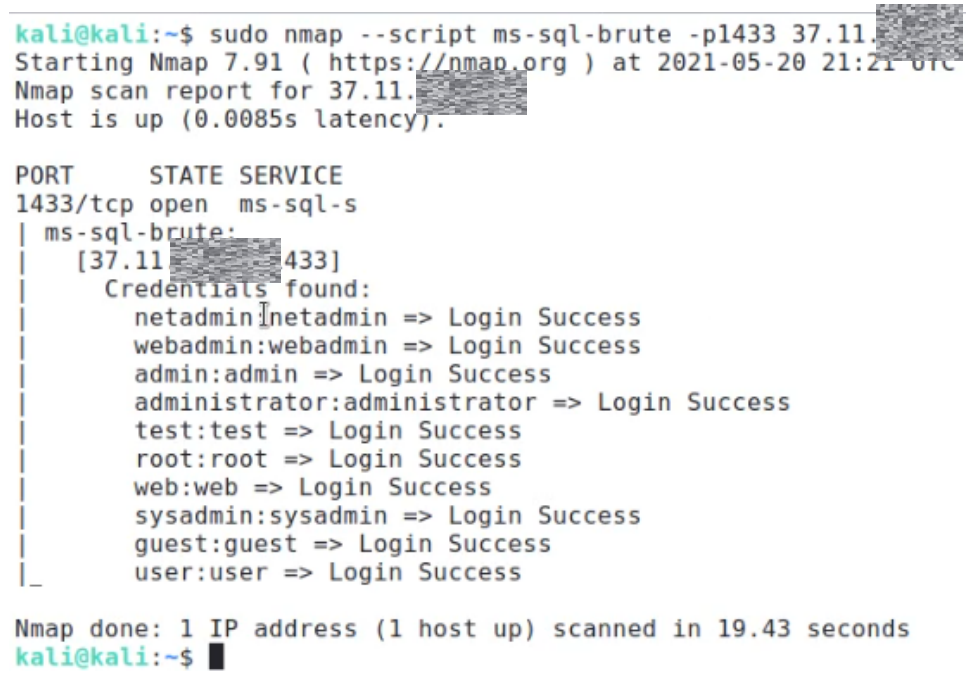


Figura 38: Resultado Nmap con script ms-sql-brute

Los siguientes scripts sirven para analizar **SMB**:

En la siguiente figura, podemos ver que nos enumera las carpetas compartidas.

$sudo nmap --script smb-enum-shares -p445 37.11.XX

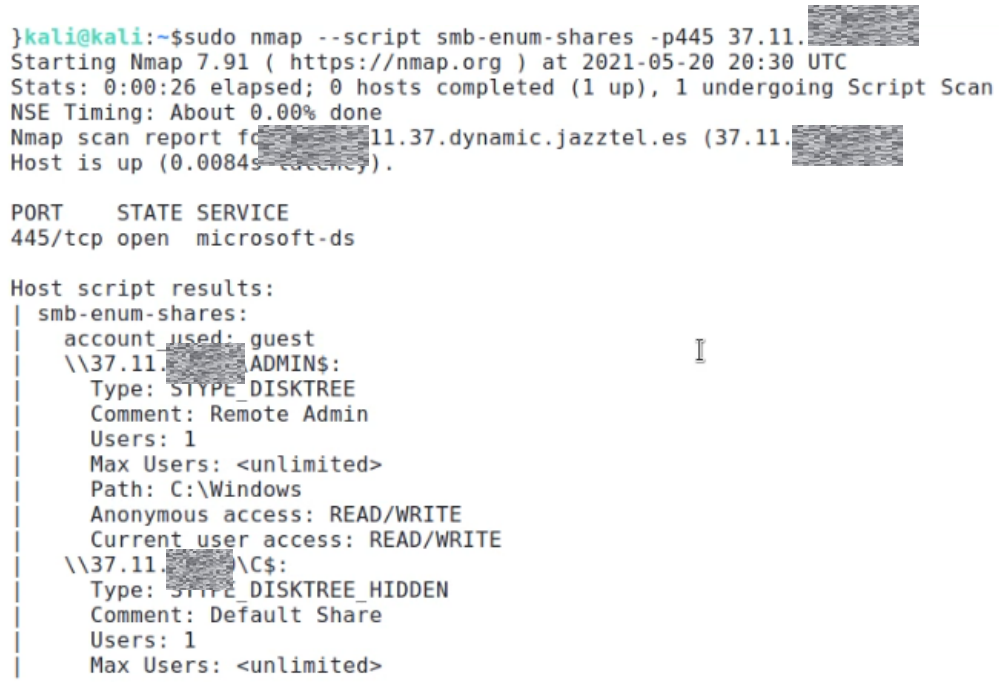


Figura 39: Resultado Nmap con script smb-enum-shares

En la siguiente figura, vemos que el script lanza un ataque de fuerza bruta contra SMB, en este caso sin éxito.

$sudo nmap --script smb-brute -p445 37.11.XX

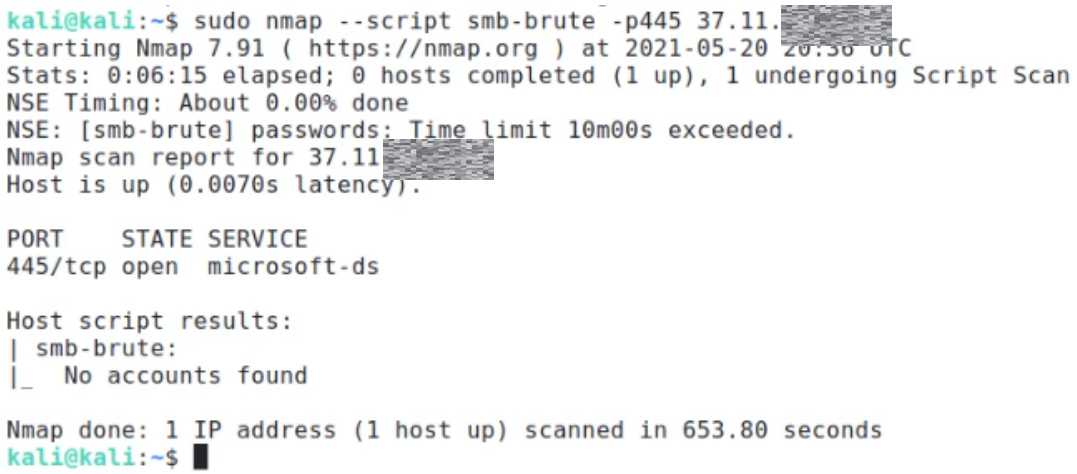


Figura 40: Resultado Nmap con script smb-brute

Comentar que también se probó la herramienta "*Legion*" (GoVanguard) incluida en Kali. Dicha herramienta permite realizar una gran cantidad de escaneo de vulnerabilidades de forma automatizada.

5.3.2 Escaneo con Nikto. RapidScan.

La ejecución de esta herramienta contra servidores no autorizados puede conllevar penas de cárcel o sentencias administrativas.

Como en el escaneo anterior hemos visto que hay servicios Web disponibles, pasamos al escaneo de vulnerabilidades web con Nikto (Sullo).

Nikto: herramienta "*Open Source*" de escaneo de vulnerabilidades web escrita en lenguaje Perl. Viene instalada en Kali por defecto y para ejecutarla simplemente debemos escribir nikto -h que indicará el host (ya sea por IP o por nombre) y -p que indica el puerto, normalmente 80 o 443.

La sintaxis de ejecución de la herramienta es la siguiente:

$nikto -h 37.11.XXX -p 80

Al pasar la herramienta al dispositivo Honeypot, puerto 80, muestra 4 ítems. Entre los ítems detectados, podemos ver una vulnerabilidad a Cross Site Scripting. También muestra la IP interna, el tipo de servicio Web Apache y sistema operativo Raspbian.

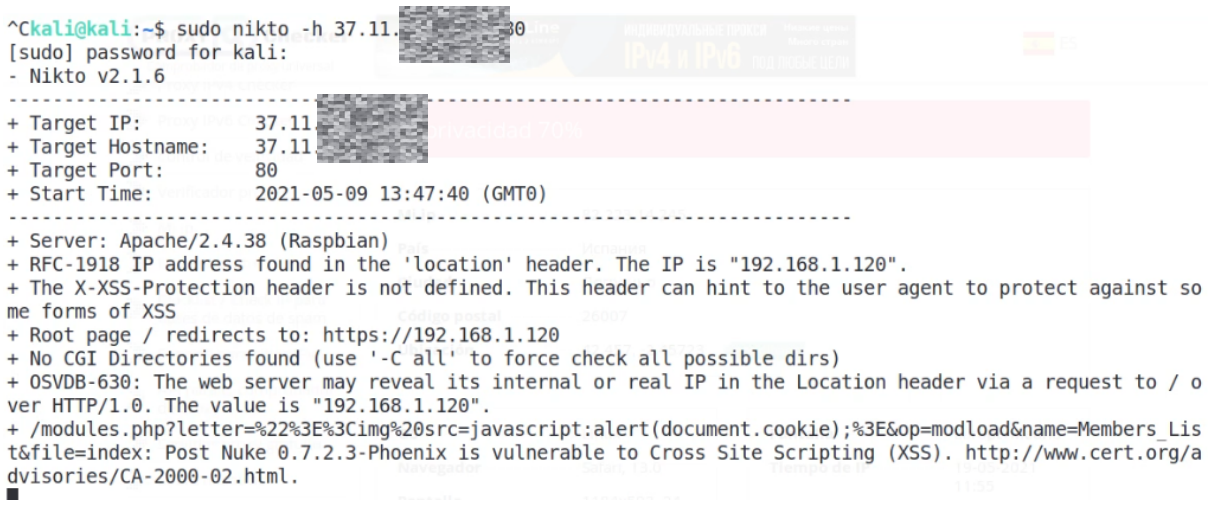


Figura 41: Escaneo Nikto http

La sintaxis es la misma para https cambiando solo el puerto.

$nikto -h 37.11.XXX -p 443

El escaneo al puerto 443 encuentra 3 ítems igual que el anterior, pero no el vulnerable a XSS.

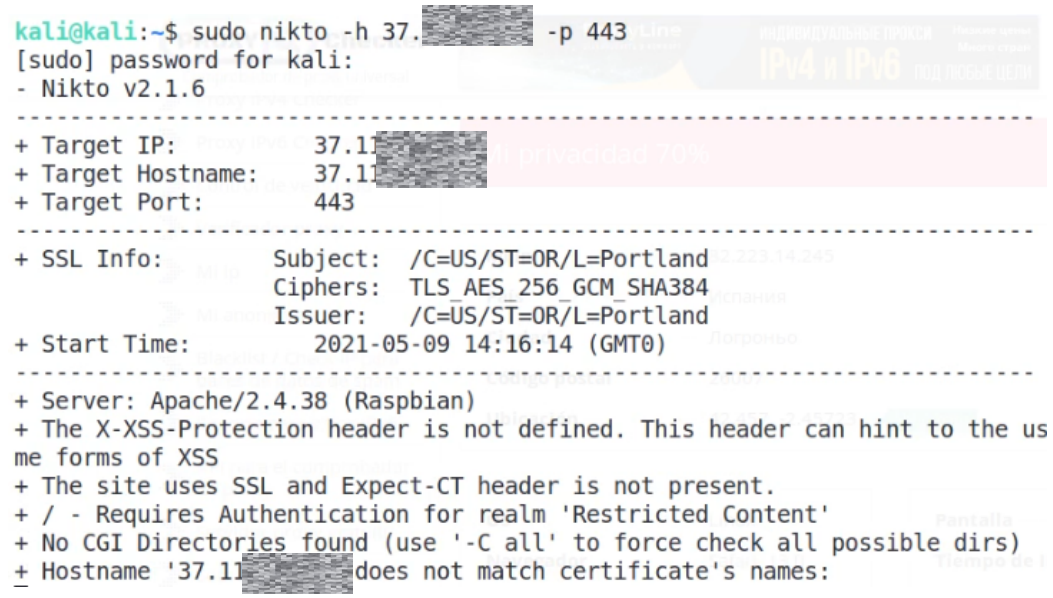


Figura 42: Escaneo Nikto https

A medida que vaya realizando el test de vulnerabilidades nos las irá mostrando por pantalla. Es una herramienta sencilla pero que nos mostrará información interesante sobre el servicio Web y sobre alguna de sus vulnerabilidades.

Podemos realizar un análisis más completo utilizando RapidScan (Skavngr) para comprobar que los resultados de Nikto son correctos y así ver si encontramos alguna vulnerabilidad más. Esta herramienta utiliza a su vez otras herramientas para analizar más de 80 vulnerabilidades Web entre ellas Nmap, Dnswalk y Nikto.

Rapidscan: Script escrito en Python. Lanza una batería de escaneo de vulnerabilidades web usando varias herramientas. En la siguiente figura podemos ver algunas de sus funcionalidades.

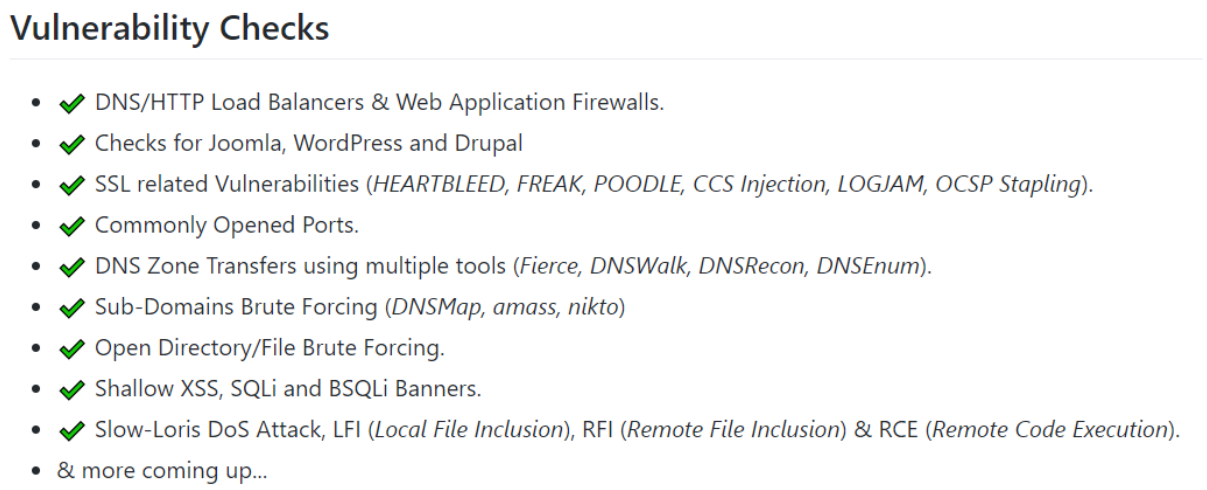


Figura 43: Algunas características del escaneo con Rapidscan (Skavngr)

Primero lo descargamos con el siguiente comando:

Lo descargamos:

$wget -O rapidscan.py https://raw.githubusercontent.com/skavngr/rapidscan/master/rapidscan.py && chmod +x rapidscan.py

Cuando lo ejecutamos por primera vez, revisará si dispone de las herramientas necesarias para lanzar el escaneo, si no dispone de ellas, podemos pararlo, instalar las necesarias y volver a lanzarlo. Kali por defecto viene sin "*Golismero*", "*Dnswalk*", "*Dnsmasp*", "*Xsser*" y "*Uniscan*". Golismero hay que descargarla de su repositorio web (Golismero) y seguir los pasos de instalación (Astudillo B). El resto simplemente con el comando "*apt-get install dnswalk dnsmap xsser uniscan*". Una vez instaladas ya aparecerán todas las disponibles.

En la siguiente figura podemos ver el cheking que realiza la herramienta al inicio de su ejecución:

Figura 44: Herramientas instaladas para pasar RapidScan

Lo ejecutamos:

$sudo python2.7 rapidscan 37.11.XXX

Una vez lanzado el escaneo, ejecutará las 80 fases. Algunas de ellas como el escaneo UDP hay que saltarlo, ya que la red Tor no lo permite, no nos aportará nada y tarda en realizarse varias horas.

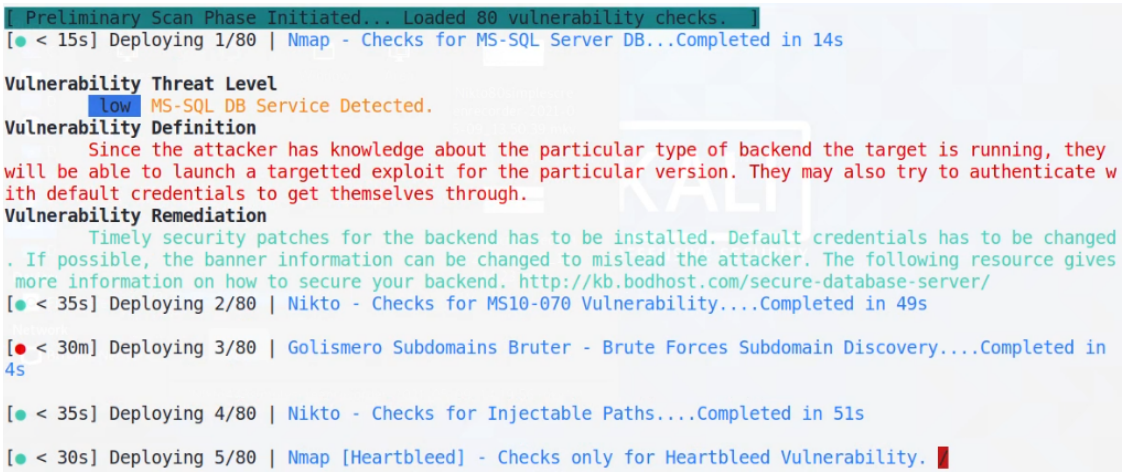


Figura 45: Proceso de escaneo con RapidScan.

La herramienta ejecutará más de 80 test de vulnerabilidades y generará dos informes, uno de vulnerabilidades llamado "*RS-Vulnerability-Report*" y otro de depuración y errores llamado "*RS-Debug-ScanLog*" que incluirá más información.

Información de las herramientas y alguno de los ejemplos de los informes son los siguientes:

RS-Vulnerability-Report

Golismero: Es una herramienta de búsqueda de vulnerabilidades escrita en Python2.X y bastante potente. Permite realizar una batería de escaneos y presentar un informe gráfico. En la siguiente captura vemos que ha usado Nikto para buscar ciertas vulnerabilidades XSS, CGI, etc. (Astudillo B).

```
/-----------------------------------------\
| GoLismero 2.0.0b6, The Web Knife        |
| Copyright (C) 2011-2014 GoLismero Project |
|                                         |
| Contact: contact@golismero-project.com  |
\-----------------------------------------/

GoLismero started at 2021-05-09 21:53:42.539145 UTC
[1m[34m[*] GoLismero[0m: Audit name: golismero-xksBAIIN
[1m[34m[*] GoLismero[0m: Added 2 new targets to the database.
[1m[34m[*] GoLismero[0m: Launching tests...
[1m[34m[*] GoLismero[0m: Current stage: [35mScanning (non-intrusive)[0m
[1m[34m[*] Nikto[0m: Launching Nikto against: 37.11[redacted]
[1m[34m[*] Nikto[0m: - Nikto v2.1.6
[1m[34m[*] Nikto[0m: ---------------------------------------------------------------------------
[1m[34m[*] Nikto[0m: + Target IP:          37.1[redacted]
[1m[34m[*] Nikto[0m: + Target Hostname:    37.1[redacted]
[1m[34m[*] Nikto[0m: + Target Port:        80
[1m[34m[*] Nikto[0m: + Start Time:         2021-05-09 21:53:46 (GMT0)
[1m[34m[*] Nikto[0m: ---------------------------------------------------------------------------
[1m[34m[*] Nikto[0m: + Server: Apache/2.4.38 (Raspbian)
[1m[34m[*] Nikto[0m: + RFC-1918 IP address found in the 'location' header. The IP is "192.168.1.120".
[1m[34m[*] Nikto[0m: + The X-XSS-Protection header is not defined. This header can hint to the user agent to protect
against some forms of XSS
[1m[34m[*] Nikto[0m: + Root page / redirects to: https://192.168.1.120
[1m[34m[*] Nikto[0m: + No CGI Directories found (use '-C all' to force check all possible dirs)
[1m[34m[*] Nikto[0m: + OSVDB-630: The web server may reveal its internal or real IP in the Location header via a request to
/ over HTTP/1.0. The value is "192.168.1.120".
[1m[34m[*] Nikto[0m: + /modules.php?letter=%22%3E%3Cimg%20src=javascript:alert(document.cookie);
%3E&op=modload&name=Members_List&file=index: Post Nuke 0.7.2.3-Phoenix is vulnerable to Cross Site Scripting (XSS).
http://www.cert.org/advisories/CA-2000-02.html.
[1m[34m[*] Nikto[0m: + 7350 requests: 0 error(s) and 4 item(s) reported on remote host
[1m[34m[*] Nikto[0m: + End Time:           2021-05-09 22:27:04 (GMT0) (1998 seconds)
[1m[34m[*] Nikto[0m: ---------------------------------------------------------------------------
[1m[34m[*] Nikto[0m: + 1 host(s) tested
[1m[34m[*] Nikto[0m: Nikto found 4 vulnerabilities for host: 37.11[redacted]
[1m[34m[*] GoLismero[0m: Current stage: [35mReporting[0m
```

Figura 46: Resultado de Golismero

Uniscan: (Velasco y Redes zone) Es una herramienta escrita en Perl que también se utiliza para buscar vulnerabilidades web. En las siguientes figuras podemos ver el resultado de alguno de sus escaneos.

```
SCAN RESULTS FOR 37.11.[redacted]:443 - 37.11[redacted]
-----------------------------------------------

* Session Renegotiation:
      Client Renegotiation DoS Attack:   OK - Not vulnerable
      Secure Renegotiation:              OK - Supported


SCAN COMPLETED IN 1.77 S
------------------------
```

Figura 47: Resultado de Uniscan

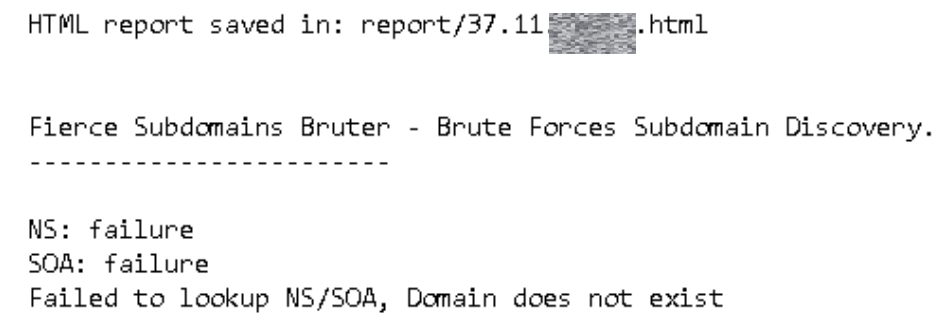


Figura 48: Resultado de Uniscan

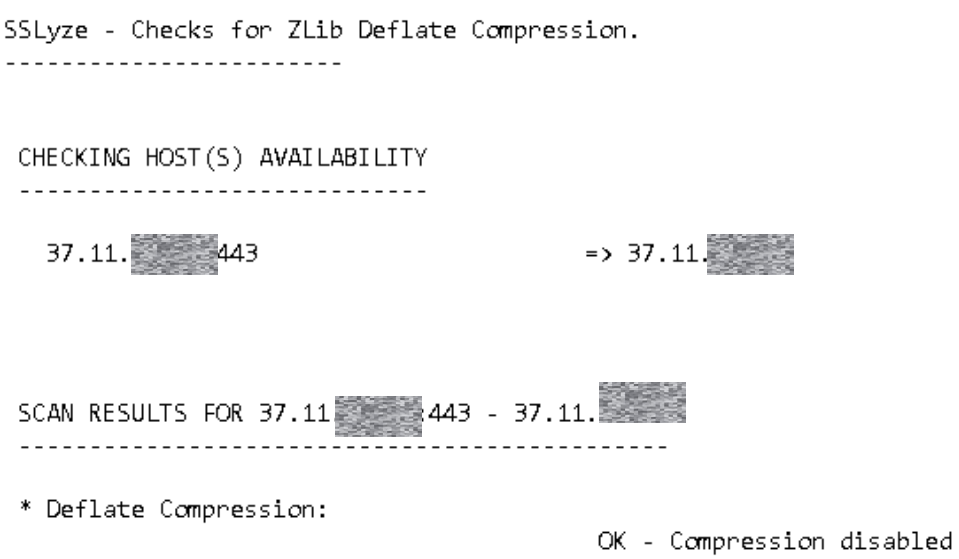


Figura 49: Resultado de Uniscan

Dirb: Es una herramienta de enumeración de subdirectorios web mediante creación de diccionario y aplicación de fuerza bruta.

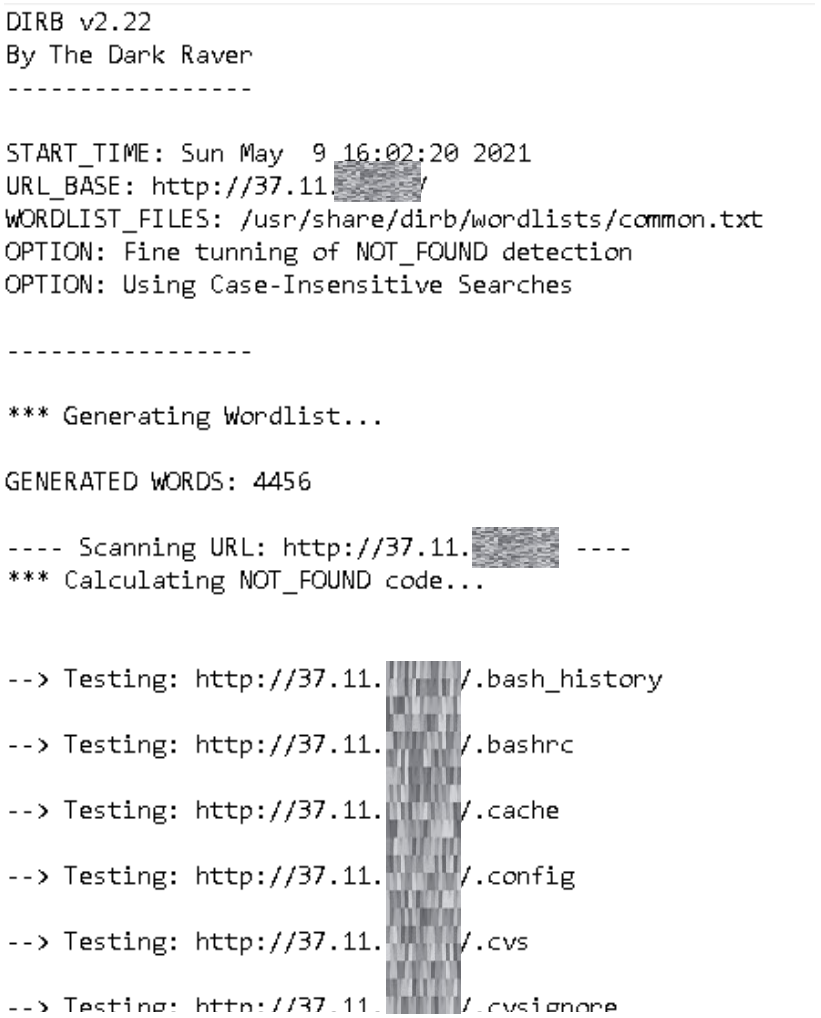


Figura 50: Resultado de DirB

Una vez finalizada la fase de escaneo y encontradas varias las líneas de ataque mostradas en la tabla anterior, podemos proseguir a la fase de "*obtener acceso*" mediante la herramienta Armitage.

5.3.3. Armitage.

La ejecución de esta herramienta contra servidores no autorizados puede conllevar penas de cárcel o sentencias administrativas.

Armitage: (Dragonjar) se trata de un interfaz gráfico de Metasploit. Se podría utilizar para las fases de escaneo, acceso, mantener acceso y limpieza de huellas debido a que dispone de múltiples herramientas. En este caso, solo lo usamos como modo de ejemplo de continuación a la fase de "*escaneo*", la fase "*obtener acceso*".

Arrancamos primero postgresql.

$sudo service postgresql start

Luego arrancamos Armitage

$sudo armitage

Nota: Al usar los exploit con Shell reversible, viene por defecto nuestra IP. Si lanzamos el exploit sin modificarlo perderemos el anonimato. También sucederá lo mismo con los ataques UDP, ya que Tor no los soporta.

Después de arrancar el servicio de postgresql, abrimos Armitage y nos dirá que es necesario levantar el servicio de escucha de Metasploit. Después de decirle que sí, veremos un entorno como el de la siguiente figura.

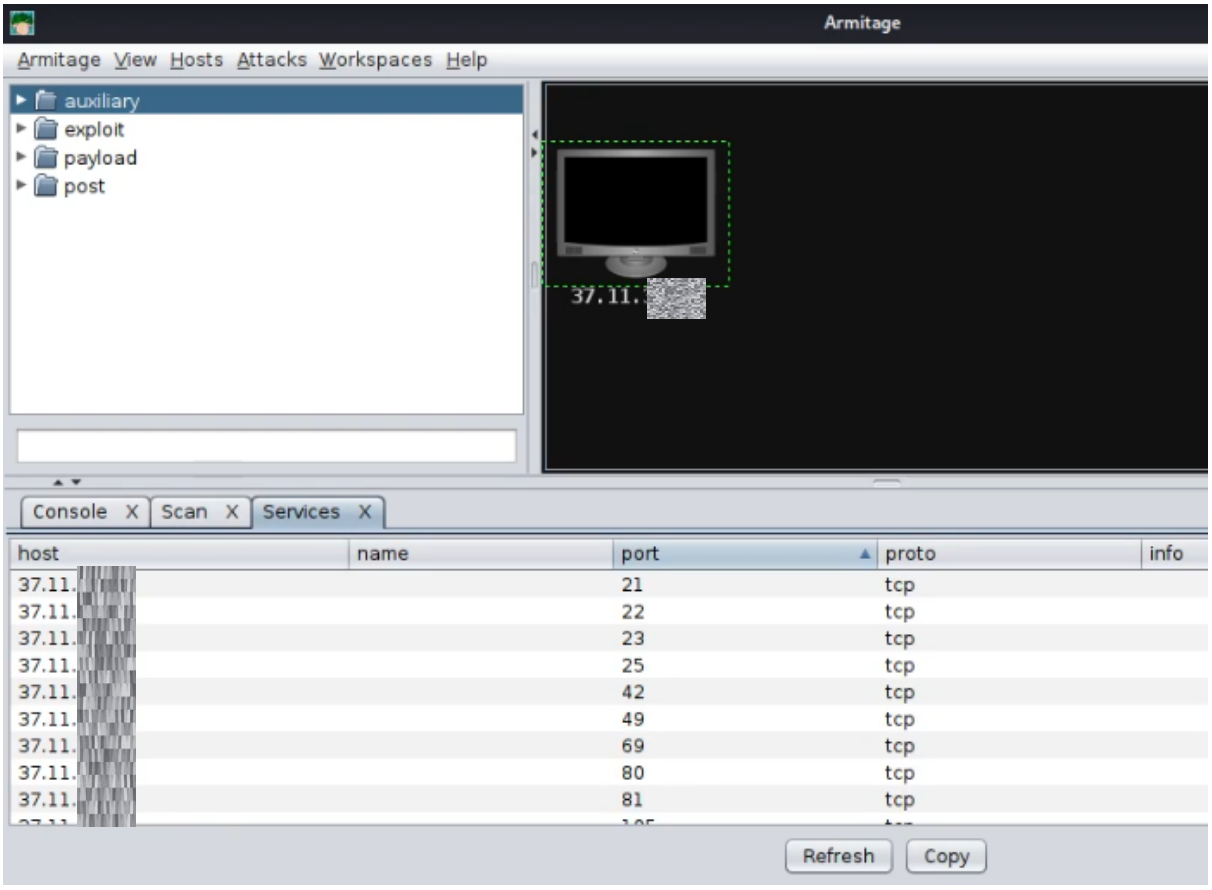


Figura 51: GUI de Armitage

En la parte superior izquierda tenemos las herramientas para crear la Shell de escucha, los exploits y payloads de Metasploit y las herramientas post explotación. En la parte superior derecha tenemos el host víctima sobre el que lanzar los

escaneos, el exploit y demás acciones. En la parte inferior, veremos las pestañas con los resultados de cada acción realizada.

El funcionamiento de Armitage se compone de varios pasos. primero se realiza la fase de escaneo o importación de los datos de la víctima, luego posteriormente se realiza obtener acceso, después se mantiene el acceso y por último se borran las huellas. Como hemos comentado, en este trabajo sólo nos interesa la fase de escaneo y la de obtener acceso.

1.- Fase de escaneo: En esta fase, importamos el escaneo ya realizado con otras herramientas o añadimos manualmente el host. En la parte superior derecha aparecerá el host víctima. En la siguiente figura, podemos ver que hemos añadido un host y que ha terminado el escaneo.

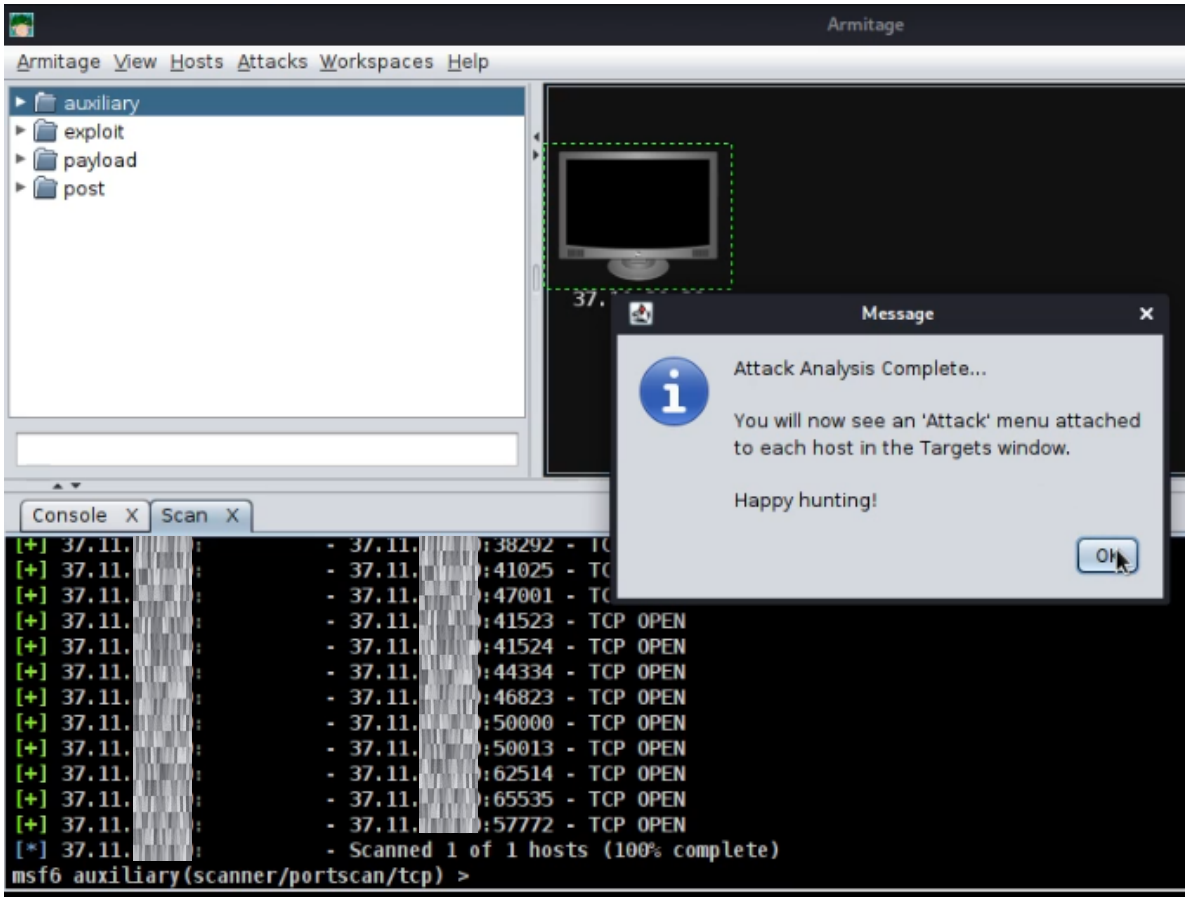


Figura 52: Fase de escaneo con Armitage

Si pulsamos botón derecho sobre el host, services, nos saldrá una pestaña en la parte de abajo indicándonos los servicios activos a explotar.

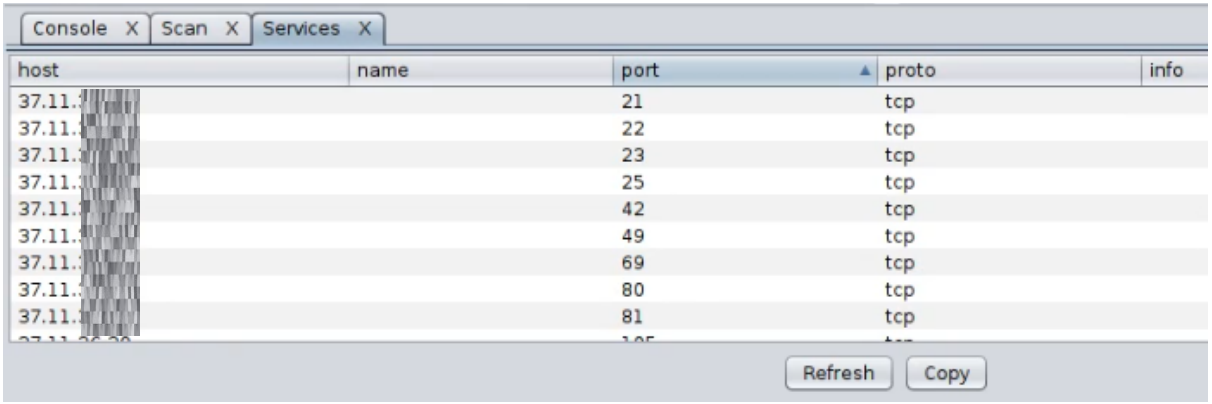


Figura 53: Servicios activos, Armitage

En la parte superior izquierda, tenemos las distintas herramientas para crear nuestra Shell inversa de escucha, buscar, preparar y lanzar nuestro exploit y payload y realizar la fase post-explotación (mantener acceso, migrar procesos, borrar huellas) en el caso de que logremos tener acceso.

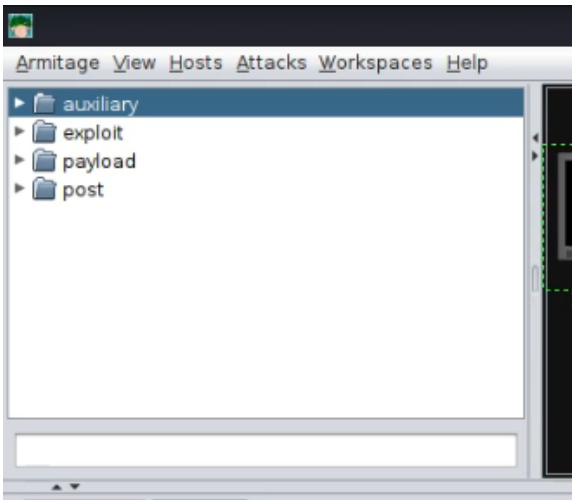


Figura 54: Ventana de opciones, Armitage

2.- Obtener acceso: En esta fase, seleccionamos uno de los exploits que anteriormente hemos confirmado como vulnerable. Configuramos los parámetros auxiliares de nuestra Shell inversa (payload) previamente preparada y lo lanzamos.

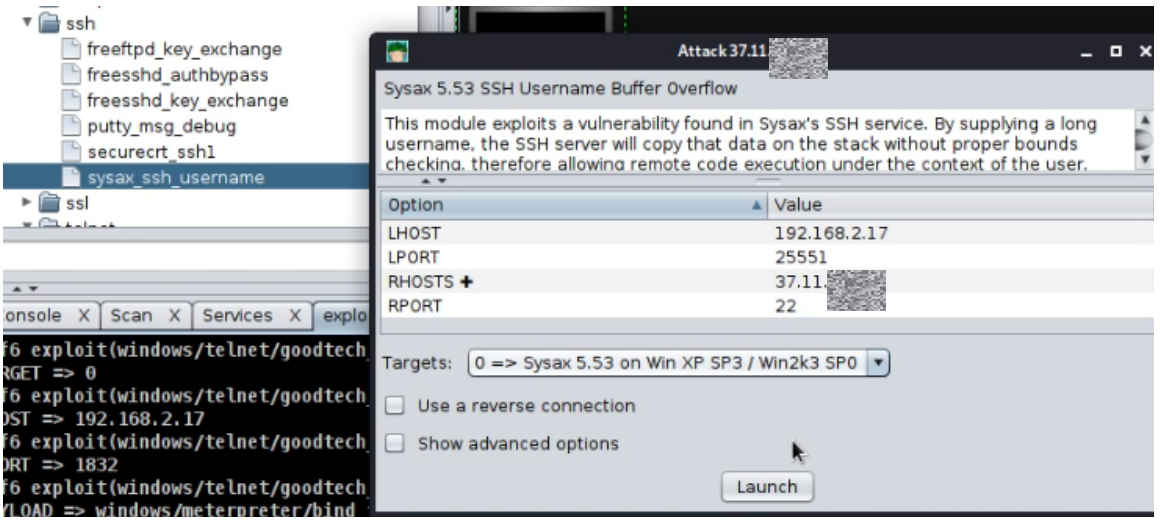


Figura 55: Configurar exploit, Armitage

Si todo ha ido bien, nos aparecerá que tenemos una sesión de Meterpreter abierta. En la siguiente figura aparecen 3 sesiones Meterpreter abiertas.

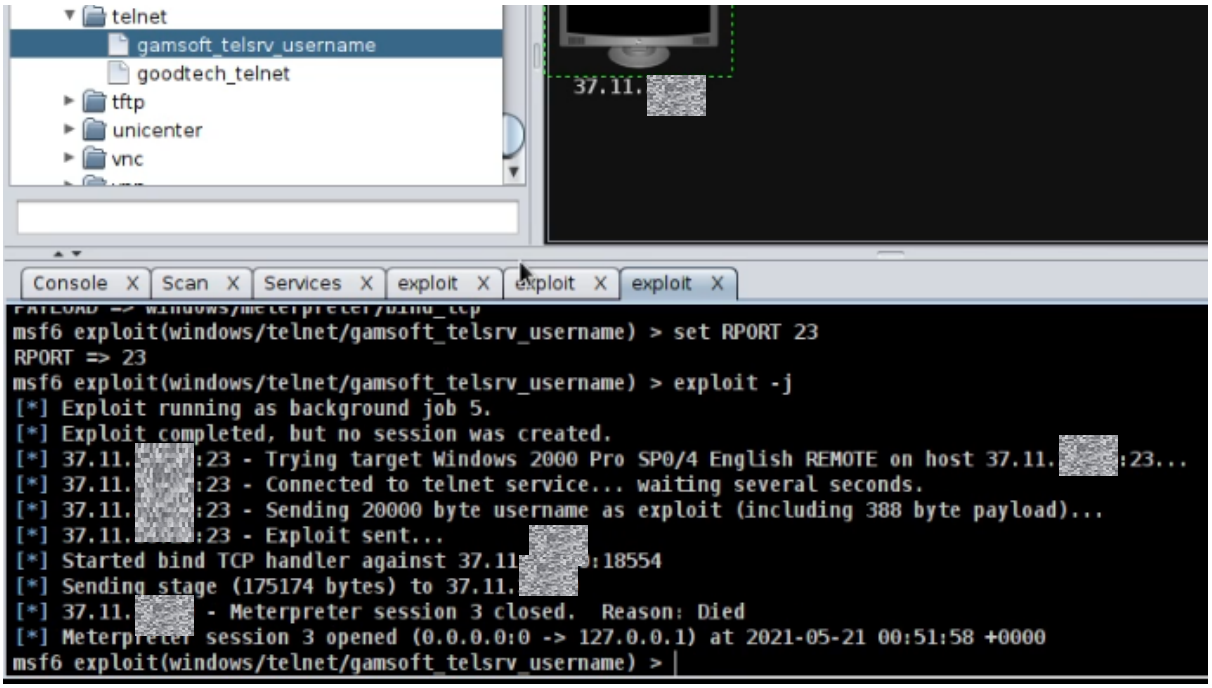


Figura 56: Lanzar exploit y Meterpreter, Armitage

Como nuestra máquina víctima era un Honeypot y no es vulnerable sino que simula serlo, no nos molestamos en preparar nuestra sesión de escucha remota (payload), pero aún así, nos comunicará que funcionó y qué la sesión de Meterpreter está abierta. Sin embargo, enseguida nos la cerrará. Esto se debe a que no es una máquina vulnerable a esos exploit.

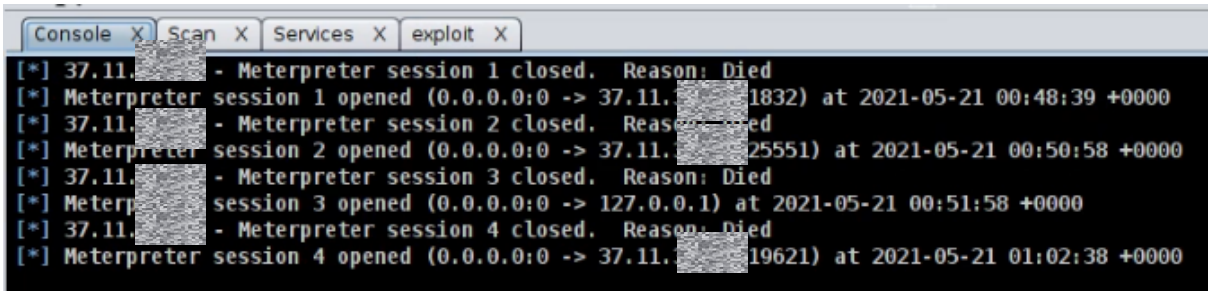


Figura 57: Sesiones Meterpreter

Para disponer de más evidencias, se lanzaron más intentos de acceso:

En la siguiente captura vemos el ataque lanzado "*sysax_ssh_username"* sobre el servicio ssh. Para ejecutarlo, debemos arrastrar el exploit desde la parte izquierda hacia la máquina víctima de la derecha, se nos abrirá la configuración del exploit y la configurarán sus parámetros y payload. En LHOST, habría que configurar la IP del sistema que tenga en escucha la Shell inversa, ya sea creada con Meterpreter o con nc (netcat).

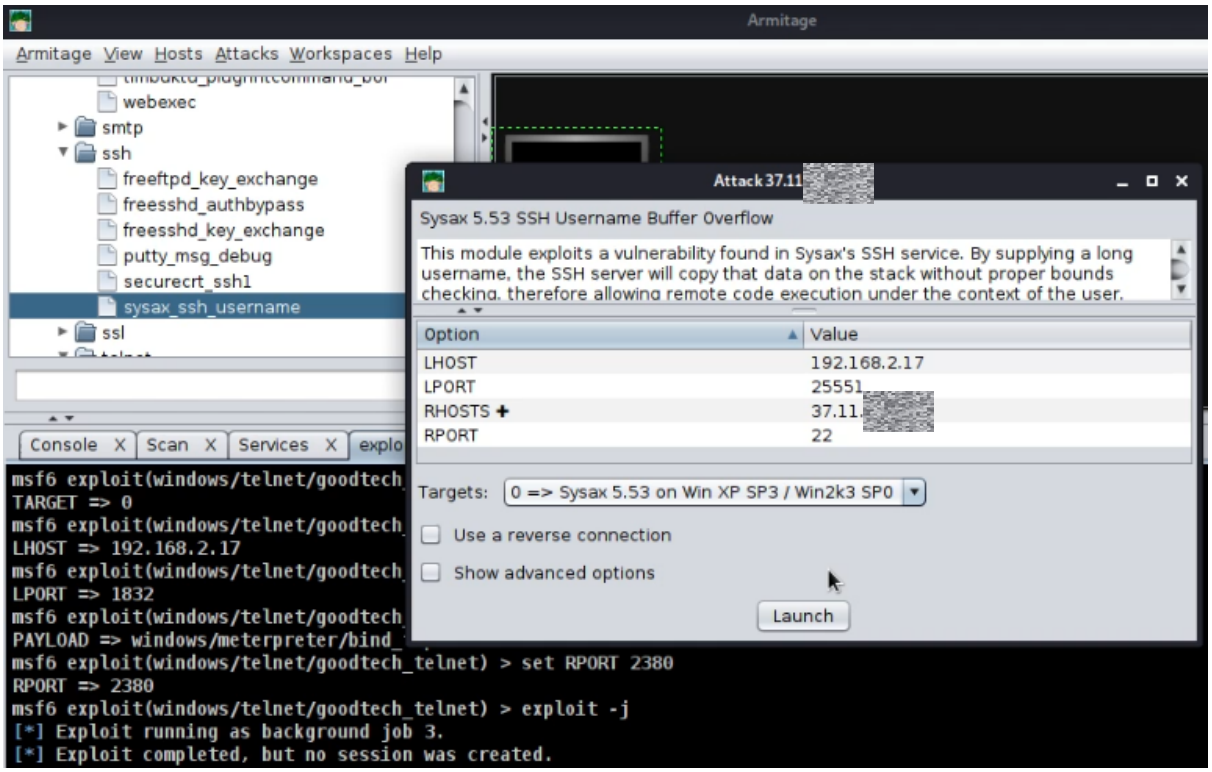


Figura 58: Pantalla pre-exploit de Armitage

Otro exploit lanzado al servicio TELNET es “*gamsoft_telsrv_username*”. En la siguiente figura también vemos que es capaz de abrir una sesión pero que inmediatamente se cierra por la misma razón que en el antrior ataque.

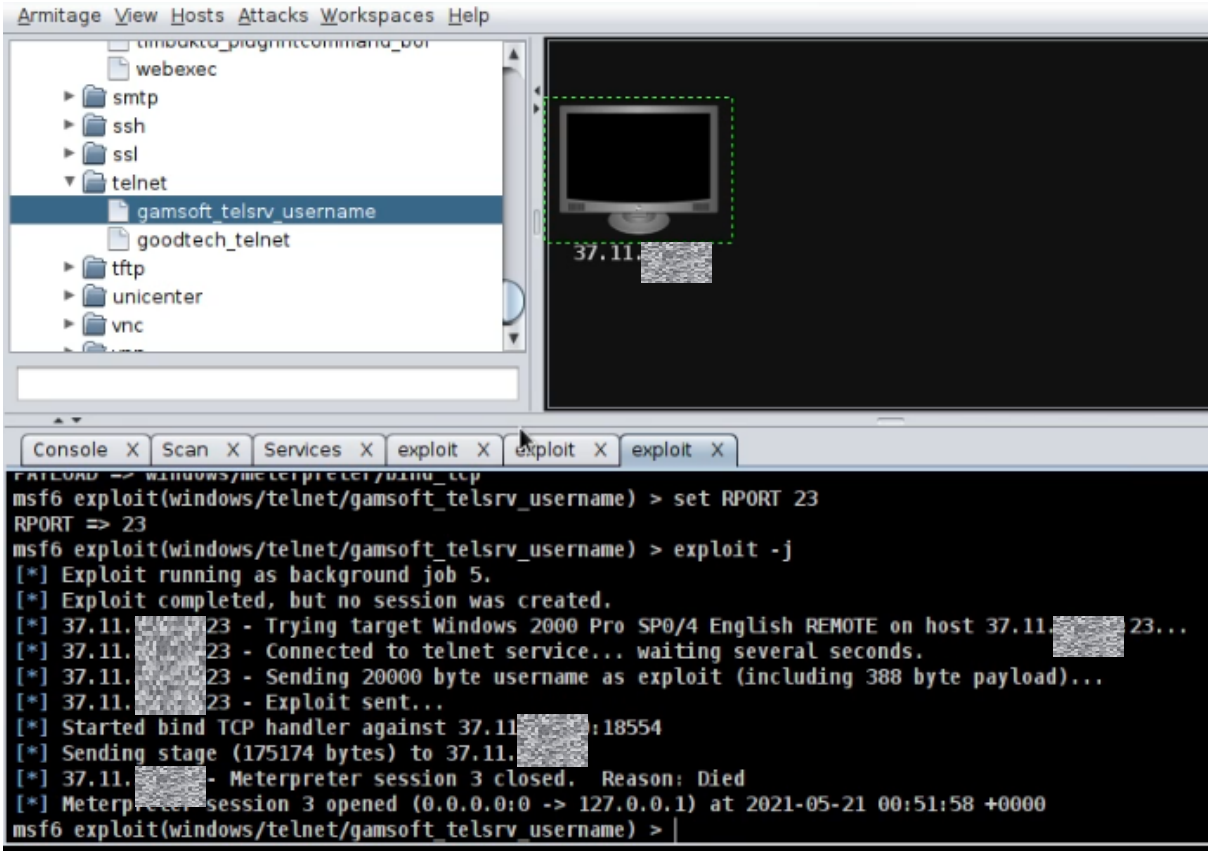


Figura 59: Exploit telnet de Armitage

El siguiente ejemplo, es un ataque sobre el servicio TFTP, como es usando protocolo UDP y Tor no es capaz de enrutarlo, no se realizó correctamente en las pruebas de conexión 3G, pero si en Ethernet. Esto es así porque el ataque lo mandó por la red interna del router a la red interna del Router doméstico, esa prueba no sería válida, ya que en un entorno real, nunca llegaría a la víctima usando el protocolo UDP.

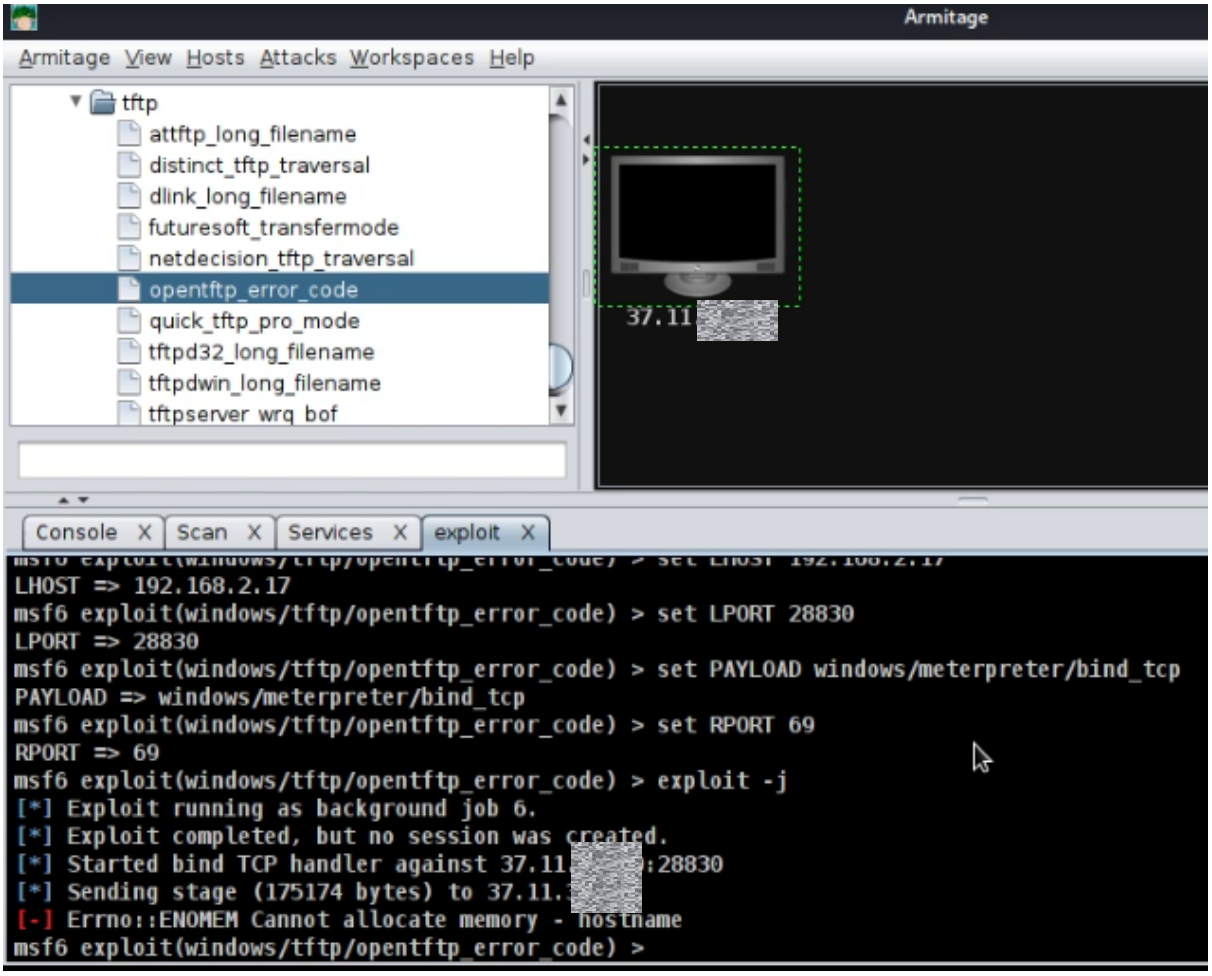


Figura 60: Exploit tftp de Armitage

El siguiente es un ejemplo de exploit “*apache_chunked*” contra el servicio web de Apache que nos generó una Shell inversa

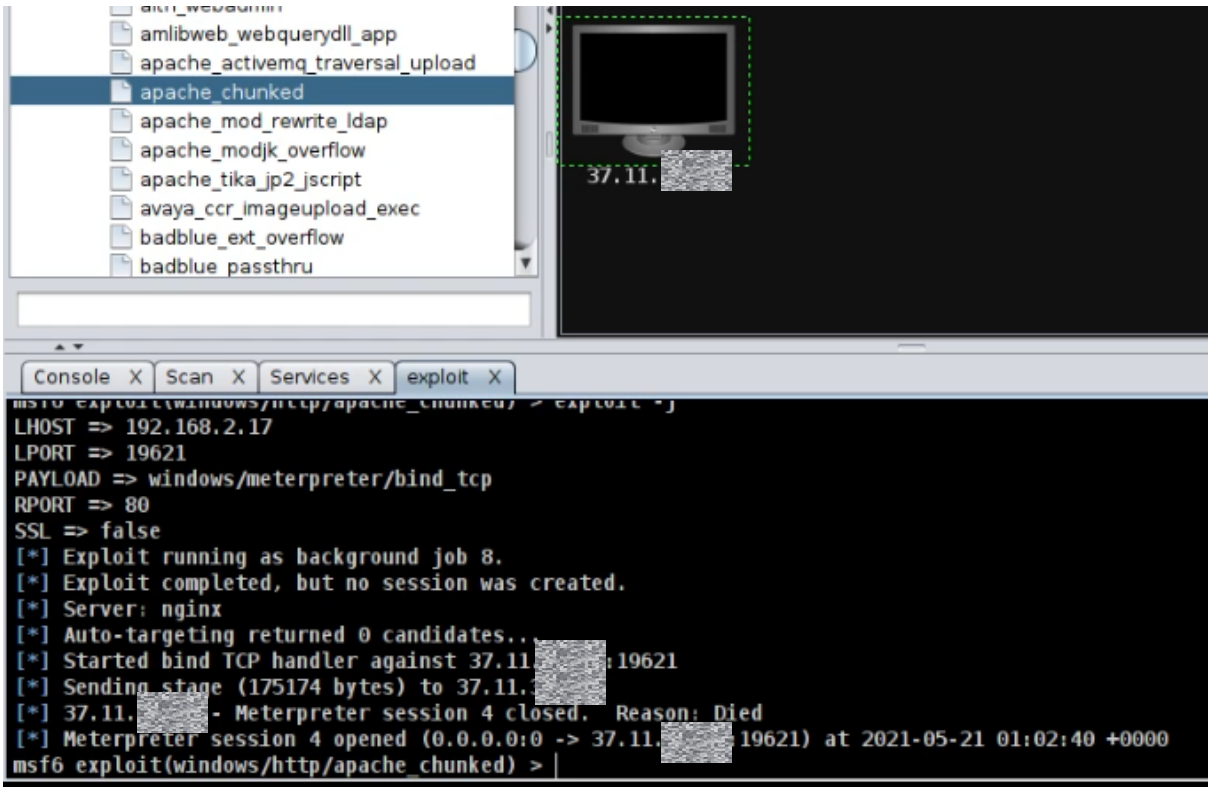


Figura 61: Exploit Apache de Armitage

Otro ataque al servicio de base de datos.

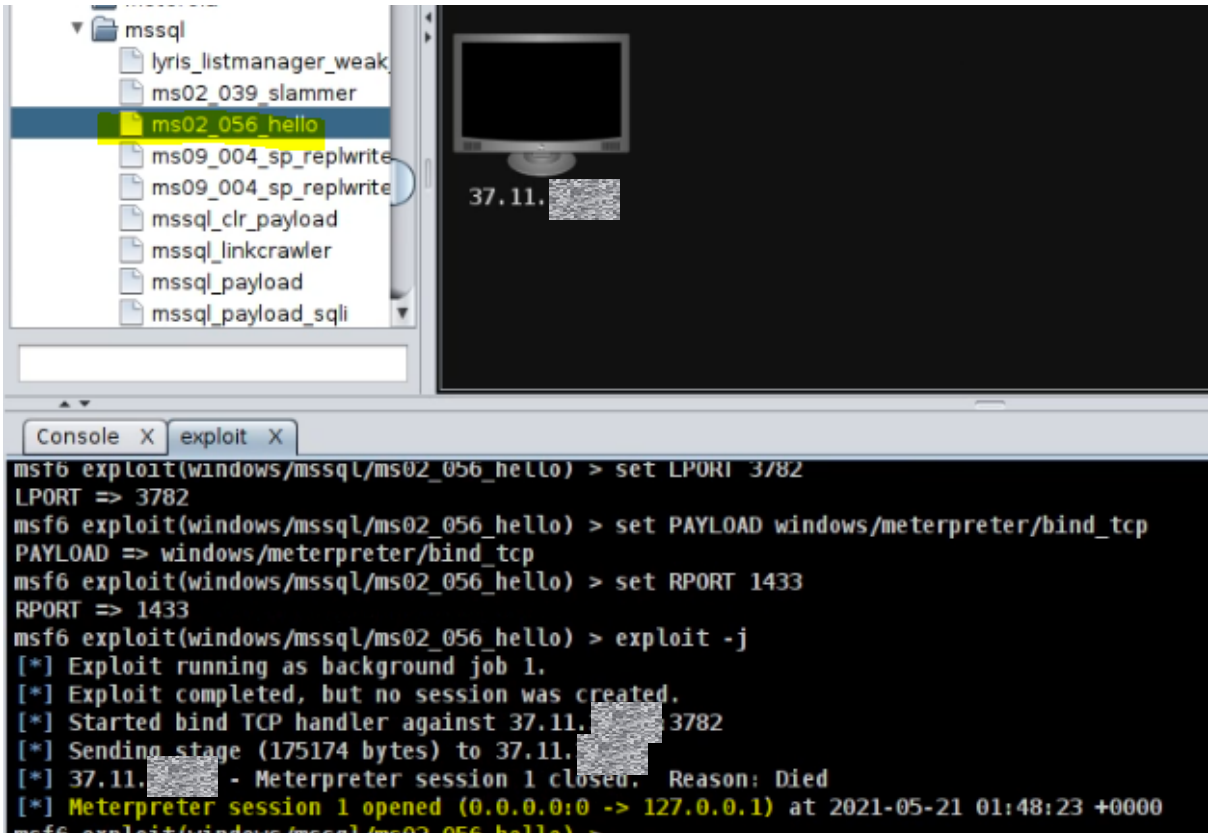


Figura 62: Exploit MSSQL de Armitage

Por último, en la siguiente captura, podemos ver la cantidad de sesiones de Meterpreter abiertas con éxito e inmediatamente cerradas por nuestro Honeypot.

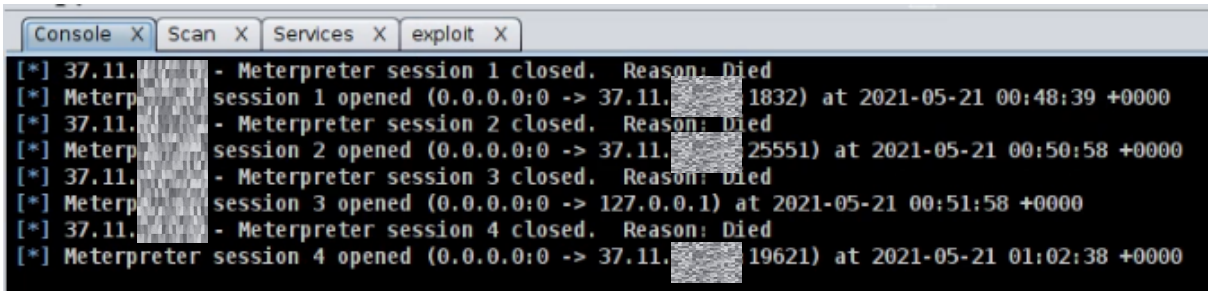


Figura 63: Sesiones Meterpreter de Armitage

5.4 Evidencias Honeypot.

Como ya hemos comentado a lo largo de este trabajo, uno de los objetivos es recoger los ataques dirigidos por la máquina pentester en busca de evidencias de pérdida de anonimato. Para ello, vamos a usar un Honeypot llamado Dionaea, un recopilador de tráfico llamado NTopng y en algún caso la herramienta Wireshark. Ver instalación en anexo D.

Una vez instalados, ejecutamos **Dionaea** con el siguiente comando:

$sudo /opt/dionaea/bin/dionaea

Figura 64: Ejecución Dionaea

Para comprobar que el servicio está funcionando correctamente, podemos ejecutar el comando "*$sudo netstat -utap*" y nos mostrará todos los servicios en escucha de la Raspberry. Vemos que tenemos los esperados, DNS, FTP, SMB, TELNET, MYSQL, MSSQL, etc.

```
pi@raspberrypi - ~ netstat -utap
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ftp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:ftp    0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:domain 0.0.0.0:*               LISTEN
tcp        0      0 localhost:telnet        0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:telnet 0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:3000            0.0.0.0:*               LISTEN
tcp        0      0 localhost:ms-sql-s      0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.ho:ms-sql-s 0.0.0.0:*               LISTEN
tcp        0      0 localhost:1723          0.0.0.0:*               LISTEN
tcp        0      0 localhost:1883          0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:1723   0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:1883   0.0.0.0:*               LISTEN
tcp        0      0 localhost:microsoft-ds  0.0.0.0:*               LISTEN
tcp        0      0 raspberryp:microsoft-ds 0.0.0.0:*               LISTEN
tcp        0      0 localhost:5601          0.0.0.0:*               LISTEN
tcp        0      0 localhost:sip           0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:sip    0.0.0.0:*               LISTEN
tcp        0      0 localhost:sip-tls       0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.hom:sip-tls 0.0.0.0:*               LISTEN
tcp        0      0 localhost:epmap         0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:epmap  0.0.0.0:*               LISTEN
tcp        0      0 localhost:27017         0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:27017  0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 localhost:nameserver    0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:mysql  0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.:nameserver 0.0.0.0:*               LISTEN
tcp        0      0 localhost:11211         0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:11211  0.0.0.0:*               LISTEN
tcp        0      0 localhost:6379          0.0.0.0:*               LISTEN
tcp        0      0 localhost:9100          0.0.0.0:*               LISTEN
tcp        0      0 raspberrypi.home:9100   0.0.0.0:*               LISTEN
```

Figura 65: Netstat puertos en escucha

Una vez terminada la batería de pruebas con el dispositivo Kali y habiendo recibido innumerables ataques y escaneos por parte de este y de otras máquinas de Internet, podemos recoger nuestras evidencias almacenadas por Dionaea. En la carpeta /opt/dionaea/var/lib/dionaea/binaries podemos encontrar los binarios ejecutados intentando obtener una Shell. En la siguiente figura aparecen los que consiguió recopilar.

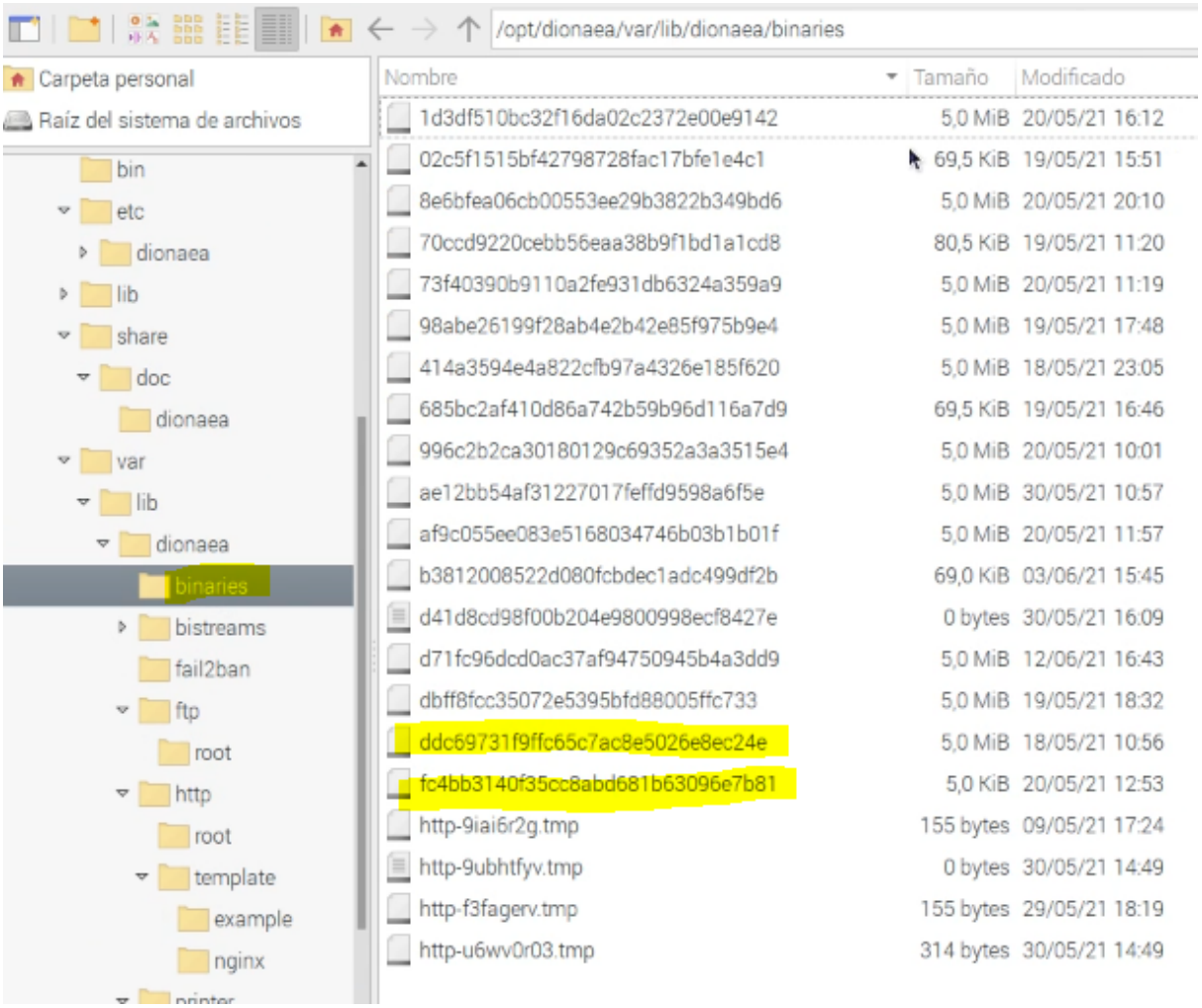


Figura 66: Dionaea, binarios recogidos

En la carpeta bistreams, clasificará, primero por carpetas, luego por servicios, los ataques recibidos a los mismos. En la siguiente figura vemos los servicios de HTTP, SMB, MSSQL, etc.

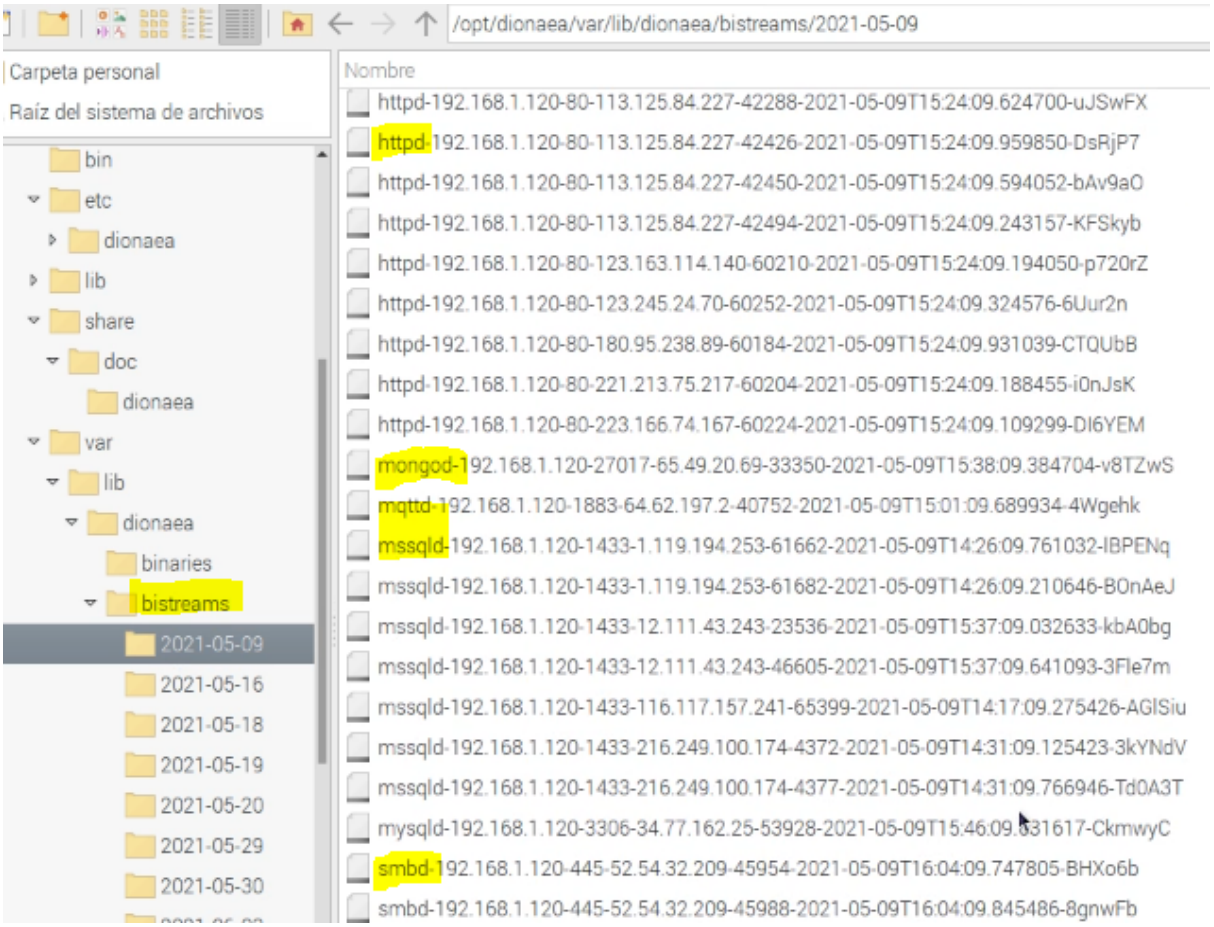


Figura 67: Dionaea, ataques recogidos

También registra en los logs las consultas y ataques realizados a cada servicio, SQL, HTTP, etc. En la siguiente captura podemos realizar una búsqueda por servicio. Como el archivo es grande, es más rápido consultarlo buscando el patrón con el comando "*cat fichero.log* | *grep tipo_de_servicio*".

Figura 68: Dionaea, búsqueda de servicios

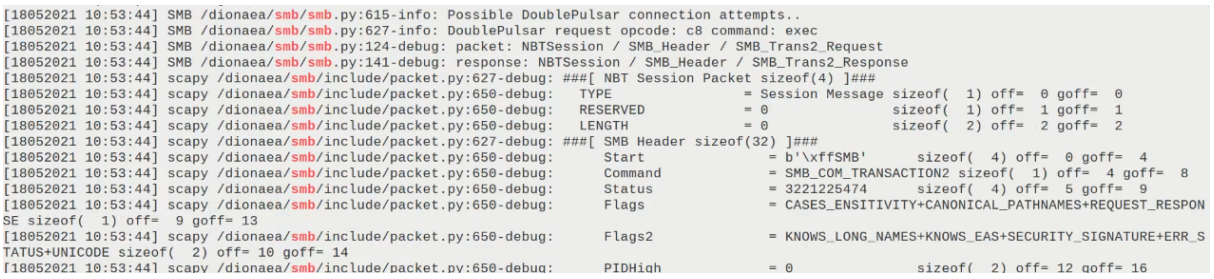


Figura 69: Dionaea, binarios recogidos

En las siguientes figuras podemos ver varios ataques recibidos al servicio MSSQL, enumeración de usuarios y varios test buscando vulnerabilidades de varios tipos.

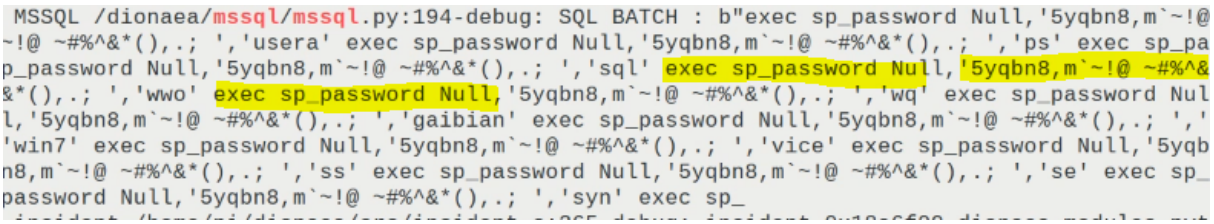


Figura 70: Dionaea, fuerza bruta contra mssql

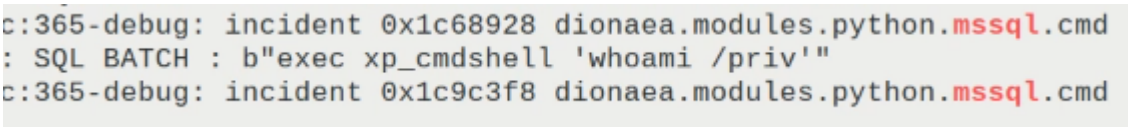


Figura 71: Dionaea, varios tipos de ataques

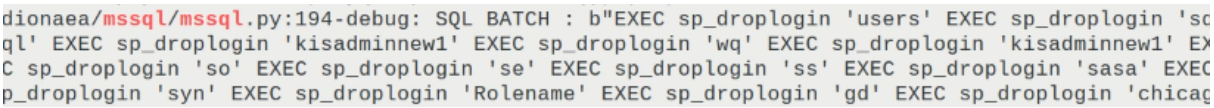


Figura 72: Dionaea, otro tipo de ataque por fuerza bruta

Como vemos, Dionaea nos proporciona muchos datos acerca de los ataques recibidos y puede ser una gran fuente de información como línea de investigación futura buscando más información de la capa de aplicación.

Por otra parte, para acceder a **NTopng**, hay que acceder al enlace a http://localhost:3000 desde el navegador. Lo primero mostrado será el "*dashboard*" inicial. En la parte superior encontraremos los menús de configuración y el recuadro "*search*" donde podremos buscar por IP. También encontramos el apartado Alertas donde nos avisará de las mismas.

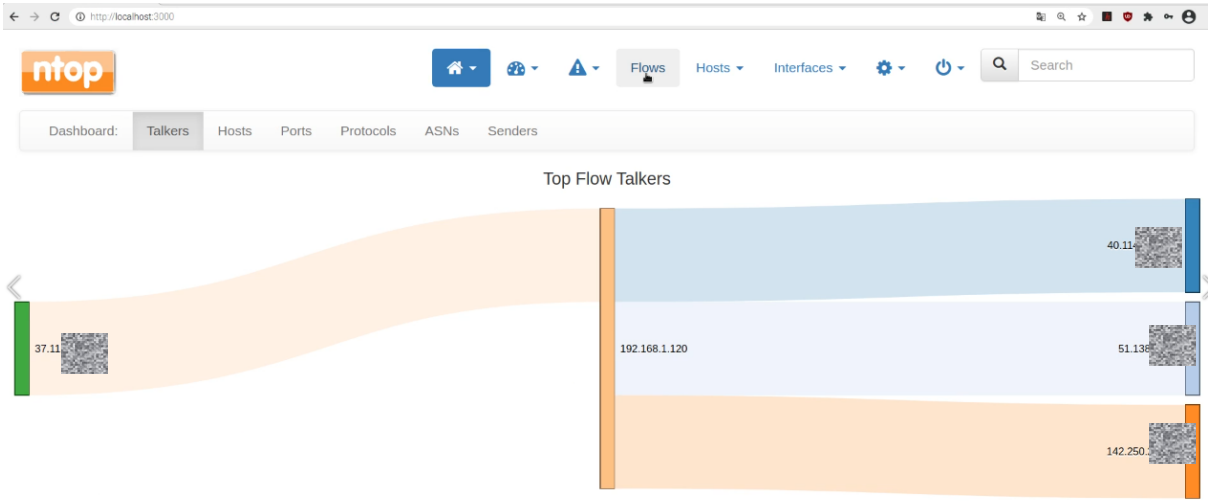


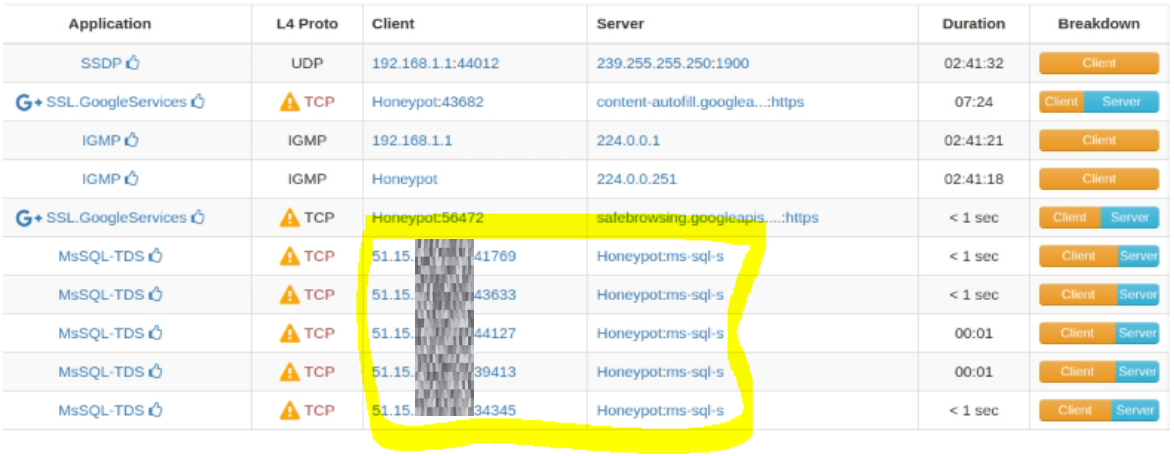
Figura 73: Dashboard NTopng

Aunque esta herramienta tiene bastantes opciones (Ver Anexo D), la que nos interesa es la del menú Flows (flujos). Mediante esta opción podemos ver todo el tráfico que está pasando en tiempo real y así detectar los escaneos y ataques realizados por NMap, Nikto, Rapiscan y Armitage. Comencemos viendo los escaneos recibidos por la herramienta **NMap**.

Los escaneos realizados en el punto anterior de Pentesting (escaneo de puertos) no reflejaron ninguna pérdida de anonimato. Podemos ver el de MYSQL, MSSQL y SMB en las siguientes figuras.

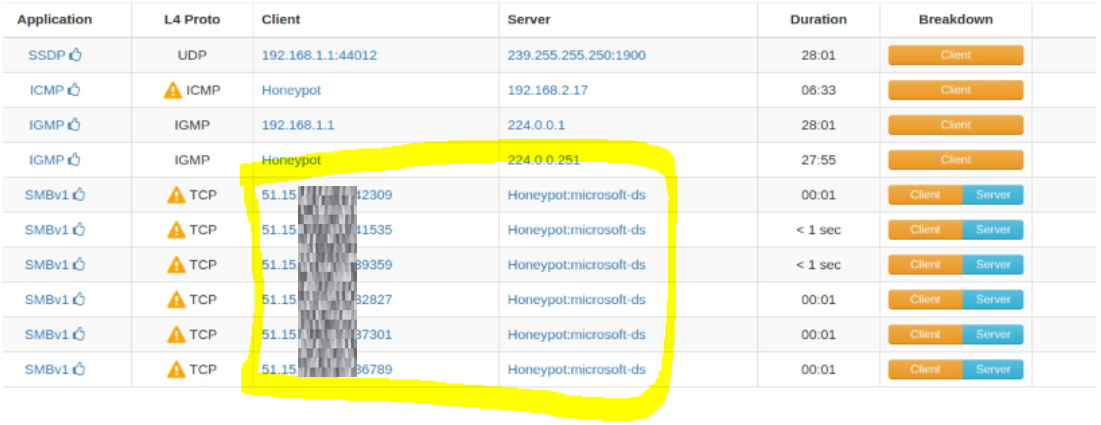
Application	L4 Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes
SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	02:40:02	Client	0 bit/s	2.59 MB
IGMP	IGMP	192.168.1.1	224.0.0.1	02:40:01	Client	0 bit/s	14.12 KB
SSL.GoogleServices	TCP	Honeypot:43682	content-autofill.googlea...:https	05:53	Client Server	0 bit/s	14.62 KB
IGMP	IGMP	Honeypot	224.0.0.251	02:40:04	Client	0 bit/s	10.83 KB
SSL.GoogleServices	TCP	Honeypot:55780	safebrowsing.googleapis....:https	03:07	Client Server	0 bit/s	5.41 KB
MySQL	TCP	51.15 :36199	Honeypot:mysql	00:01	Client Server	0 bit/s	894 Bytes
MySQL	TCP	51.15 :34681	Honeypot:mysql	00:01	Client Server	0 bit/s	889 Bytes
MySQL	TCP	51.15 :38255	Honeypot:mysql	00:01	Client Server	0 bit/s	889 Bytes
MySQL	TCP	51.15 :46705	Honeypot:mysql	00:01	Client Server	0 bit/s	884 Bytes
MySQL	TCP	51.15 :33375	Honeypot:mysql	00:01	Client Server	0 bit/s	869 Bytes

Figura 74: Escaneo Nmap MYSQL. NTopng.



Application	L4 Proto	Client	Server	Duration	Breakdown
SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	02:41:32	Client
SSL.GoogleServices	TCP	Honeypot:43682	content-autofill.googlea...:https	07:24	Client Server
IGMP	IGMP	192.168.1.1	224.0.0.1	02:41:21	Client
IGMP	IGMP	Honeypot	224.0.0.251	02:41:18	Client
SSL.GoogleServices	TCP	Honeypot:56472	safebrowsing.googleapis....:https	< 1 sec	Client Server
MsSQL-TDS	TCP	51.15. 41769	Honeypot:ms-sql-s	< 1 sec	Client Server
MsSQL-TDS	TCP	51.15. 43633	Honeypot:ms-sql-s	< 1 sec	Client Server
MsSQL-TDS	TCP	51.15. 44127	Honeypot:ms-sql-s	00:01	Client Server
MsSQL-TDS	TCP	51.15. 39413	Honeypot:ms-sql-s	00:01	Client Server
MsSQL-TDS	TCP	51.15. 34345	Honeypot:ms-sql-s	< 1 sec	Client Server

Figura 75: Escaneo Nmap SQL. NTopng.



Application	L4 Proto	Client	Server	Duration	Breakdown
SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	28:01	Client
ICMP	ICMP	Honeypot	192.168.2.17	06:33	Client
IGMP	IGMP	192.168.1.1	224.0.0.1	28:01	Client
IGMP	IGMP	Honeypot	224.0.0.251	27:55	Client
SMBv1	TCP	51.15 42309	Honeypot:microsoft-ds	00:01	Client Server
SMBv1	TCP	51.15 41535	Honeypot:microsoft-ds	< 1 sec	Client Server
SMBv1	TCP	51.15 39359	Honeypot:microsoft-ds	< 1 sec	Client Server
SMBv1	TCP	51.15 32827	Honeypot:microsoft-ds	00:01	Client Server
SMBv1	TCP	51.15 37301	Honeypot:microsoft-ds	00:01	Client Server
SMBv1	TCP	51.15 36789	Honeypot:microsoft-ds	00:01	Client Server

Figura 76: Escaneo Nmap smb. NTopng.

También podemos ver las estadísticas de la cantidad de tráfico recibido al puerto 445 SMB sin embargo, no hubo pérdida de anonimato.

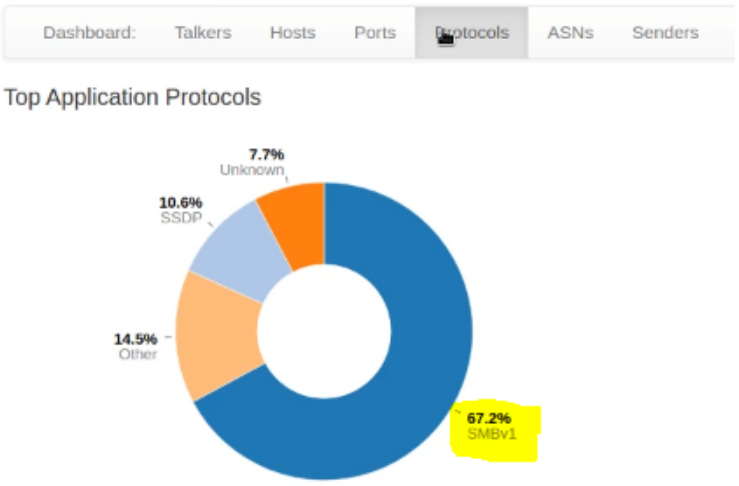


Figura 77: Estadísticas por protocolo SMB. NTopng.

Al lanzar el escaneo con **Nikto**, vemos que la IP reflejada en todo momento es la proporcionada por Tor. A la derecha, en Info, podemos ver que está ejecutando parámetros en el navegador para detectar vulnerabilidades web como la encontrada XSS.

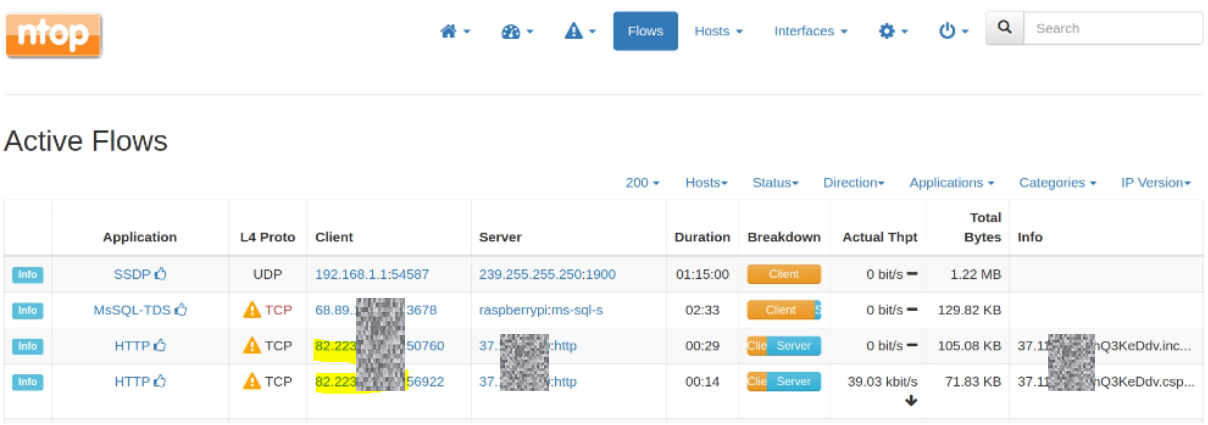


Figura 78: Escaneo http Nikto, NTopng.

También aparece el escaneo lanzado al servicio HTTPS.

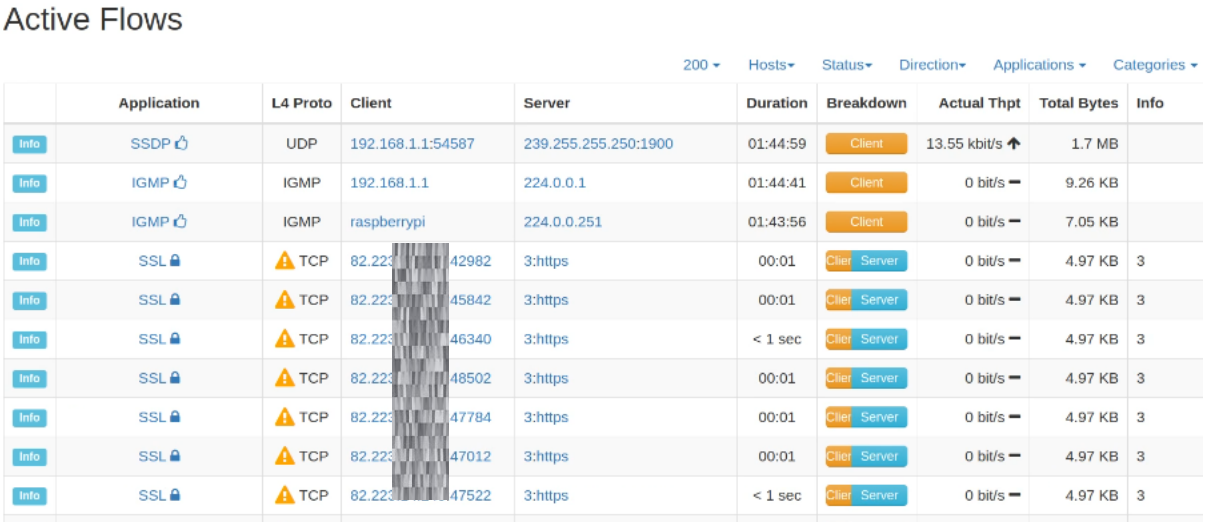


Figura 79: Escaneo https con Nikto, NTopng.

Hasta ahora todo está anonimizado. Vamos a ver qué sucede con el escaneo de **Rapidscan**, ya que esta herramienta utiliza a su vez otras herramientas y podemos perder un poco más el control de los escaneos usados.

Active Flows

200 ▾ Hosts▾ Status▾ Direction▾ Applications ▾ Categories ▾ IP Version▾

	Application	L4 Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info
Info	SSDP	UDP	192.168.1.1:54587	239.255.255.250:1900	09:33:26	Client	0 bit/s	9.28 MB	
Info	HTTP	TCP	82.22[redacted]:57942	37.11[redacted]:http	00:18	Client Server	0 bit/s	83.85 KB	37.1[redacted]/protectedpag...
Info	IGMP	IGMP	192.168.1.1	224.0.0.1	09:32:44	Client	0 bit/s	50.39 KB	
Info	HTTP	TCP	82.22[redacted]:35446	37.11[redacted]:http	00:18	Client Server	41.16 kbit/s	90.0 KB	37.1[redacted]/webcalendar/...
Info	IGMP	IGMP	raspberrypi	224.0.0.251	09:32:36	Client	0 bit/s	38.63 KB	
Info	HTTP	TCP	82.22[redacted]:34102	37[redacted]:http	00:01	Client Server	0 bit/s	8.24 KB	37.1[redacted]/rubrique.asp...
Info	SSL.GoogleServices	TCP	raspberrypi:47674	content-autofill.googlea...:https	06:20	Client Server	2.94 kbit/s	9.26 KB	content-autofill.googlea...

Figura 80: Escaneo Rapidscan. ntopng.

Tampoco se ven reflejados indicios de pérdida de anonimato. Vamos ahora con la parte de escaneo con **Armitage**.

Al utilizar la herramienta de escaneo de Armitage, vemos que realizó un escaneo de puertos TCP.

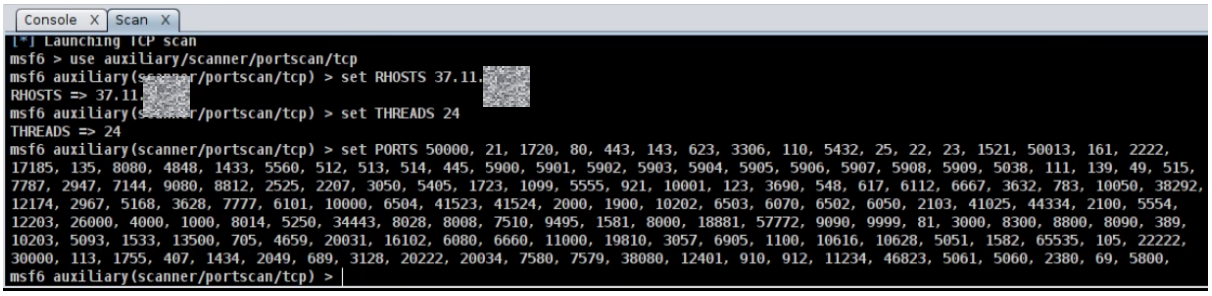


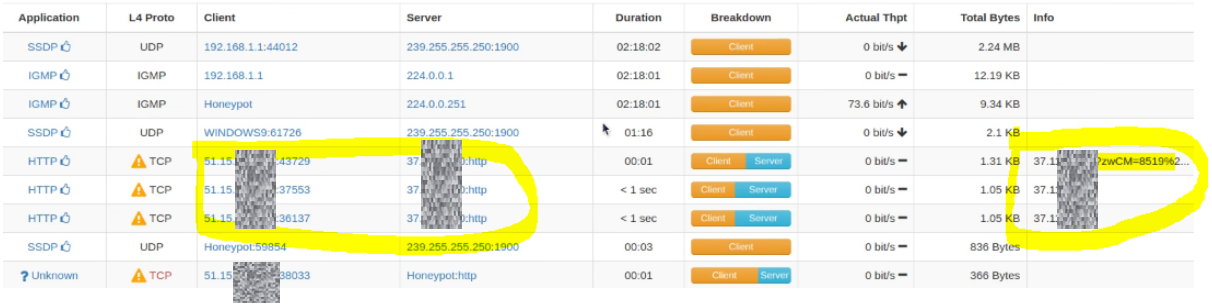
Figura 81: Escaneo de puertos Armitage

Como vemos en la siguiente figura, en este caso tampoco hubo pérdida de anonimato.

	Application	L4 Proto	Client	Server	Duration	Breakdown
Info	SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	05:55:39	Client
Info	IGMP	IGMP	192.168.1.1	224.0.0.1	05:56:03	Client
Info	IGMP	IGMP	Honeypot	224.0.0.251	05:55:18	Client
Info	SSL.GoogleServices	TCP	Honeypot:41188	content-autofill.googlea...:https	05:49	Client Server
Info	SMBv1	TCP	138.121[redacted]:61602	Honeypot:microsoft-ds	00:33	Client Server
Info	SSDP	UDP	Honeypot:50678	239.255.255.250:1900	00:03	Client
Info	MySQL	TCP	51.210.[redacted]:42440	Honeypot:mysql	< 1 sec	Client Server
Info	Unknown	TCP	51.210.[redacted]:10238	Honeypot:sip	< 1 sec	Client Server
Info	Unknown	TCP	51.210.[redacted]:38286	Honeypot:sip-tls	< 1 sec	Client Server
Info	Unknown	TCP	51.210.[redacted]:37520	Honeypot:3000	< 1 sec	Client Server

Figura 82: Escaneo con Armitage

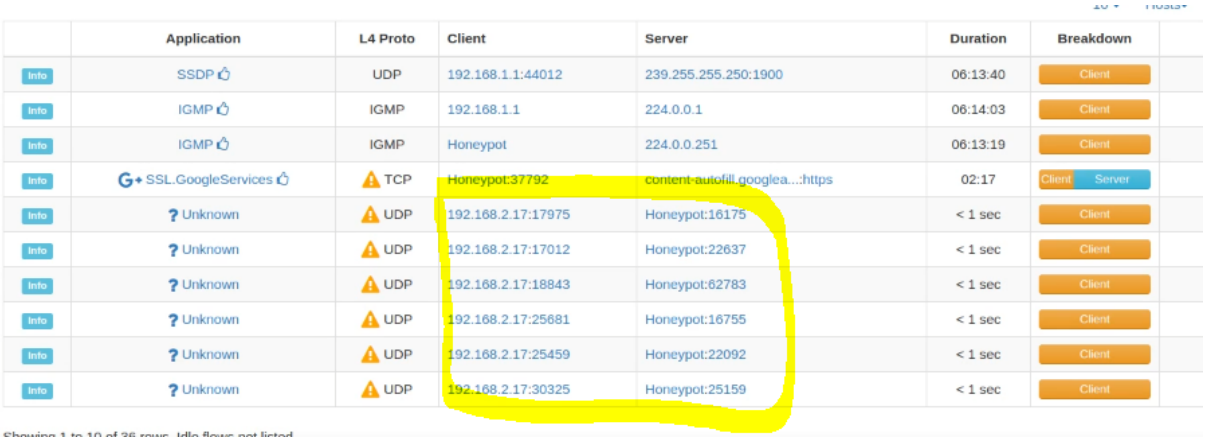
El resto de exploits lanzados como al servicio Apache HTTP, o los de SQL o MYSQL, tampoco rompieron el anonimato.



Application	L4 Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info
SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	02:18:02	Client	0 bit/s	2.24 MB	
IGMP	IGMP	192.168.1.1	224.0.0.1	02:18:01	Client	0 bit/s	12.19 KB	
IGMP	IGMP	Honeypot	224.0.0.251	02:18:01	Client	73.6 bit/s	9.34 KB	
SSDP	UDP	WINDOWS9:61726	239.255.255.250:1900	01:16	Client	0 bit/s	2.1 KB	
HTTP	TCP	51.15 .43729	37 :http	00:01	Client Server	0 bit/s	1.31 KB	37.1 zwCM=8519%2...
HTTP	TCP	51.15 .37553	37 :http	< 1 sec	Client Server	0 bit/s	1.05 KB	37.1
HTTP	TCP	51.15 .36137	37 :http	< 1 sec	Client Server	0 bit/s	1.05 KB	37.1
SSDP	UDP	Honeypot:59854	239.255.255.250:1900	00:03	Client	0 bit/s	836 Bytes	
Unknown	TCP	51.15 .38033	Honeypot:http	00:01	Client Server	0 bit/s	366 Bytes	

Figura 83: Armitage ataque HTTP

Sin embargo, una de las pruebas de lanzar un exploit TFTP con Armitage, al usar éste el protocolo UDP, si que envió nuestra dirección sin enmascarar a través de la red Interna 192.168.2.0/24 a la 192.168.1.0/24. En la siguiente captura vemos la ip 192.168.2.17 como origen. Como las pruebas se realizaron mediante la red interna doméstica conectada a Internet, sí que fue capaz de llegar al destino. Posteriormente se realizó el mismo test de intrusión usando el Modem 3G con el exploit de TFTP pero no se logró, ya que Tor no enruta a través de UDP con lo que, de nuevo, el dispositivo Kali seguía siendo anónimo.



	Application	L4 Proto	Client	Server	Duration	Breakdown
Info	SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	06:13:40	Client
Info	IGMP	IGMP	192.168.1.1	224.0.0.1	06:14:03	Client
Info	IGMP	IGMP	Honeypot	224.0.0.251	06:13:19	Client
Info	SSL.GoogleServices	TCP	Honeypot:37792	content-autofill.googlea...:https	02:17	Client Server
Info	Unknown	UDP	192.168.2.17:17975	Honeypot:16175	< 1 sec	Client
Info	Unknown	UDP	192.168.2.17:17012	Honeypot:22637	< 1 sec	Client
Info	Unknown	UDP	192.168.2.17:18843	Honeypot:62783	< 1 sec	Client
Info	Unknown	UDP	192.168.2.17:25681	Honeypot:16755	< 1 sec	Client
Info	Unknown	UDP	192.168.2.17:25459	Honeypot:22092	< 1 sec	Client
Info	Unknown	UDP	192.168.2.17:30325	Honeypot:25159	< 1 sec	Client

Figura 84: Escaneo UDP recogido por la red interna

Podemos ver más datos recogidos por Ntopng en el Anexo D.

Por último, también hemos utilizado la herramienta Wireshark en algunos casos para ver más en detalle la red y para poder guardar el tráfico en un fichero. Con la herramienta Armitage, lanzamos los ataques vistos anteriormente y recogemos y revisamos el tráfico de red con dicha herramienta.

En la siguiente figura, podemos ver distintos accesos desde varias IPs a los puertos usados por los exploits de Armitage. Tráfico como TELNET o MySQL aparecen junto con la trama completa.

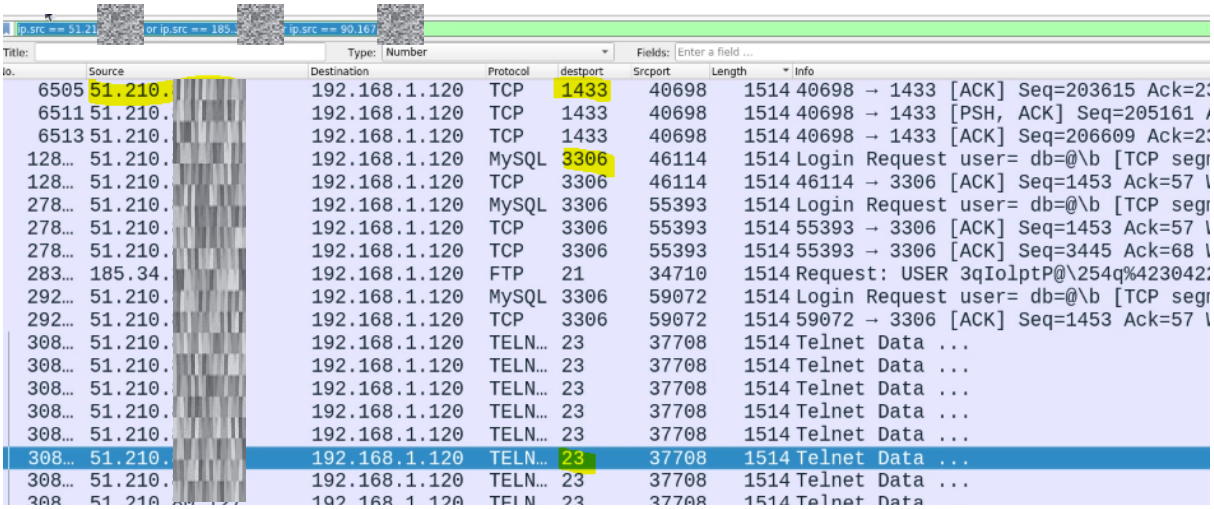


Figura 85: Captura Wireshark varios puertos

Si abrimos la trama del ataque lanzado a Apache, podemos ver en detalle el exploit que se envió a la víctima.

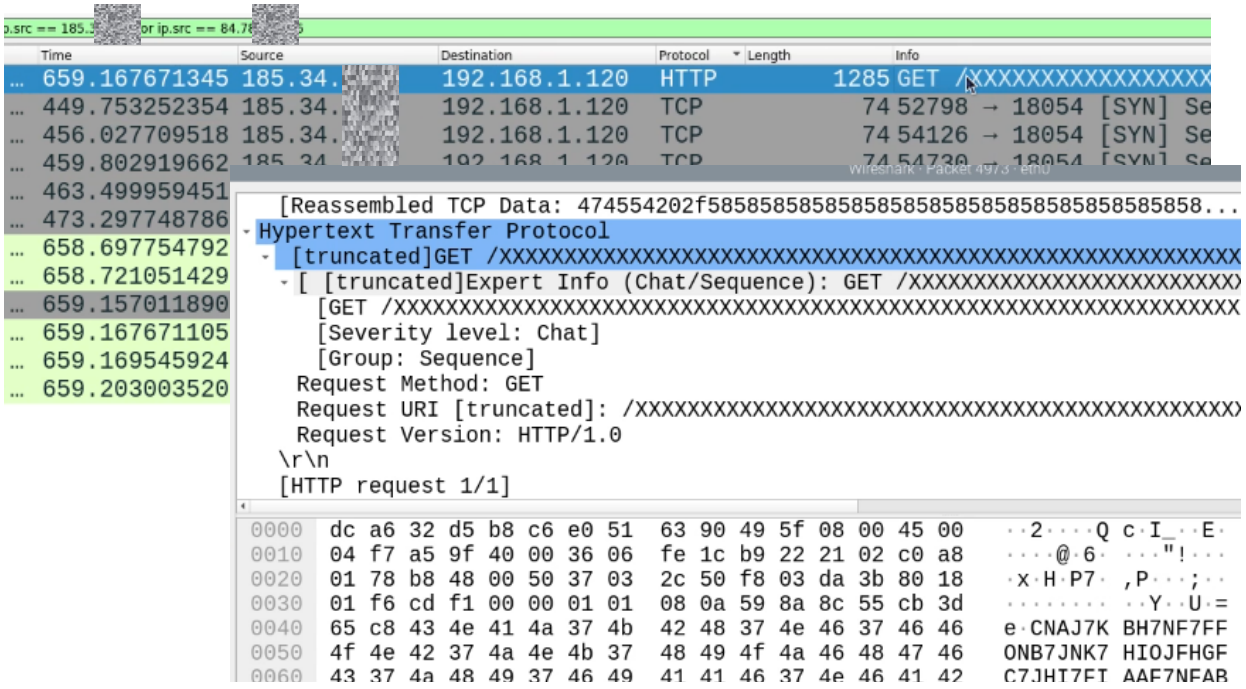


Figura 86: Captura Wireshark trama en detalle parte aplicación

En otra trama, podemos ver el intento de login lanzado por Armitage:

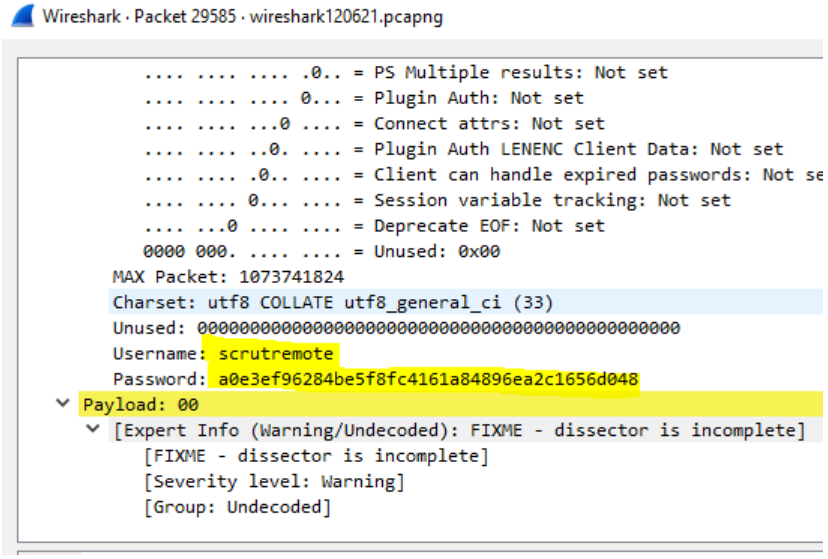


Figura 87: Captura Wireshark ataque scrutremote

Analizando varios de esos ataques tampoco encontramos pruebas de des-anonimización de algún tipo.

6. Planificación y presupuesto

6.1 Planificación del Trabajo.

A continuación, presentamos nuestro diagrama de Gantt en el que indicamos las fechas aproximadas de inicio y fin de los distintos hitos importantes del proyecto. En algunos casos se solapan y por ejemplo, la investigación y documentación se realizó a lo largo del mismo periodo, pero para no mezclarlo, se han separado. Dentro del apartado de investigación se incluye el análisis de la red Tor y la red Surface/Dark web.

Nombre	Fecha de inicio	Fecha de fin
Estudio de viabilidad	1/03/21	5/03/21
Compra y montaje de Material	5/03/21	11/03/21
Instalación imágenes	15/03/21	19/03/21
Configurar Imagen Router Tor	22/03/21	9/04/21
Configurar Imagen HoneyPot	9/04/21	16/04/21
Configurar Imagen Kali	19/04/21	19/04/21
Batería de pruebas	20/04/21	12/05/21
Investigación y documentación	14/05/21	10/06/21
Presentación	11/06/21	15/06/21

Figura 88: Fechas diagrama de Gantt

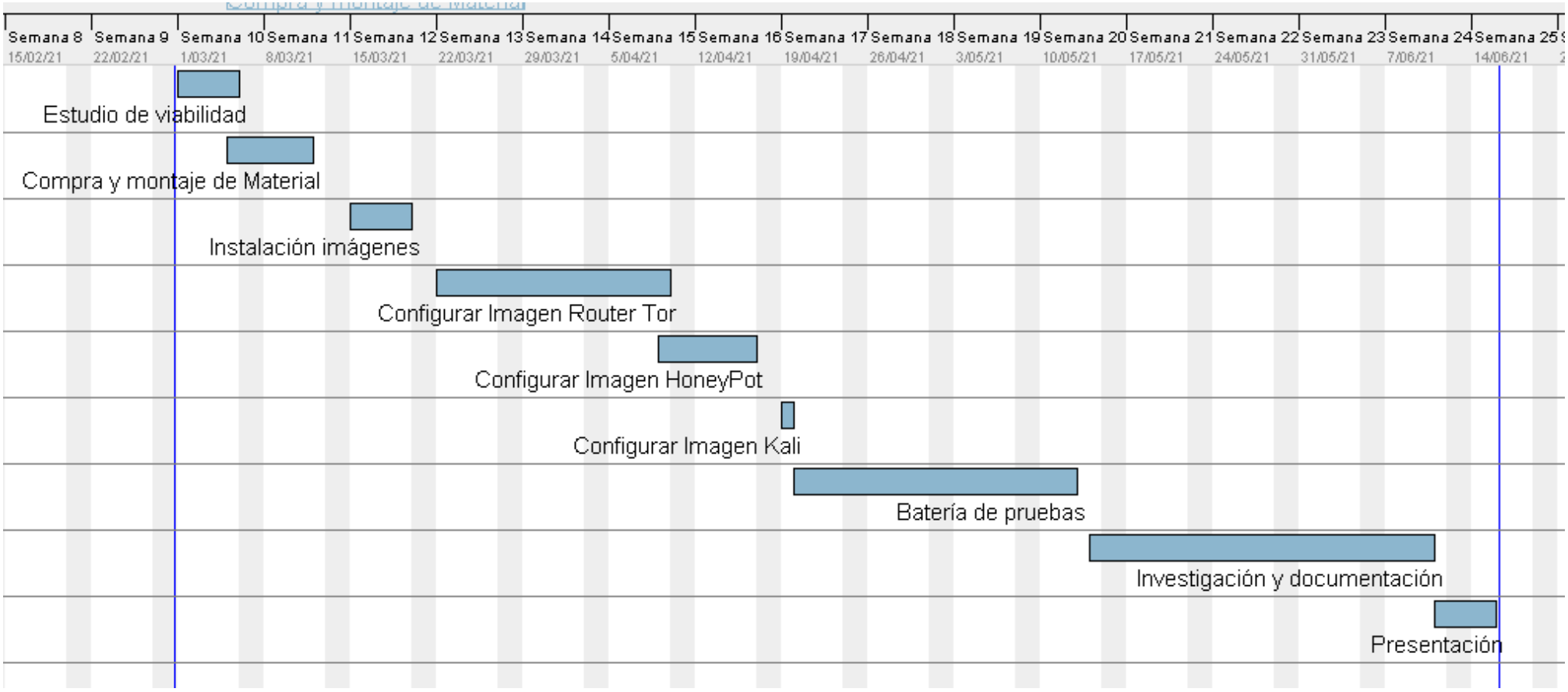


Figura 89: Planificación del proyecto

6.2 Inventario de hardware/software y costes.

En la siguiente tabla podemos ver el inventario de hardware y presupuesto del proyecto.

Componente	Precio en €	Características	Finalidad
Raspberry Pi 4	80	Model B 8 GB Ram	Honeypot, monitorización y análisis
Raspberry Pi 4	30	Model B 2 GB Ram	Enrutador anonimizador
Raspberry Pi 3 Carcasa y disipador incluido	20	Model B 1 GB Ram	Pentester
Modem Huawei	5	220 2G/3G	Salida a Internet 3G
Carcasa Raspberry Cargador USB C Disipadores	20	Bruphny	Protector y cargador Honeypot
Carcasa Raspberry Disipadores	5	Carcasa de ABS	Protector enrutador anonimizador
Tarjeta micro SD	12	SanDisk clase 10 64 GB.C 10	Para RPI Enrutador anonimizador
Tarjeta micro SD	11	Kingston clase 10 64 GB.C 10	Para RPI Honeypot, monitorización y análisis
Tarjeta micro SD	9	Philips clase 10 32 GB.C 10	Para RPI Pentester
Tarjeta micro SD	5	Transcend 16 GB.C 10	Para pruebas
Tarjeta micro SD	Cedido	Scandisk 8 GB	Para pruebas
Cargador USB C	Cedido	Reutilizado de un móvil	Para RPI Enrutador anonimizador
Cargador USB B	4	Cargador Rasp. Pi 3B. 2,5 A.	Para RPI Pentester
Teclado + ratón inalámbrico	15	Logitech bluetooth	
Tarjeta datos 4G prepago	Propia		
Regleta de enchufes + Carga USB	15€		
Total	**231**€		

Tabla 3: Inventario y presupuesto

En la siguiente tabla podemos ver el software utilizado:

Nombre	Características	Rol	Versión
Raspberry			
Debian	Raspberry Pi OS Lite (no desktop)	Router Tor	v2021.03.04 armv7l
Debian	Raspberry Pi OS	Honeypot	v2021.03.04 armv7l
Debian	Kali 4.19.127-Re4son-v7+	Pentester	v2021.1 armv7l
Simplescreenrecorder	Graba pantalla en distintos formatos	Grabar proceso	v0.4.2
VSDC	Editar vídeo	Editar vídeo	v6.6.7.275
Raspberry Pi Imager	Graba y borra MicroSD	Grabar imágenes MicroSD	v1.6
Diagrams.net	Diseño de red	Crear gráficos de red	14.6.13
Honeypot			
NTopng	Captura todo el tráfico y lo muestra de manera gráfica.	Honeypot: Recoger tráfico	v3.8.190204
Dionaea	Emula servicios como Honeypot de todo tipo. Los más comunes.	Honeypot: Emular servicios	v0.11
Pentester			
Nikto	Herramienta de escaneo de vulnerabilidades Web.	Pentester: Vulnerabilidades Web	v2.1.6
Rapidscan	Más de 20 herramientas: DnsMap, Golismero, Nmap, Wapiti, Xsser, etc.	Pentester: Escaneo automático Vulnerabilidades	v1.0
Nmap	Herramienta de escaneo de red.	Pentester: Escaneo de puertos	v7.91
OpenVas	Herramienta de escaneo de vulnerabilidades	Pentester: Escaneo automático Vulnerabilidades	v20.8.1
Router			

Tor	Herramienta que permite anonimato y conexión a la red Tor.	Router Tor: Anonimato	v0.2.5.2-alpha
Hostapd	Crea punto wifi.	Router Tor: Punto wifi	v2.8-devel
Dnsmasq	Permite dar servicio DHCP, DNS, etc.	Router Tor: DHCP	v2.80
WVdial	Herramienta de conexión a Internet usando nuestro modem con comandos AT.	Router Tor: Modem 3G	v1.61
PPPD	Herramienta para establecer conexión punto a punto.	Router Tor: Modem 3G	v2.4.7
Eschalot	Generar dominio .onion		v1.2.0

Tabla 4: Inventario de software

En la siguiente tabla podemos ver las horas empleadas en el trabajo realizado. Las fases de análisis, creación de memoria y creación del router fueron las que más tiempo nos ocuparon. Una vez realizadas, la reproducción y creación de otros dispositivos nos ocupa mucho menos tiempo, lo que abarataría los costes en el caso de volver a generar los productos.

Descripción tarea	Horas	Total servicio
Fase de análisis herramientas	55	900€
Fase montaje Raspberry	2	40€
Fase documentación	60	800€
Fase configuración Router	50	900€
Fase configuración Pentester	10	60€
Fase configuración Honeypot	15	300€
Fase configuración enrutamiento	8	100€
Fase de laboratorios	40	800€
Fase de creación de memoria y vídeos	60	1.200€
Total	**300**	6.000€
Precio x hora	20€	

Tabla 5: Inventario de tareas

El coste total de ejecución del presente trabajo, incluyendo costes materiales y humanos, asciende a 6.231€.

7. Conclusiones y líneas futuras

7.1 Conclusiones.

Durante la realización de este trabajo se han conseguido los objetivos inicialmente planteados. El estudio del funcionamiento de la red Tor, la creación de los dispositivos y la ejecución de los laboratorios ha sido un trabajo enriquecedor y nos ha permitido llegar a los siguientes resultados:

- Hemos logrado nuestro objetivo principal de crear un dispositivo de bolsillo capaz de proporcionar acceso a la red Tor mediante conexión wifi sin utilizar herramientas automáticas ni distribuciones ya preconfiguradas.
- Hemos conseguido nuestros objetivos secundarios de configurar un dispositivo de bolsillo capaz de realizar auditorías de pentesting, otro con el rol de Honeypot y monitorización de red, este último también sin usar herramientas automáticas o imágenes preconfiguradas.
- También hemos logrado analizar la red Tor, tanto a nivel de uso como de funcionamiento de los circuitos Out-proxy e In-proxy.
- Aprovechando la infraestructura creada y mediante los laboratorios, hemos conseguido demostrar que a nivel de paquete de red (IP), no hemos sido capaces de des-anonimizar la conexión realizada con la red Tor a través de nuestro dispositivo.
- Además de eso y gracias a la infraestructura creada, hemos realizado parte de las tareas de hacking ético (fase de escaneo y acceso) y de monitorización actuando un dispositivo como "*Red Team*" y otro como "*Blue Team*" respectivamente para confirmar el anonimato.

Por otro lado, durante el proceso de ejecución de este trabajo se han presentado algunas situaciones que nos han llevado a las siguientes conclusiones, pasamos a comentarlas en el siguiente orden: primero relacionadas con la Raspberry Pi y luego con la red Tor.:

- Aunque hemos podido demostrar que en ninguno de los escaneos y ataques ha enviado la IP original y siempre enviaba la IP proporcionada por Tor, no se descarta que a nivel de aplicación se estén enviando datos significativos del usuario, ya sea por cookies al navegar o con señuelos que redirija a otras

páginas. En este sentido, nos hubiera gustado ahondar más a fondo en este aspecto, incluso crear Hidden services e implementar señuelos.

- Los dispositivos Raspberry Pi son más potentes y funcionales de lo que en un principio se esperaba. Capaces de ejecutar gran cantidad de herramientas ya compiladas para chips ARM.
- También hemos comprobado la gran diferencia de rendimiento de la Raspberry Pi 3 a la Raspberry Pi 4. La primera falló en varias ocasiones al cargar herramientas consumidoras de gran cantidad de memoria. A parte de eso, al ser la única sin ventilador en los disipadores tuvimos problemas de calentamiento llegando a superar los 80º en algunas ocasiones.
- Con respecto a la compra del material para este trabajo, se intentó conseguir de segunda mano, pero no hay tanto mercado y el precio de los vendedores es muy similar al de venta en tienda con la ventaja de que esta última tiene garantía y tiene las últimas versiones de hardware.
- Algunas de las herramientas de Pentesting o del dispositivo Honeypot no hemos sido capaces de hacerlas funcionar, ya sea por falta de recursos o por incompatibilidad.
- Mediante las fases de escaneo y ataque, hemos podido comprobar que existen dos perfiles en "*Red team*" a la hora de realizar Pentesting. Por un lado, hemos configurado escaneos adaptados a la necesidad de cada servicio de forma granulada, pero al mismo tiempo hemos usado herramientas automatizadas que permiten a cualquier persona, sin conocimientos de ningún tipo, realizar ataques a cualquier infraestructura con el peligro que ello implica.
- Lo mismo que en el punto anterior sucedió con la configuración del router y el Honeypot. A pesar que en el mercado se encontraron muchas herramientas ya preconfiguradas o automatizadas, se optó por herramientas configuradas manualmente para aprender más sobre su funcionamiento.
- Aunque todo el software utilizado es libre, después de analizar la red Tor, planteamos como líneas futuras de investigación crear este mismo dispositivo router, pero con un software VPN de pago para mayor confidencialidad y sobretodo confianza.
- Durante la investigación de la red Tor hemos comprobado que es un tema tan extenso a tratar que se podría haber dividido en varios trabajos.

- La parte de la navegación In-proxy es muy compleja y nos ha abierto varias líneas de investigación, una de ellas que nos interesaba realizar era crear un Hidden service publicado en la red Tor y realizar pruebas Pentesting contra el mismo.
- Hemos detectado que Tor cambia de nodo de salida cuando el tráfico es elevado o cuando se intenta acceder a servicios que el nodo en el que estamos actualmente no es capaz de proporcionar. Eso nos ha demostrado que los nodos de salida filtran el tráfico.
- Aunque Tor es una red distribuida muy segura, el hecho de que haya 9 nodos autoritativos nos demuestra que el ente que los controle (para bien o para mal) si está administrando la red Tor, ya que decide los nodos de salida y de entrada, indispensables para proporcionar el anonimato.
- Como puntos negativos de Tor, hemos detectado que la velocidad es excesivamente lenta, que muchos de los enlaces .onion están rotos y que hay páginas de enlaces que carecen de mantenimiento y no los actualizan. También, que existen innumerables Dark webs de venta de artículos de dudosa confianza, u otras páginas que pueden ser de carácter ilícito como venta de drogas, armas, ciberterrorismo, hacking, venta de malware, venta de datos, falsificación de documentos, pornografía ilegal, etc.
- Al navegar por la Surface web, muchas de las páginas muestran su desconfianza al detectar que nuestra IP viene de la red Tor e imponen captcha u otras medidas de control de acceso para evitar ataques. Al haber un listado público cada hora de las IPs de los nodos de salida esto es posible.
- Noticias sobre que el 25% del tráfico de salida de Tor está filtrado, de que un grupo de hackers se hizo con más del 10% de los nodos de salida y de que el FBI consiguió destapar y cerrar varios servidores Hidden services no ayudan tampoco a generar confianza con respecto a dicha red.
- Mediante este trabajo, no pretendemos juzgar la red Tor, simplemente dejar claro lo que nos podemos encontrar en ella y ya cada individuo es libre de entrar o no según sus necesidades o su curiosidad, pero es bueno que las personas tengan ese conocimiento antes de acceder. Añadir que es una herramienta que fue premiada en su día, que es capaz de salvar vidas al dar capacidad de expresión en países con fuertes medidas de censura, pero que

también produce cierta desconfianza y es que, para bien o para mal, permite hacer precisamente eso, tanto el bien como el mal sin poder ser juzgado por los actos cometidos.

7.2 Líneas futuras.

Durante el desarrollo de este trabajo, hemos visto la gran cantidad de posibilidades para futuros proyectos. Algunas de ellas las hemos podido realizar en el trabajo, sin embargo, otras nos ocupan un tiempo superior al disponible para este TFM. Por si pudieran servir para que realizáramos futuros proyectos o para la comunidad, se procede a nombrar las posibles líneas futuras de investigación.

- Raspberry Pi con Nagios. Se trata de convertir nuestra Raspberry Pi en un servidor de monitorización usando el famoso software Nagios. (Emmet y Pimylife)
- Módulo GPIO con Leds que avise cuando hay anonimato. Mediante un módulo protoboard, realizar las conexiones a la GPIO y configurar leds e interruptores y/o pulsadores para ir cambiando entre anonimato y no anonimato y avisar si está activo o no activo.
- Módulo 4G conectado directamente a la Raspberry Pi. En vez de usar un módem externo, se trata de adquirir uno para integrar con la Raspberry, acoplarlo a nuestro dispositivo y de esa manera tener Internet sin cables. (Estremadoyro y Altronics)
- Honeypot web de gran interactividad, por ejemplo, un Damn Vulnerable Linux (DVL) y Damn Vulnerable Web Application (DVWA). Mediante cookies, intentar obtener información del atacante. (Null byte)
- Honeypot de servicios industriales Scada.
- Realizar un ataque al Honeypot Dionaea mediante un exploit configurado con Msfvenom para ver si es capaz de recogerlo como indican en sus especificaciones.
- Pentesting In-proxy de Tor. Crear un servidor Honeypot dentro de la red Tor para recoger los ataques recibidos y realizar Pentesting al mismo. (Echeverri y Incibe) (Rodríguez et al.)

- Anonimato con otras herramientas similares a Tor como pueden ser: Freenet, i2p, ZeroNet, etc. (Isaac)
- Configurar Raspberry Pi junto con VPNs comerciales para proporcionar anonimato más seguro y confiable. Utilizar VPN en conjunto con Tor, Freenet, etc, para mayor anonimato. (Gaitatzis)
- Configurar el Honeypot en AWS o con Dockers para que sea un entorno más potente y real.
- Desgranar más a fondo el protocolo Tor, recogiendo todo el proceso y capturando el tráfico con Wireshark, viendo los ficheros de almacenamiento donde se encuentran los nodos "*consensus*" y donde aparecen los nodos guarda, sus IPs, certificados, etc. Analizar el tráfico con Wireshark quizá nos permita des-anonimizar a través de los datos de la capa de aplicación.

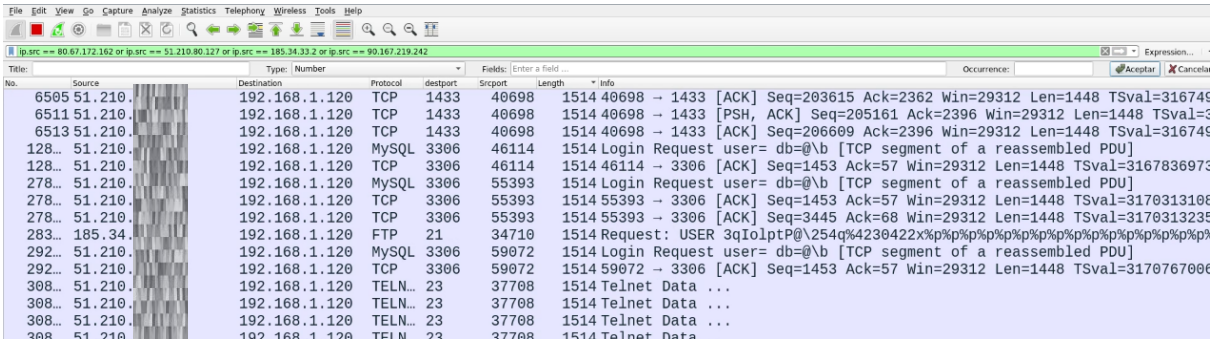


Figura 90: Captura Wireshark varios puertos

Anexos

Anexo A: Características y montaje de Raspberry PI.

En este anexo podemos ver las especificaciones técnicas de las Raspberry Pi 3 y 4.

Especificaciones.

Las siguientes son las especificaciones de la Raspberry Pi 4.

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

Las siguientes son las especificaciones de la Raspberry Pi 3B.

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU

- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

En los siguientes vídeos, podemos ver el montaje físico de las dos Raspberry Pi 4. Se les pone disipador y ventilador junto con la caja protectora.

Disipador con ventilador Bruphny.

https://www.youtube.com/watch?v=lLIwTMNcMwA

Disipador con ventilador genérico.

https://www.youtube.com/watch?v=AhFTWKE845s

Instalación de imágenes.

Descarga de imágenes de Raspbian, Kali y Honeypot.

Raspbian: La imagen de Raspbian viene en el instalador de imágenes "*Raspberry Pi Imager v1.6*".

Kali: La imagen de Kali debemos descargarla de la página web de Offensive Security. (Kali org)

Honeypot: Existen varias opciones, instalar una imagen de Raspbian, Ubuntu, etc y sobre ella instalar el Honeypot y herramientas de monitorización. También existe la posibilidad de descargarse una imagen ya preinstalada con el Honeypot.

Pasos para la instalación de las imágenes:

1.- Instalamos el software Raspberry "*Pi Imager*" de la página de Raspberry.

2.- Seleccionamos el sistema operativo, por defecto Raspbian. Pulsamos en "*Choose storage*" y seleccionamos la tarjeta SD en la que queramos instalar la imagen.

3.- Una vez seleccionada la tarjeta SD, se nos habilitará el botón de "*Write*". Si queremos instalar Raspbian pulsamos en el que viene con Imager.

4.- Nos avisará que se eliminarán todos los datos de la tarjeta SD.

5.- Cuando finalice la escritura, comenzará la verificación.

6.- Cuando termine el proceso completo, nos avisará de que ya podemos retirar la tarjeta.

7.- En el caso de Kali u otra imagen, el proceso es muy similar, lo único que cambia es el principio en el que debemos elegir la imagen correspondiente. Pulsamos en el botón de "*Operating System*". Dentro de las opciones, hay multitud de ellas, según su propósito: emulador de juegos, reproductor de multimedia, etc.

8.- Seleccionamos "*Use custom*" para poder introducir la imagen correspondiente.

9.- Previamente tenemos que descargar el archivo .img. En este caso la imagen de kali Linux ARM para Raspberry.

10.- Una vez terminado el grabado de la imagen, extraemos la tarjeta micro SD y la insertamos en la la parte posterior de nuestra Raspberry Pi, en la ranura MicroSD y la encendemos a través del alimentador eléctrico. Una vez arrancada, nos solicitará cierta configuración que pasamos a describir:

- Configuración de país, idioma y zona horaria.
- Establecer una contraseña.
- Configuración de wifi (no realizar si se va a usar como Punto de acceso wifi o Hotspot, en ese caso conectarla a la salida ethernet).
- Se actualizará sola y pedirá reinicio.
- Cuando termine, ya estará operativa.

11.- Después, en cada imagen instalaremos *XRDP* (excepto la del router) para poder conectarnos a través del servicio de *RDP* de Windows (*MSTSC*), *SSH*

para poder acceder a la Raspberry vía terminal de consola y “*simplescreenrecorder*” (Baert) para documentar el proceso en vídeo.

Desde la consola o terminal, ejecutamos el siguiente comando.

$sudo apt-get install xrdp ssh simplescreenrecorder

12.- Otra opción para consumir menos recursos es la instalación sin entorno gráfico. Esta instalación es la correcta para la parte del enrutamiento, esto se debe a que en principio no se usaría para otro servicio más que para enrutar. Se encuentra en el programa “*Raspberry Pi Imager*”, “*Raspberry Pi SO (Others)*”, luego los demás pasos son los mismos que en las anteriores imágenes.

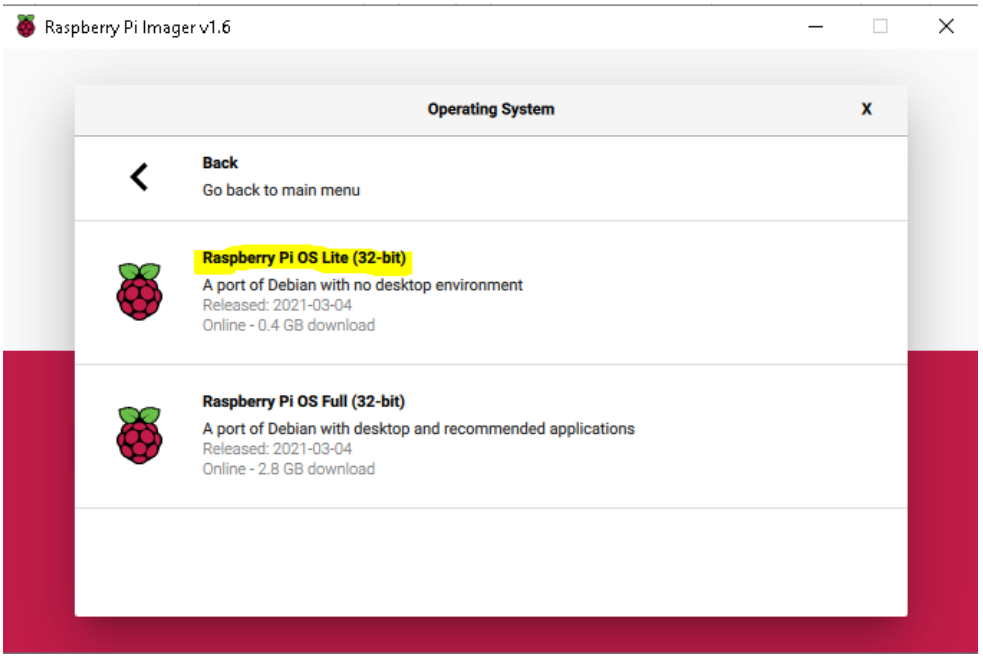


Figura 91: Raspberry Pi Imager OS Lite sin entorno gráfico

En el siguiente vídeo podemos ver una instalación completa desde cero:

https://www.youtube.com/watch?v=vXtE3yBxaK0

Anexo B: Configuración Router.

Instalación y configuración del punto de acceso wifi.

Una vez terminados los pasos anteriores, nuestra Raspberry estará operativa y podemos continuar con la instalación de los demás roles, en este caso el de punto de acceso wifi.

Se han barajado varias opciones, algunas automáticas como "*RaspAP"*, pero como al final muchas de ellas utilizan "*Hostapd"* para el servicio de punto de acceso, "*Dnsmasq"* o "*Isc-dhcp-server"* para el servicio de "*DHCP"*, se ha decidido instalarlas y configurarlas desde cero para tener mayor control.

Enumeramos las herramientas probadas:

Hostapd: Es el servicio que se encarga de proporcionar el punto de acceso. (Alcocer)

Dnsmasq: Es el servicio encargado de proporcionar *DNS* y *DHCP*, nosotros lo usaremos para el *DHCP* de los dispositivos que se conecten al punto de acceso. (Sánchez Alés y Linuxnomicon)

Isc-dhcp-server: Es una alternativa al anterior, nosotros usamos *Dnsmasq* pero podríamos haber elegido Isc-dhcp-server. (Ubuntu help)

RaspAP: Es una herramienta que configura el punto de acceso de manera automática pero optamos por instalar y configurarlo a mano. (Raspap)

El proceso de instalación es el siguiente (Gus y PiMyLifeUp):

- Instalamos las herramientas Hostapd y Dnsmasq.

  $sudo apt-get install hostapd dnsmasq

- Paramos el servicio Dnsmasq para configurarlo:

  $sudo systemctl stop dnsmasq

- Editamos el archivo Dnsmasq para añadir el ámbito DHCP:

  $sudo nano /etc/dnsmasq.conf

  interface=wlan0 # Use interface wlan0

  server=1.1.1.1 # Use Cloudflare DNS

  dhcp-range=192.168.2.10,192.168.2.50,12h # IP range and lease time

- Paramos el servicio hostapd para configurarlo:

  $sudo systemctl stop hostapd

- Editamos el archivo de configuración y le añadimos los parámetros necesarios, entre ellos el SSID y la contraseña.

$sudo nano /etc/hostapd/hostapd.conf

interface=wlan0 → Indica el interfaz a usar como punto de acceso
driver=nl80211 → El tipo de driver que usará
hw_mode=g → El tipo de conexión wifi, a, b o g
channel=6 → El canal del medio a utilizar
ieee80211n=1 → Indica que usará ieee80211n
wmm_enabled=0 → No usará esta opción
macaddr_acl=0 → No usará esta opción
ignore_broadcast_ssid=0 → No usará esta opción
auth_algs=1 → Método de autenticación
wpa=2 → Autenticación WPA 2
wpa_key_mgmt=WPA-PSK → Autenticación WPA-PSK
wpa_pairwise=TKIP → Autenticación TKIP
rsn_pairwise=CCMP
 # Nombre del SSID
*ssid=**Raspator***
 # Contraseña de la wifi
*wpa_passphrase=**RaspaTor2002***

- Editamos el fichero de configuración por defecto y el del servicio para asegurarnos de que usa el que hemos modificado.

$sudo nano /etc/default/hostapd
#DAEMON_CONF=""
Replace with:
DAEMON_CONF="/etc/hostapd/hostapd.conf"

- También hacemos lo mismo en el archivo de servicio.

$sudo nano /etc/init.d/hostapd
DAEMON_CONF=

Lo cambiamos por:

DAEMON_CONF=/etc/hostapd/hostapd.conf

- Editamos el fichero de configuración del demonio DHCP para darle una ip fija a nuestro interfaz wifi, de esa manera hará de router para los dispositivos que se conecten.

  $sudo nano /etc/dhcpcd.conf
  interface wlan0
  static ip_address=192.168.2.1/24
  nohook wpa_supplicant

- Reiniciamos el servicio DHCP.

  $sudo systemctl restart dhcpcd

- Editamos el fichero de configuración *sysctl.conf* para habilitar el enrutamiento entre interfaces. Buscamos la línea net.ipv4…, la descomentamos y añadimos un 1.

  $sudo nano /etc/sysctl.conf
  net.ipv4.ip_forward=1

- El siguiente comando hace lo mismo que lo anterior, pero habilitando el enrutamiento en ese mismo momento, sin reiniciar.

  $sudo echo 1 > /proc/sys/net/ipv4/ip_forward

- El siguiente comando "*"Iptables""* sirve para redireccionar todo el tráfico a la red *eth0* (esto no es necesario más adelante cuando el tráfico se enruta por la red Tor, pero sí, si solo lo queremos como punto de acceso).

  $sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

- Lo siguiente permitirá que la regla recién añadida se mantenga persistente después del reinicio.

  $sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

- Un último paso de configuración será añadir lo siguiente en el fichero "*rc.local*", justo antes de la línea "*exit 0*". De esta manera se ejecutarán nuestras líneas "*Iptables"* en cada inicio.

  $sudo nano /etc/rc.local

  Antes de exit 0 ponemos la siguiente línea:

iptables-restore < /etc/iptables.ipv4.nat

exit 0

- Ya solo nos queda configurar los servicios para que se levanten automáticamente y reiniciar.

  $sudo systemctl unmask hostapd
  $sudo systemctl enable hostapd
  $sudo systemctl start hostapd
  $sudo service dnsmasq start
  $sudo reboot

  Una vez reiniciado, veremos nuestro punto de acceso wifi llamado Raspator.

  En el siguiente vídeo podemos ver el proceso completo:

  https://youtu.be/G6fx-vSDsaQ

Instalación y configuración del modem 3G.

Una vez configurada nuestra Raspberry Pi como punto de acceso, nuestro siguiente objetivo es instalar y configurar el modem 3G. Para ello, hemos realizado varias operaciones hasta dar con la más acertada para nuestro caso.

Existen varias aplicaciones que nos ayudarán a hacer funcionar nuestro modem, algunas automáticas como SAKIS3G y UMTSkeeper que funcionan muy bien y automatizan mucho el proceso, y otras de configuración manual. Como en el caso anterior, para no perder el control, hemos decidido hacerlo solo con las herramientas necesarias y configurándose en base a nuestro objetivo.

Vamos a comentar las herramientas testadas, para que vale cada una y como las hemos usado y configurado.

Las dos siguientes las probamos, pero no las implantamos en este trabajo:

SAKIS3G: Realiza la conexión de forma automática preguntando por el dispositivo, el proveedor de servicios, etc y luego permite cerrar la conexión, todo ello mediante entorno gráfico. (Scott)

UMTSKeeper: Es una herramienta de línea de comandos que permite lanzar la conexión automáticamente y lo más importante, en caso de que se corte, volver a lanzarla. Dentro del paquete de UMTSkeeper viene Sakis3G. (Daladim)

Las siguientes herramientas si son necesarias para este trabajo:

Usb-modeswitch: Este ya viene preinstalado y se encarga de reconocer que modem se conecta. Su principal labor es decirle que no es un dispositivo de almacenamiento sino un modem 3G/4G. Sin este programa, hay muchos modems 3G/4G que no funcionarán en Linux al detectar la partición donde se almacenan los drivers y no detectarlo como dispositivo de comunicaciones. Es transparente al usuario, excepto en algunos casos que se puede modificar para cambiar el driver y que use el de esta aplicación (/etc/usb-modeswitch) en vez de cargar el del sistema. En nuestro caso, no requiere configuración. (Dietze y Draisberghof)

Pppd: Point to point protocol, es un protocolo punto a punto muy utilizado en las antiguas conexiones de módem a través de líneas telefónicas. Lo necesitaremos para que "*wvdial*" realice la conexión punto a punto con el modem 3G a nuestro operador. No requiere configuración. (Ubuntu)

Wvdial (Lachance y Wlach): Es el programa encargado de realizar la conexión con nuestro operador, establecer la velocidad, asignar IPs, etc. Al ser por línea de comandos, nos permite realizar una primera configuración en el fichero */etc/wvdial.conf* para posteriormente arrancar automáticamente de varias maneras, ya sea en el archivo de sistema *rc.d*, en el fichero de arranque el usuario Pi /home/pi/.bashrc (esta fue la opción elegida), o incluso instalarlo como servicio. Para que realice la conexión en nuestro modem 3G, debemos configurar el fichero /etc/wvdial.conf con los siguientes parámetros:

```
pi@Router:~ $ cat /etc/wvdial.conf
[Dialer Defaults]
Stupid mode = 1
Modem = /dev/ttyUSB0
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0
Init3 = ATX3
Init4 = AT+CGDCONT=1,"IP","orangeworld","",0,0
Phone = *99#
Username = orange
Password = orange
New PPPD = yes
pi@Router:~ $
```

Figura 92: Configuración archivo wvdial.conf

- Metemos la línea "*wvdial"* en el archivo del usuario "*pi" ".bashrc"* para que inicie la conexión nada más arrancar la Raspberry.

  $sudo nano /home/pi/.bashrc

- Añadimos la siguiente línea

  sudo wvdial &

- Por último, para que Tor sea capaz de actualizar su salida a Internet, debemos reiniciar su servicio así que, a continuación de la línea anterior "*sudo wvdial*", debemos añadir las dos siguientes líneas, la primera para esperar 30 segundos y la siguiente reinicia el servicio de Tor:

  sleep 30
  sudo service tor restart

Nota: Para que funcione esto último, en la imagen de Raspberry sin entorno gráfico, es necesario modificar dos cosas. Hay que hacer que haya ***autologin*** y que en el archivo "*sudoers*" **no se requiera contraseña al usuario "*pi"***.

Para configurar el autologin, en un terminal de consola, ejecutar "*raspi-config*", seleccionar S5 "*BOOT / AUTO LOGIN*" y B2 "*CONSOLE AUTOLOGIN*", aceptamos dos veces y finalizar. Reiniciará la Raspberry.

Para configurar "*Sudoers*": Editamos "*/etc/sudoers"* y justo debajo de la línea "*%sudo ALL=(ALL:ALL) ALL*", añadir la línea "*%pi ALL=(ALL) NOPASSWD: ALL*"

Podemos ver un vídeo de las herramientas y su configuración en el siguiente enlace:

https://www.youtube.com/watch?v=FRPwkUBaEA8

Podemos ver un vídeo de cómo establecer autologin y ejecutar sin contraseña en el siguiente enlace:

https://www.youtube.com/watch?v=IUX52loPyxY

Instalación y configuración de la aplicación Tor.

La aplicación Tor dispone de un fichero llamado torrc situado en "*/etc/tor*" con multitud de opciones, entre ellas nos permite enrutar casi todo el tráfico a su red. También, tiene opciones para elegir el país de entrada y salida a la red Tor, etc.

Instalación y configuración (Gus y PiMyLifeUp):

- Instalamos Tor.

  $sudo apt-get install tor

- Una vez instalado, tenemos que añadir las siguientes líneas al final del archivo torrc. Estas líneas indican los puertos que estarán a la escucha para enrutar a TOR, tanto transporte como DNS. También el ámbito de direcciones de la red virtual creada para tal efecto y por supuesto la línea de archivado de logs. También añadiremos las dos últimas líneas, la primera para que no nos esté cambiando de circuito cada 10 minutos y la siguiente para que solo se conecte a los nodos de entrada y salida de los países de España, Francia y Portugal.

  $sudo nano /etc/tor/torrc

  Log notice file /var/log/tor/notices.log → *Logs*
  VirtualAddrNetwork 10.192.0.0/10 → *Direccionamiento virtual*
  AutomapHostsSuffixes .onion,.exit → *Mapear los sufijos .onion (inproxy)*
  AutomapHostsOnResolve 1 → *Asigna ip virtual cuando accedemos a sufijos .onion o .exit*
  TransPort 192.168.2.1:9040 → *Puerto de entrada al proxy*

DNSPort 192.168.2.1:53 → *Puerto de resolución DNS*

MaxCircuitDirtiness 3600 → *Tiempo antes de cambiar de circuito. (aún así, nos lo cambió varias veces, no sabemos si por problemas con el nodo de salida)*

EntryNodes {ES},{FR},{PO} → *Establecer nodos de entrada, StrictNodes1 es opcional por si quiere que sea algo estricto*

ExitNodes {ES}{FR}{PO} → *Establecer nodos de salida, StrictNodes1 es opcional por si quiere que sea algo estricto*

- Para asegurarnos de que "*"Iptables"*" está limpio, ejecutamos los siguientes comandos para borrarlas.

  $sudo iptables -F → *Borra las líneas introducidas*

  $sudo iptables -t nat -F → *Borra las entradas pero las de tipo nat*

- Las siguientes líneas sirven para direccionar el tráfico proveniente del interfaz WIFI antes de enviarlas a Internet. Las dos primeras permiten no redireccionar el puerto 22 ni el 53. La última sirve para redireccionar el resto de tráfico TCP a la red Tor. Vemos que no indica interface de salida, de esa manera saldrá por el que esté por defecto en la configuración del comando "*route*", si arrancamos con el cable RJ45 irá por cable y si lo quitamos y arrancamos por modem irá por modem. Si se arranca con los dos irá por RJ45, pero mejor no hacerlo por no tener problemas de enrutamiento.

  $sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j REDIRECT --to-ports 22

  $sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --to-ports 53

  $sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-ports 9040

- Con el siguiente comando hacemos los cambios persistentes de tal manera que, aunque reiniciemos no se borren.

  $sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

- Los siguientes comandos sirven para almacenar los eventos.

  $sudo touch /var/log/tor/notices.log

  $sudo chown debian-tor /var/log/tor/notices.log

  $sudo chmod 644 /var/log/tor/notices.log

- Levantamos el servicio de Tor de forma correcta.

  $sudo service tor start

- El siguiente comando meterá la aplicación de tor en los servicios de arranque para que se inicie de forma automática.

  $sudo update-rc.d tor enable

  Podemos ver un vídeo de las herramientas y su configuración en el siguiente enlace:
  https://www.youtube.com/watch?v=HyLNmvZg6Yc

Configuración del enrutamiento.

En este apartado debemos comentar dos operaciones de enrutamiento. La primera es que las líneas de "*Iptables"* indicarán que todo lo proveniente de la red WLAN se enruta a través de la red TOR. Sin embargo, debemos tener en cuenta que hay dos enlaces para salir a Internet, un Modem 3G o una red ethernet y si se enciende la Raspberry Router con los dos conectados hay que modificarlo según el caso.

Nota: Esto es automático si se arranca con uno (RJ45) u otro (MODEM), pero lo comentamos por si se quiere automatizar el proceso mediante un interruptor o un pulsador (ver apartado líneas futuras) en la propia Raspberry Pi.

Por velocidad y ahorro de datos, la ruta por defecto siempre será la tarjeta de red, pero hay casos en los que no tendremos disponible una línea Ethernet donde conectarnos, en ese caso tendremos que activar la salida a través de nuestro modem.

- Eliminamos la ruta por defecto ETH0

  $sudo route del default eth0

- Ahora añadimos nuestro router 3G como enrutador por defecto.

  $sudo route add default ppp0

- En el caso de que nos conectemos por modem pero más tarde disponemos de una conexión ETH0, tendremos que repetir el paso, pero eliminando el router por defecto ppp0 con la salvedad que ahora tendremos que especificarle la ruta como "*gateway"*. Esto es así porque se queda guardado/cacheado y tarda mucho tiempo en refrescarse automáticamente. Otra opción más sencilla es apagarla y encenderla con el RJ45 conectado y el modem desconectado.

  $sudo route del default ppp0
  $sudo route add default gw 192.168.1.1 eth0

Recordatorio: Por defecto y para automatizar el proceso, si la Raspberry Router arranca con el modem 3G sin conexión Ethernet el tráfico de todos los dispositivos que se conecten a la wifi Raspator saldrán por el mismo. Si por el contrario arrancamos la Raspberry Router sin el modem y conectado a una toma de red Ethernet con DHCP, todo el tráfico saldrá por Ethernet, de esa manera evitamos entrar en configuraciones y solo tenemos que arrancarlo, conectarnos al punto wifi que produce y ya estaremos conectados a la red Tor.

La segunda operación debemos realizarla en nuestro router doméstico, para ubicar ahí nuestro Honeypot/NSM/SIEM.

Debemos configurar la IP del Honeypot en la DMZ y entrar en el apartado de "*port forwarding*" (NAT/PAT) para redireccionar todas las peticiones a la IP interna correspondiente.

La configuración de la DMZ en el router Livebox fibra del fabricante Arcadyan, modelo PRV3399B_B_LT es la siguiente:

- Primero reservamos la Ip en nuestro DHCP entrando en "*Configuración avanzada*", "*Configuración de la red*", "*DHCP*" y añadimos ahí nuestra IP a la MAC del Honeypot.

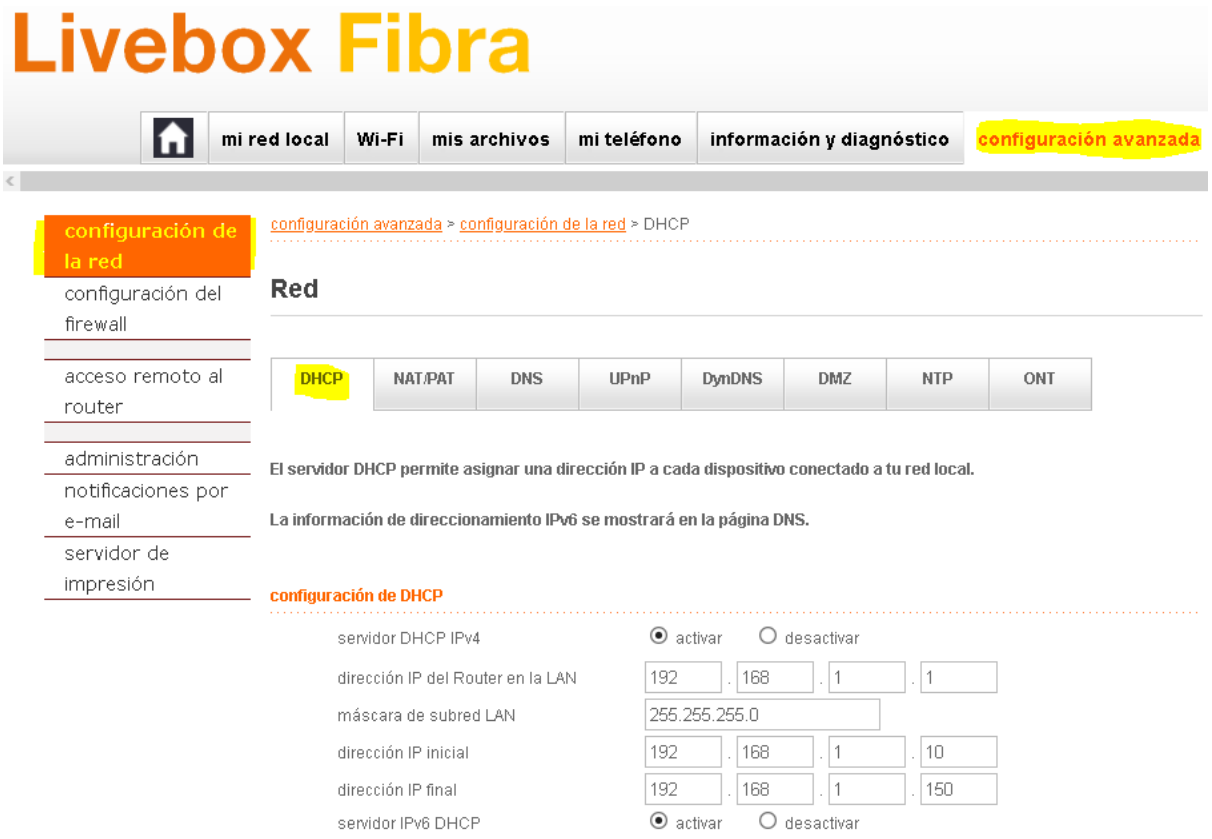


Figura 93: Configuración router doméstico DHCP

Figura 94: Configuración router doméstico DHCP asignar IP

- Después pulsamos en "*Configuración avanzada*", "*configuración de la red*", pestaña "*DMZ*" y añadir la IP reservada. Así, expondremos nuestro Honeypot en la DMZ.

Figura 95: Configuración router doméstico establecer DMZ

- El último paso es redireccionar los puertos que queramos a la ip interna. De esa manera todos los ataques recibidos los enviará al Honeypot.

Pulsamos "*configuración avanzada*", "*configuración de la red*", pestaña "*NAT/PAT*" y vamos añadiendo los puertos necesarios.

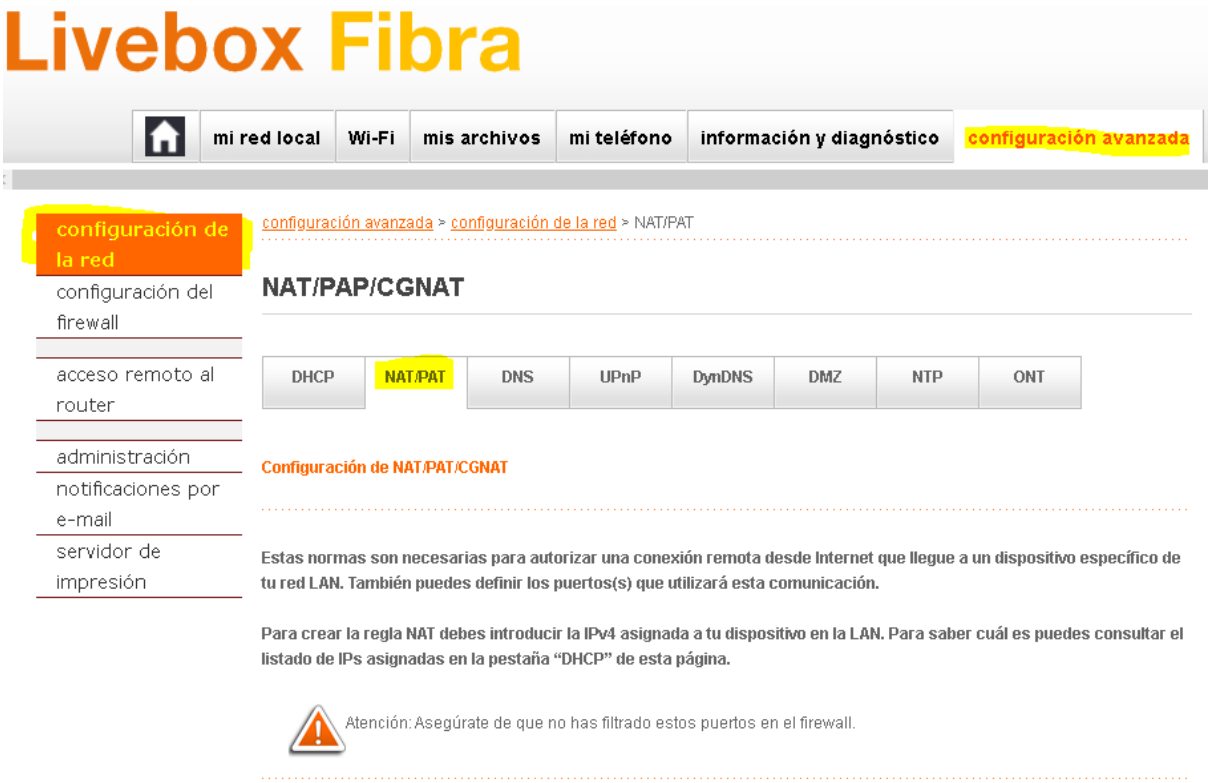


Figura 96: Configuración router doméstico NAT/PAT

Figura 97: Configuración router doméstico Port forwarding

	Secure Shell Server (SSH)	22	22	TCP	192.168.1.120	delete
	Telnet	23	23	TCP	192.168.1.120	delete
	FTP Server	21	21	TCP	192.168.1.120	delete
	6379	6379	6379	TCP	192.168.1.120	delete
	631	631	631	TCP	192.168.1.120	delete
	135	135	135	TCP	192.168.1.120	delete
	53	53	53	TCP	192.168.1.120	delete

Figura 98: Configuración router doméstico Port forwarding

Con estas operaciones, ya entraría todo el tráfico directamente a nuestro Honeypot.

Anexo C: Configuración pentester.

Una vez descargada la imagen ARM de Kali (Kali org) e instalada mediante Raspberry Imager, nuestro siguiente paso es entrar y actualizarla. Para logearnos por primera vez usamos el usuario kali y contraseña kali. Ejecutamos *"$sudo apt-get update && sudo apt-get upgrade"* para actualizarla y posteriormente "*$sudo apt-get dist-upgrade*" para actualizar los paquetes nuevos y descartar los antiguos.

Ahora nos queda realizar la batería de pruebas anotada en pasos anteriores:

Acceso anónimo.

Simplemente nos conectamos al punto wifi Raspator y ya estaremos anonimizados (Ver apartado navegación anónima).

Acceso a la red Tor.

Simplemente nos conectamos al punto wifi Raspator, abrir un buscador, encontrar una página índice de enlaces .onion y ya podremos navegar por la red Tor (Ver apartado navegación In-proxy).

Ataques a nuestro Honeypot.

Siguiendo las guías de test de penetración, podemos dividir las fases del test de penetración en las siguientes (Ver apartado test de penetración):

Fase 1 – Reconocimiento (Reconnaissance): …
Fase 2 – Escaneo (Scanning): …
Fase 3 – Obtener acceso (Gaining Access): …
Fase 4 – Mantener Acceso (Maintaining Access): …
Fase 5 – Limpiar Huellas (Clearing Tracks):

Para este trabajo nos centraremos en las que nos interesan para llegar a nuestros objetivos que son Fase 2 y 3 (escaneo y acceso). Las demás fases no son posibles porque precisamente es contra un Honeypot y está protegido y parcheado sin servicios vulnerables.

Para preparar estas fases, utilizaremos las herramientas Nmap, Rapidscan, Nikto, Dnsmap, Sqlmap, etc. La instalación y configuración vienen en detalle en el capítulo de Hacking ético: Test de penetración..

Anexo D: Configuración Honeypot.

En este anexo, mostraremos la instalación y configuración de las herramientas Dionaea y Ntopng para nuestro Honeypot.

Para instalar Dionaea debemos seguir los pasos indicados en el siguiente enlace (DinoTools).

- Accedemos al apartado instalación:

  https://dionaea.readthedocs.io/en/latest/installation.html

- Descargamos la herramienta con el siguiente comando y entramos en la carpeta:

  $git clone https://github.com/DinoTools/dionaea.git
  $cd dionaea

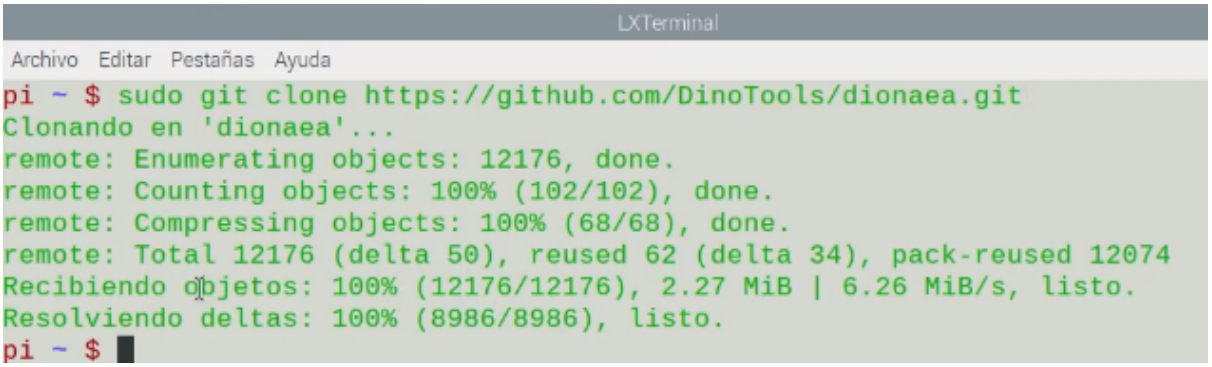


Figura 99: Clonado repositorio Dionaea

- Instalamos las dependencias necesarias para compilarlo:

$sudo apt-get install \
build-essential \
cmake \
check \
cython3 \
libcurl4-openssl-dev \
libemu-dev \
libev-dev \
libglib2.0-dev \
libloudmouth1-dev \
libnetfilter-queue-dev \
libnl-3-dev \
libpcap-dev \
libssl-dev \
libtool \
libudns-dev \
python3 \
python3-dev \
python3-bson \
python3-yaml \
python3-boto3 \
fonts-liberation

- Creamos el directorio y lo preparamos para su compilación:

  $sudo mkdir build

  $cd build

  $sudo cmake -DCMAKE_INSTALL_PREFIX:PATH=/opt/dionaea ..

- Por último, lo compilamos e instalamos:

  $sudo make

  $sudo make install

  Dentro de "*/opt/dionaea*" podremos encontrar los archivos necesarios para que funcione. En esta carpeta estará el binario "*bin/dionaea*" que podemos ejecutar para que ya empiece a comportarse como Honeypot. En el directorio "*etc/dionaea*" encontraremos el fichero de configuración "*dionaea.cfg*" donde podremos cambiar el comportamiento de los servicios emulados, cantidad de conexiones, interfaz elegido, etc. En el directorio "*var/lib/dionaea/*" encontraremos el registro de logs con los ataque recibidos, los binarios de los mismos así como los paquetes de red.

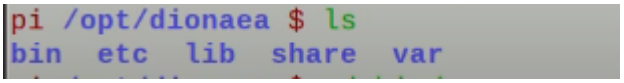


Figura 100: Estructura directorios Dionaea

NTopng:

La instalación de Ntopng se realiza usando el siguiente comando:

$sudo apt-get install ntopng

Para acceder al programa, utilizaremos el puerto 3000. Entramos con usuario admin contraseña admin abriendo un navegador a la dirección http://localhost:3000 y nos pedirá que establezcamos una nueva contraseña y ya podremos ver el tráfico de entrada y salida junto con los datos de puertos, IPs, etc.

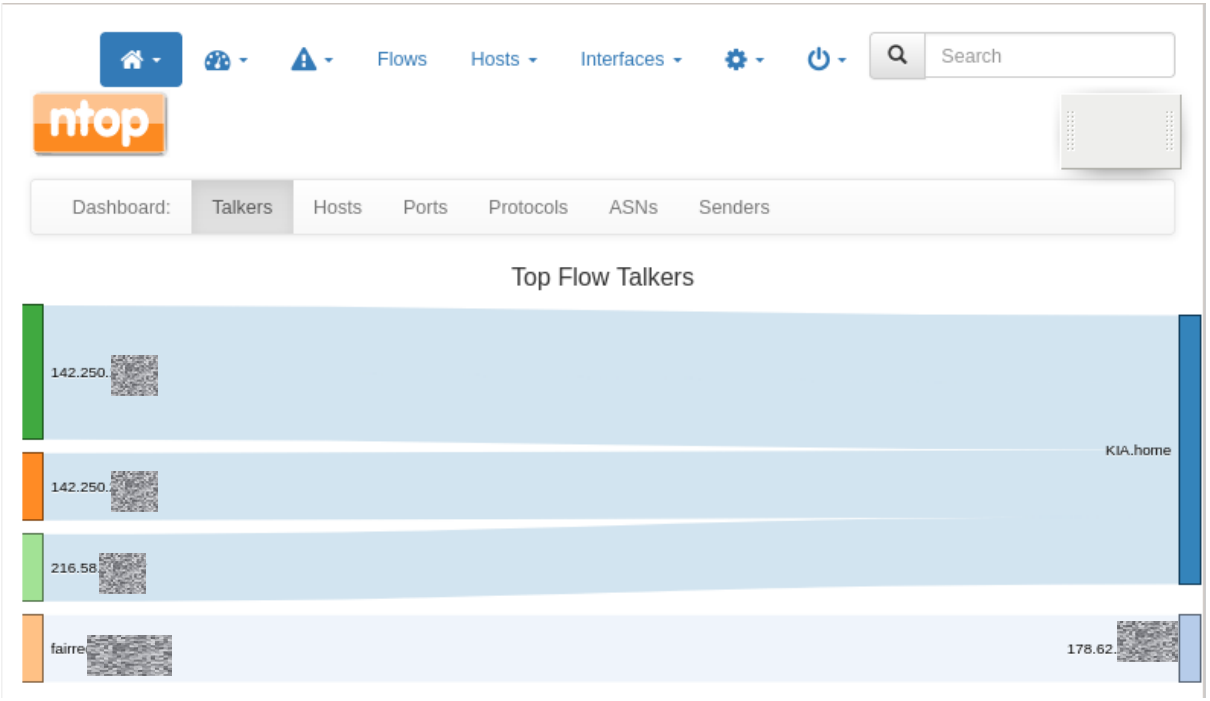
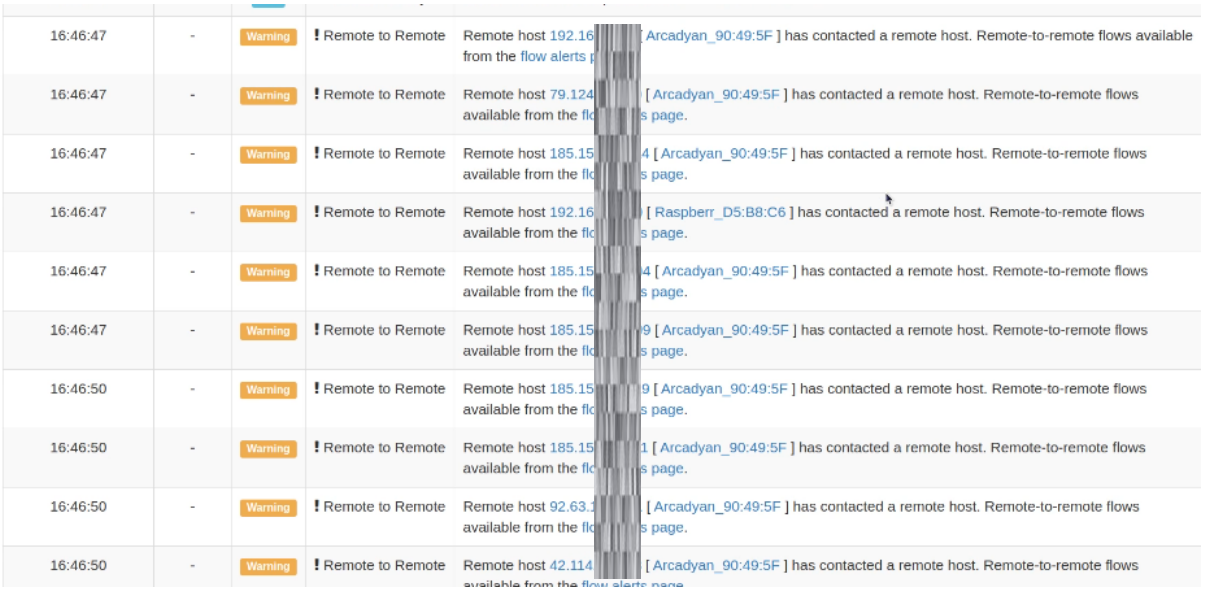


Figura 101: Muestra gráfica flujo de red NTopng

A parte de los datos recogidos, podemos ver otro tipo de información. Por ejemplo, en la siguiente figura podemos ver las alertas.

16/05/2021 11:21:23	-	Info	User Activity	User 'admin' logged in.
16/05/2021 11:35:00	02:00	Error	Flows Flood	Host raspberrypi.home is under flood attack (26 flows received in 00:03)
16/05/2021 11:35:01	01:00	Error	Flows Flood	Host 23.129 is a Flooder (26 flows sent in 00:03)
16/05/2021 11:36:01	01:00	Error	Flows Flood	Host 23.129 is a Flooder (26 flows sent in 00:03)
16/05/2021 12:50:47	-	Error	Process	Started after anomalous termination (bug report) ntopng v.3.8.190204 () [pid: 586][options: --daemon --interface 'eth0' --http-port '3000']

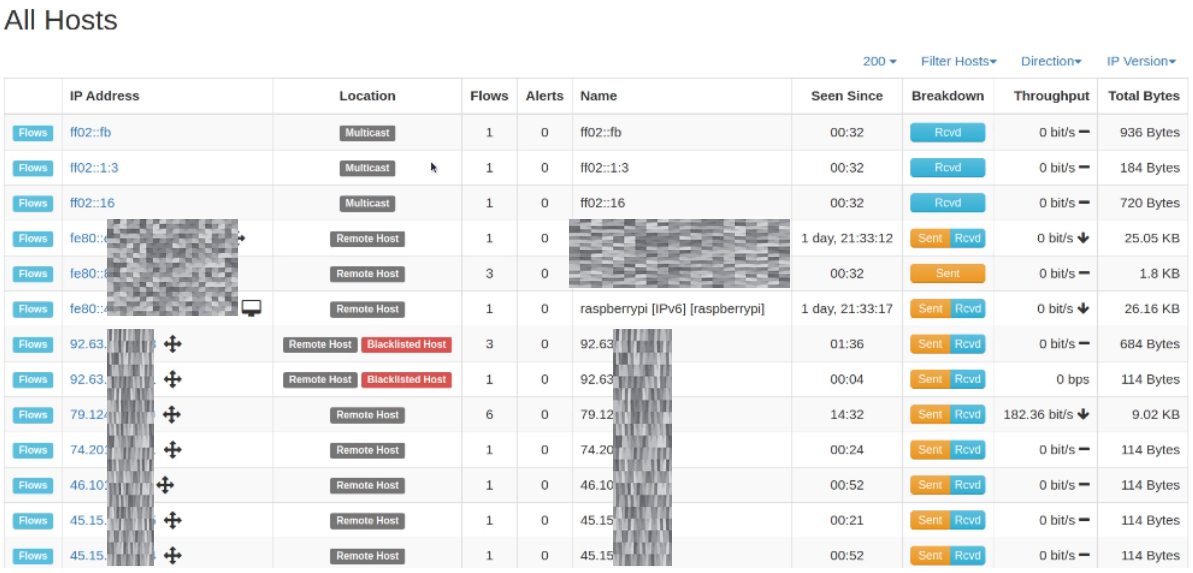
Figura 102: Alertas recogidas por Ntopng



16:46:47	-	Warning	! Remote to Remote	Remote host 192.16 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flow alerts p
16:46:47	-	Warning	! Remote to Remote	Remote host 79.124 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:47	-	Warning	! Remote to Remote	Remote host 185.15 4 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:47	-	Warning	! Remote to Remote	Remote host 192.16 [Raspberr_D5:B8:C6] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:47	-	Warning	! Remote to Remote	Remote host 185.15 4 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:47	-	Warning	! Remote to Remote	Remote host 185.15 9 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:50	-	Warning	! Remote to Remote	Remote host 185.15 9 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:50	-	Warning	! Remote to Remote	Remote host 185.15 1 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:50	-	Warning	! Remote to Remote	Remote host 92.63.1 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flo s page.
16:46:50	-	Warning	! Remote to Remote	Remote host 42.114 [Arcadyan_90:49:5F] has contacted a remote host. Remote-to-remote flows available from the flow alerts page.

Figura 103: Alertas recogidas por Ntopng

Si pulsamos en una IP o realizamos una búsqueda por host, nos aparecerán más información del mismo, incluso su tráfico de subida y bajada.



All Hosts

200 ▾ Filter Hosts▾ Direction▾ IP Version▾

	IP Address	Location	Flows	Alerts	Name	Seen Since	Breakdown	Throughput	Total Bytes
Flows	ff02::fb	Multicast	1	0	ff02::fb	00:32	Rcvd	0 bit/s	936 Bytes
Flows	ff02::1:3	Multicast	1	0	ff02::1:3	00:32	Rcvd	0 bit/s	184 Bytes
Flows	ff02::16	Multicast	1	0	ff02::16	00:32	Rcvd	0 bit/s	720 Bytes
Flows	fe80::[redacted]	Remote Host	1	0	[redacted]	1 day, 21:33:12	Sent Rcvd	0 bit/s	25.05 KB
Flows	fe80::[redacted]	Remote Host	3	0	[redacted]	00:32	Sent	0 bit/s	1.8 KB
Flows	fe80::[redacted]	Remote Host	1	0	raspberrypi [IPv6] [raspberrypi]	1 day, 21:33:17	Sent Rcvd	0 bit/s	26.16 KB
Flows	92.63.[redacted]	Remote Host Blacklisted Host	3	0	92.63[redacted]	01:36	Sent Rcvd	0 bit/s	684 Bytes
Flows	92.63.[redacted]	Remote Host Blacklisted Host	1	0	92.63[redacted]	00:04	Sent Rcvd	0 bps	114 Bytes
Flows	79.124.[redacted]	Remote Host	6	0	79.12[redacted]	14:32	Sent Rcvd	182.36 bit/s	9.02 KB
Flows	74.20[redacted]	Remote Host	1	0	74.20[redacted]	00:24	Sent Rcvd	0 bit/s	114 Bytes
Flows	46.10[redacted]	Remote Host	1	0	46.10[redacted]	00:52	Sent Rcvd	0 bit/s	114 Bytes
Flows	45.15.[redacted]	Remote Host	1	0	45.15[redacted]	00:21	Sent Rcvd	0 bit/s	114 Bytes
Flows	45.15.[redacted]	Remote Host	1	0	45.15[redacted]	00:52	Sent Rcvd	0 bit/s	114 Bytes

Figura 104: Host recogidos por Ntopng

Disponemos de un botón para descargarnos el archivo pcap si queremos analizarlo en profundidad.

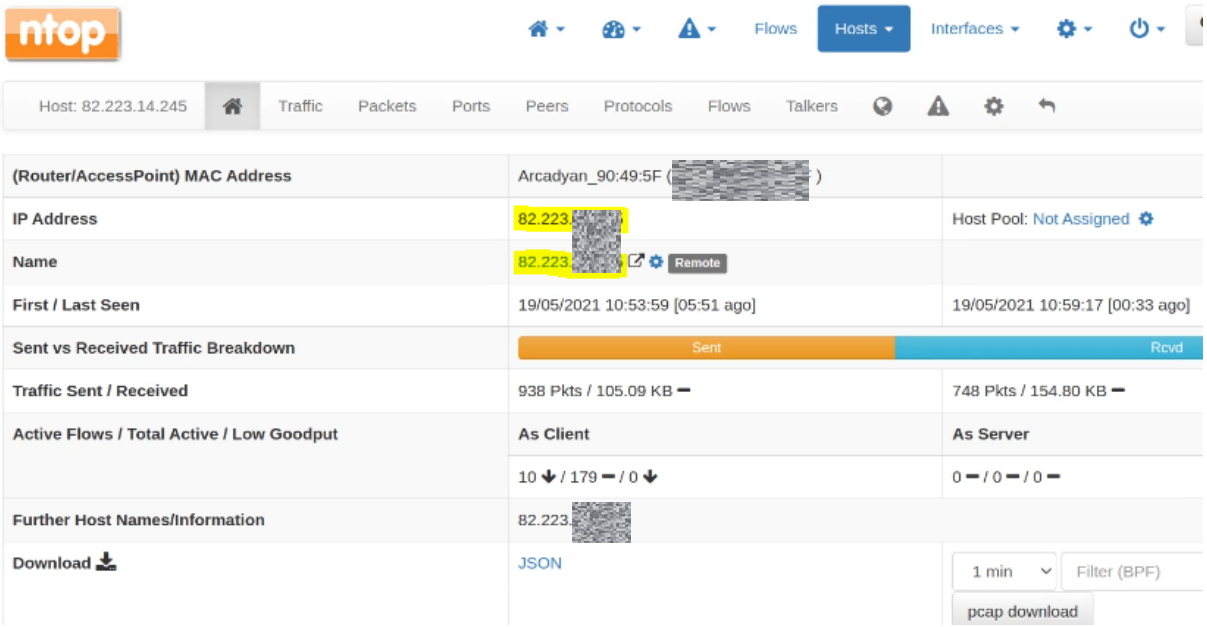


Figura 105: Vista detalle Host en NTopng

En la siguiente captura, podemos ver que los escaneos de puertos y lanzamiento de exploits no pasan desapercibidos para nuestro Honeypot. Se ven los puertos SMB, SSL, HTTP, etc.

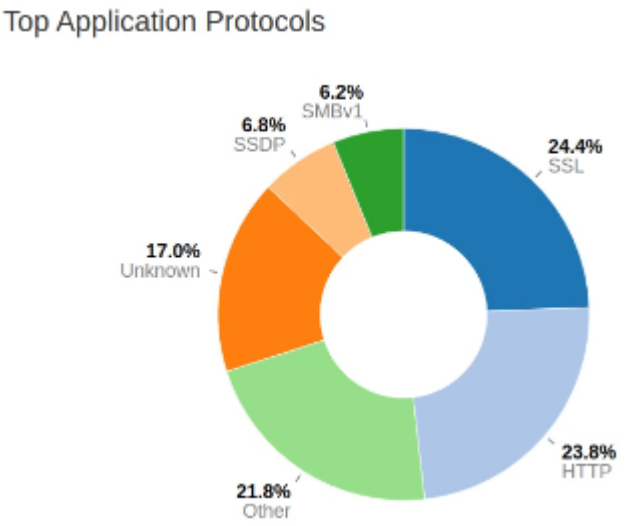


Figura 106: Tráfico de puertos recogidos por NTopng

Otra opción es el flujo de tráfico mostrado por el dashboard. En una sola pantalla podemos ver las IPs de las conexiones tanto entrantes como salientes, así como hacernos una idea de la cantidad de tráfico consumido.

Figura 107: Flujo de tráfico mostrado por el Dashboard en NTopng

La parte más importante de la herramienta utilizada en este trabajo es la del flujo de tráfico, ya que muestra las conexiones por cliente/servidor junto con sus respectivos puertos y protocolos.

Application	L4 Proto	Client	Server	Duration	Breakdown
SSDP	UDP	192.168.1.1:44012	239.255.255.250:1900	06:40:11	Client
SSL.GoogleServices	TCP	Honeypot:37792	content-autofill.googlea...:https	28:31	Client Server
IGMP	IGMP	192.168.1.1	224.0.0.1	06:40:03	Client
IGMP	IGMP	Honeypot	224.0.0.251	06:40:01	Client
NetBIOS	UDP	WORKGROUP:netbios-ns	192.168.1.255:netbios-ns	01:07	Client
SSDP	UDP	WORKGROUP:51605	239.255.255.250:1900	01:13	Client
MDNS	UDP	192.168.1.138:mdns	224.0.0.251:mdns	02:22	Client
SSDP	UDP	Honeypot:45105	239.255.255.250:1900	00:03	Client
Unknown	TCP	51.15 :40491	Honeypot:ms-sql-s	< 1 sec	Client Server

Figura 108: Escaneo MSSQL reflejado por NTopng

En el siguiente vídeo podemos ver la instalación completa de las dos herramientas:

https://www.youtube.com/watch?v=pkN_zWQFNGg

Wireshark:

Para la instalación de Wireshark debemos ejecutar el siguiente comando:

$sudo apt-get install wireshark

Anexo E: Cifrado conexión OP contra OR.

Comentamos el tipo de cifrado que usa el cliente OP (onion proxy) cuando se conecta con el nodo OR (onion router). Como mínimo, usará "*TLS/SSLv3"* para el enlace de autenticación y la encriptación. En cualquier caso, Tor Project recomienda usar como mínimo la suite de cifrado "*TLS_DHE_RSA_WITH_AES_128_CBC_SHA*". El iniciador de cifrado no puede incluir ninguna otra suite diferente a las tres siguientes:

"*TLS_DHE_RSA_WITH_AES_256_CBC_SHA*"
"*TLS_DHE_RSA_WITH_AES_128_CBC_SHA*"

"*SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA*".

(ver Tor-Spec (Dingledine et al.)).

Comentamos que significa cada campo.

SSL/TLS: Security socket layer o la evolución TLS Transport layer security. Se utiliza para proporcionar el canal seguro.

DHE: Ephemeral Diffie Hellman. Intercambio de claves, pero serán diferentes en cada negociación (CCN CERT).

RSA: Rivest, Shamir y Adleman. Sistema criptográfico de clave pública (cifrado asimétrico). Se usará para establecer las capas de los nodos entrada, puente y salida con las claves públicas de cada uno (CCN CERT).

AES: Advanced Encryption Standard. Se utiliza para cifrar datos o conexiones (cifrado simétrico) (CCN CERT).

CBC: Cipher block chaining. Es un tipo de cifrado de bloque en cadena que usa el bloque anterior para cifrar el siguiente (CCN CERT).

SHA: Secure hash algorithm. Sirve para generar una función hash, ya sea un resumen para integridad de datos, para autenticación, demostrar identidad en firmas, etc (CCN CERT).

El uso de todos estos estándares aseguran el anonimato en la red TOR.

Anexo F: Archivo de configuración torrc.

La herramienta Tor dispone de un archivo de configuración del que ya hemos hablado. Tiene suficientes opciones como para que hagamos un anexo sobre el mismo. Se encuentra en /etc/tor/torrc. Pasamos a comentar las opciones más interesantes:

- Las primeras opciones son las usadas en este trabajo:

  #Log notice file /var/log/tor/notices.log

- Almacenará los logs generados por Tor en el archivo que le digamos.

  #VirtualAddrNetwork 10.192.0.0/10

- El direccionamiento creado virtualmente para enrutar a Tor.

  #AutomapHostsSuffixes .onion,.exit

- Los sufijos que es capaz de interpretar.

  #AutomapHostsOnResolve 1

- Cuando accedemos a una dirección con sufijo nos asigna una ip virtual.

  #TransPort 192.168.220.1:9040

- La dirección IP y puerto de escucha de Tor.

  #DNSPort 192.168.220.1:53

- La dirección IP y puerto de escucha del servicio DNS.

  #NewCircuitPeriod NUM

- Número mínimo de tiempo de creación de otro circuito. 30 segundos por defecto.

  #MaxCircuitDirtiness

- Tiempo máximo para restablecer el circuito. Por defecto son 10 minutos, pero se puede modificar.

  #ExcludeNodes ABCD1234CDEF5678ABCD1234CDEF5678ABCD1234, {cc}, 255.254.0.0/8)

- Sirve para excluir nodos por si tenemos conocimiento de alguno que sea malicioso.

#ExcludeExitNodes node,node,…

- Excluir nodos de salida.

#ExitNodes node,node,…

- Establecer nodo o nodos de salida.

#MiddleNodes node,node,…

- Sirve para establecer nodos intermedios.

Hay otros nodos llamados bridge que no están publicados por Tor para evitar que se bloqueen en países con medidas de bloqueo.

Para encontrar los nodos puente podemos acceder a la siguiente web.

https://bridges.torproject.org/

#UseBridges 1
#Bridge obfs4 <ip:port><key>

- Usa este bridge para conectarse a Tor.

#EntryNodes node,node,…

- Establecer nodo o nodos de entrada.

#StrictNodes 0|1

- Si esta opción está en 1 obliga a utilizar la opción de los nodos que hayamos configurado.

#HiddenServicePort 80 127.0.0.1:2580

- Permite publicar puertos a la red Tor, por ejemplo servicios de un Honeypot (ver líneas futuras).
- Configuración de ExitNodes o EntryNodes:

#ExitNodes {ES}

- Le indicamos que salga por España pero si queremos Francia podemos poner ,{FR}, etc

#ExcludeExitNodes {ES}

- Le indicaremos que **no** salga por España

#EntryNode {ES}

- Le podemos decir nodo del país de entrada. En los dos casos se puede pasar el "*fingerprint*" para seleccionar un nodo en concreto. Nota: El "*fingerprint*" lo podemos obtener con la herramienta que comentamos a continuación (https://metrics.torproject.org/rs.html).

Una vez revisadas las opciones del archivo de configuración torrc, pasamos a comentar algunos comandos útiles:

- Comandos **útiles con Tor:**

$tor-resolve dominio.es

Sirve para resolver dominios usando tor. Como vemos, no obtenemos la IP destino de ese dominio debido a que es un hidden services, ni el cliente sabe la ip del servidor destino ni el servidor destino conoce al cliente. Lo que nos muestra es la dirección IPv6 del punto de encuentro. Si reiniciamos el servicio de Tor vemos que cada vez nos da una dirección diferente.

Figura 109: Tor resolver

$sudo service tor star

- Levanta el servicio tor.

$sudo torStatus

- Muestra las direcciones IP de los nodos y sus nombres.

$sudo proxychains curl ipinfo.io

- Envía la petición de la página ipinfo.io a través de la red tor y esta web nos dice la ip de tor.

$sudo torsocks o tsocks

- Viene incluido con Tor y se encarga de torificar una consola. Abre una consola y todo lo que se ejecuta en ella va por la red Tor. “. *torsocks on*”.

Otras características interesantes de Tor:

Encontrar bridges de Tor para usar en el archivo torrc.

https://bridges.torproject.org/.

Figura 110: Encontrar bridges para usar red Tor. (Tor Project Inc.)

- Encontrar el estado de los relés y bridges.

En la siguiente web https://metrics.torproject.org/rs.html podemos encontrar el estado de los nodos, su IP, velocidad y los servicios que es capaz de proporcionar.

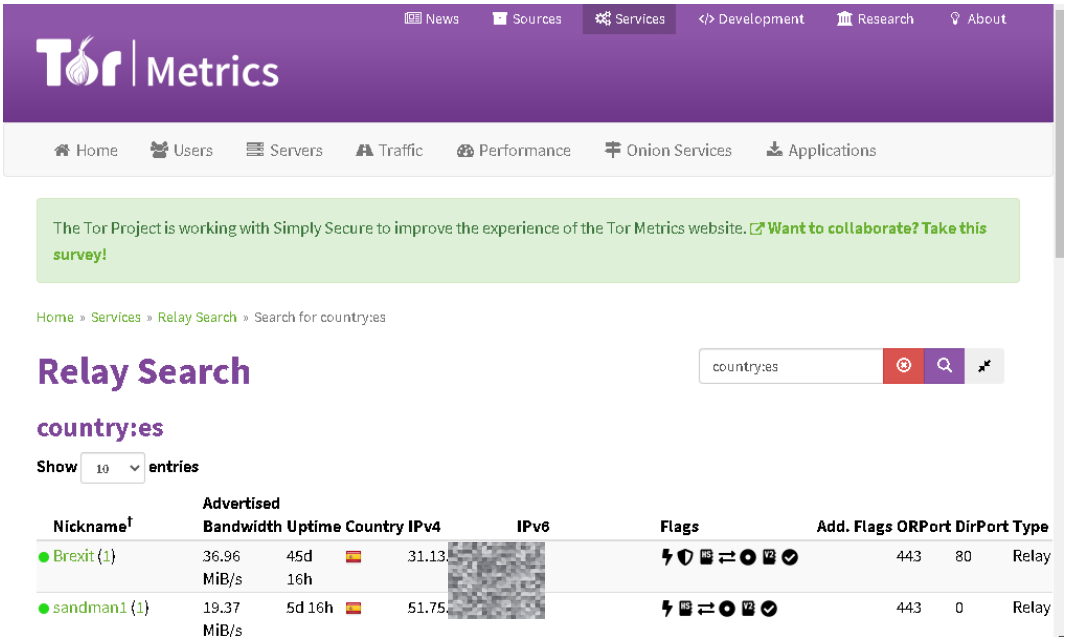


Figura 111: Ver estado de los nodos red Tor. (Tor Project Inc.)

- **ooni explorer**

Web que indica países que tienen impuestas medidas de censura (Open Observatory of Network Interference (OONI)).

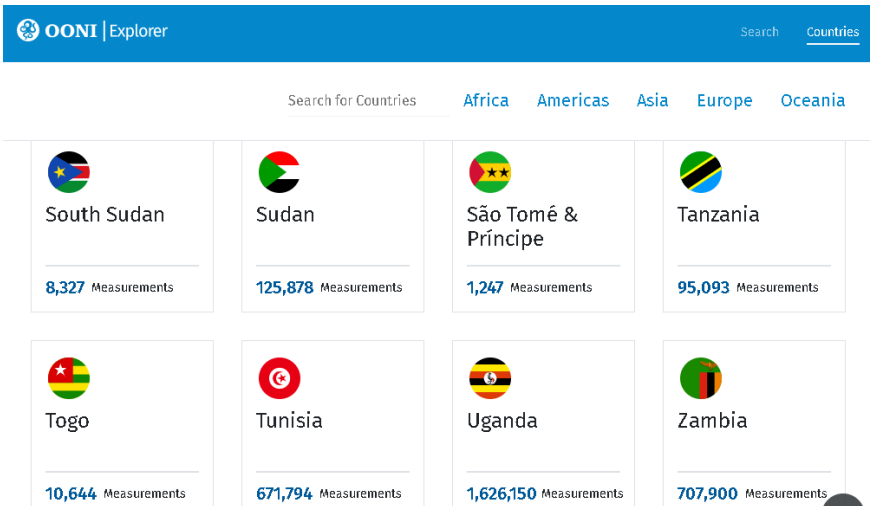


Figura 112: Ver censura por países. (Open Observatory of Network Interference (OONI))

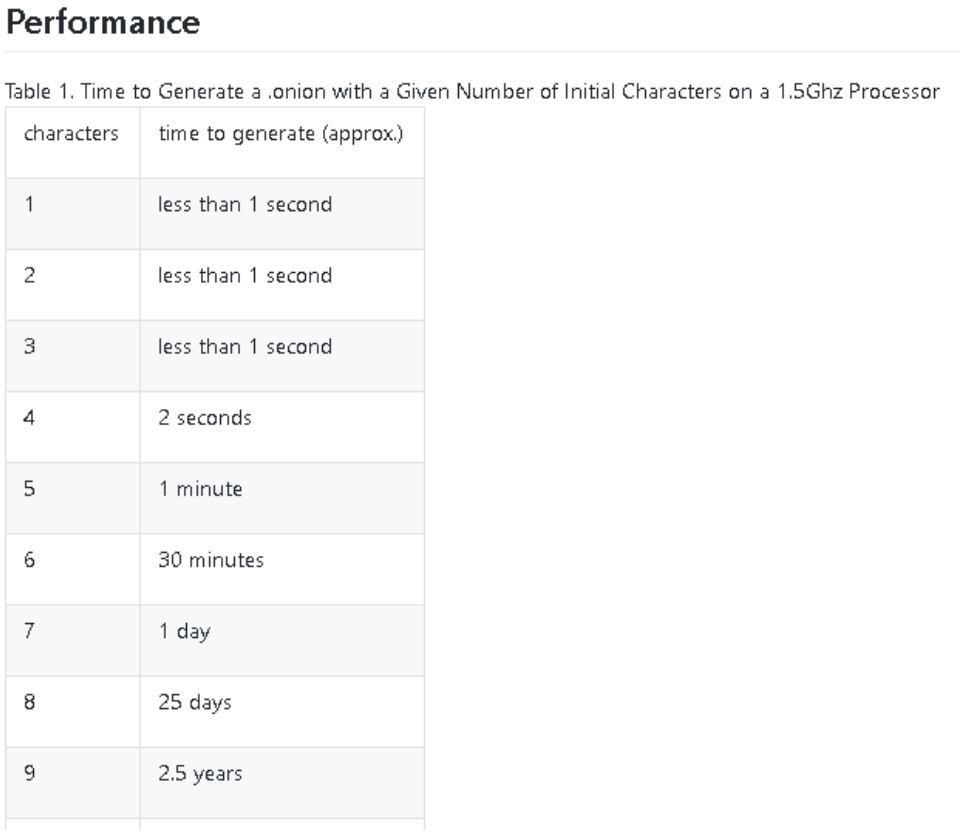
- **Onionscan**

Escaneo de la red Tor y sus dominios, es capaz de escanear servicios usando proxychains. (Lewis y MIT)

- **Shallot**

  Fuerza la creación de dominios .onion personalizados (Katmagic). Genera una clave privada para meter en el archivo torrc que será su identidad Tor. Genera nombres similares, ya que es difícil generarlos completamente iguales. No hubo éxito en la instalación debido a que había errores al compilarlo, en su lugar se utilizó "*eschalot*". El proceso es el mismo, se descarga de GitHub, se compila y se ejecuta proporcionando la clave del dominio incluyendo la palabra que queramos.

  En la siguiente captura podemos ver la relación de tiempo con respecto del tamaño buscado de palabra.



Performance

Table 1. Time to Generate a .onion with a Given Number of Initial Characters on a 1.5Ghz Processor

characters	time to generate (approx.)
1	less than 1 second
2	less than 1 second
3	less than 1 second
4	2 seconds
5	1 minute
6	30 minutes
7	1 day
8	25 days
9	2.5 years

Figura 113: Performance de tiempos aplicación Shallot. (Katmagic)

- Mediante **eschalot** (Reclaim Your Privacy (Cloak)) generamos un nombre de dominio .onion que empiece por unedmuc (Uned Máster Universitario en Ciberseguridad).

```
root@debian:/home/deb/Descargas/eschalot-master# ./eschalot -vt4 -r "^unedmuc"
Verbose, single result, no digits, 4 threads, prefixes 1-16 characters long.
Thread #1 started.
Thread #2 started.
Thread #3 started.
Thread #4 started.
Running, collecting performance data...
Total hashes: 32430070, running time: 10 seconds, hashes per second: 3243007
Total hashes: 98331765, running time: 30 seconds, hashes per second: 3277725
Total hashes: 226011159, running time: 70 seconds, hashes per second: 3228730
Total hashes: 482391802, running time: 150 seconds, hashes per second: 3215945
Total hashes: 996927573, running time: 310 seconds, hashes per second: 3215895
Total hashes: 2062668603, running time: 630 seconds, hashes per second: 3274077
Total hashes: 4238876785, running time: 1270 seconds, hashes per second: 3337698
Total hashes: 8432123818, running time: 2550 seconds, hashes per second: 3306715
Found a key for unedmucoyjez46bm (16) - unedmucoyjez46bm.onion
----------------------------------------------------------------
unedmucoyjez46bm.onion
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDyDyBUm8emUf30n5z21GrTekudXn95tP551ym1Au8Xpj8LsO5u
YY++GQohGNCghB9hbgpQ36IJjL+0Wwr+2KRQa0dFrZQiD+fwJ1kDdMlxZxjiJlK/
s2q0vrGTdU3bmMTzEFS3e7E1dmeQR8CsMSdttpgkpa08hhrMNbM3N6aWMwIEEn5+
hwKBgD+Ncj2/HD2aDidIKRwJH7l3MkIfPpn3pDgZAswIfYopPIK6ZzNptbfO/Iwp
agF5AMhUmV7qKV2ZaYH9PscaZV7f9VicED3WSX3fvnca/ovIyF+LyJXSH2/YTIMj
+XA/mUEiM7OadNC0ZliQVH+M48v+ngJA1p7Tca5QiaqmXc2bAkEA/7hvwBlLRrZz
8OYj3PZYw26xfK6PPs2X6lj72FMnV5PRyLyQr/YcKYdM14arYn4QgWXc1eCgGByk
QuZX+CMrfQJBAPJS3ds9Jwx6706bUWULNKYfBVe46VS2qjFed58qPk63NQ7f8kvt
WnbPIts/rb9wCOQrUB5AYcqCB/d9xZTAl28CQHd3UhHWAOvOo0Z3dTTqYMUuPkeV
cAIuD1hzFtfp9A8ApYoTQECd/HxGIBjrUykX032DMy6vUiHYIlQyXiiWJN8CQBh0
K/OWu3peObPQI41JxawW+CRSVvYELbDiNb9wNpcc+q2Aejo+eEWV4pXFySxchdoy
TOt1Ye5qZh72fT5UY7UCQFctZrZN7g/JIqakOkzDBJZxdULJxJh+RYlAhhhVm0X8
ZA5DXaz3VyettgyLPt4gy+60IehNwlmdk+A5I1EhAPc=
-----END RSA PRIVATE KEY-----
```

Figura 114: Generar URL .onion con eschalot.

Mediante esta clave privada, podemos disponer de un dominio en la red Tor.

El dominio sería "*unedmucoyjez46bm.onion*". Podemos probar a ponerle en nuestro archivo de configuración las líneas abajo nombradas y en la carpeta tor_service los ficheros "*hostname*" y "*private_key*", en "*hostname*" pondremos el "*unedmucoyjez46bm.onion*" y en "*private_key*" la clave generada con "*eschalot*".

En el archivo torrc, añadir estas líneas, la primera para decir dónde está el servicio que aloja hostname y private_key y la segunda la IP local de la máquina y el puerto compartido.

 # Hidden Service
HiddenServiceDir C:\Users\Name\tor_service

HiddenServicePort 80 127.0.0.1:80

Nota: Esto se utilizaría para un proyecto futuro, el de Honeypot y escaneo de vulnerabilidades dentro de la red Tor (Ver líneas futuras).

- **Compartir puertos**

Para compartir puertos, hay que editar el fichero torrc y HiddenServicePort decirle lo que se quiere publicar, no tiene porque ser servicio Web, puede ser cualquier otro servicio que no use UDP.

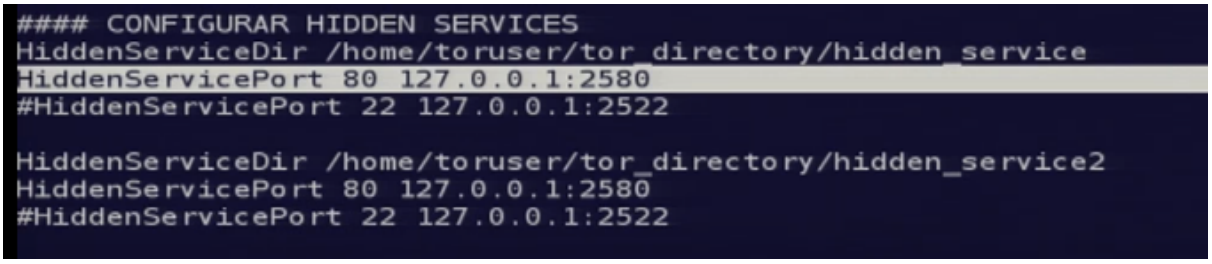


Figura 115: Compartir puertos en la red Tor

Nota: Este proceso es antiguo y es posible que haya cambiado. En el siguiente enlace, afirman que el proceso ha cambiado o que está en vías de cambiarlo y es del año 2018 (Junquera Sánchez y XII Jornadas STIC CCN-CERT.).

Nota: Esto se utilizaría para un proyecto futuro, el de Honeypot y escaneo de vulnerabilidades dentro de la red Tor (Ver líneas futuras).

- **Torspec**:

En Torspec tenemos todas las especificaciones de TOR. (Dingledine et al.)

Anexo G: Problemas encontrados.

En este anexo hemos recopilado los problemas encontrados durante el trabajo realizado así como las soluciones.

- Fallo al arrancar TTopng después de instalarlo y apagar la Raspberry.

“Failed to star ntopng.service: Unir ntop.service os masked”

Solución: Abrir el terminal y ejecutar.

$sudo apt-get update

$sudo apt-get install ntopng

- Si se cambia de interfaz de salida a Internet eth0 a ppp0 o viceversa no funciona Tor.

Solución: Hay que reiniciar el servicio. Se coloca una instrucción en el archivo de arranque de usuario como se hizo con "*wvdial*".

sudo service tor restart

- IPv6 dió problemas en alguna de las pruebas, para evitar futuros errores, se deshabilita.

  Solución:

  Editar fichero "*/etc/sysctl.conf*" y añadir la línea *"net.ipv6.conf.all.disable_ipv6 = 1"*
  Ejecutar en terminal: *$sudo sysctl -p*
  Ejecutar en terminal: ifconfig para confirmar que no aparecen direcciones IPv6.

- Poner tarjeta eth0 del Honeypot en modo promiscuo.

  Solución: Ejecutar en terminal:

  $sudo ip link set eth0 promisc on

- Como el problema 2, para que no se solapen "*wvdial*" con el inicio de Tor o el arranque del interfaz eth0, poner una espera de 30 segundos antes de reiniciar el servicio tor.

  Solución: Editar .bashrc y meter entre "*sudo wvdial*" y "*sudo service tor restart*" la línea "*sleep 30*"

- Pérdida de conexión y anonimato. Al realizar escaneos automatizados con las herramientas que tardan mucho tiempo (Rapidscan), como el circuito de Tor por defecto se reinicia cada 10 minutos, es conveniente aumentar ese tiempo, ya que hemos comprobado que pierde el anonimato durante un rato y envía el escaneo con la IP legítima.

  Solución: Para solucionarlo, en el archivo de configuración /etc/tor/torrc, hay que indicarle que el tiempo de renovación sea más amplio.

  MaxCircuitDirtiness 3600

- Lentitud. Al conectarnos a la red Tor, se nota mucho la latencia del paso a través de tres nodos Tor. Para solucionarlo en parte, elegimos un nodo más cercano físicamente de salida. Aunque si uno de los nodos de entrada y

"*middle*" son de otro país más alejado o con conexión más lenta también influirá, pero algo de velocidad ganamos al evitar más latencia.

Solución: añadir la siguiente entrada en el archivo /etc/tor/torrc para que los nodos de salida sean de España, Francia o Portugal.

ExitNodes {ES}{FR}{PO}

- Suspensión del sistema Kali irrecuperable. Al pasar varios minutos de inactividad en la Raspberry con Kali instalado, light-locker bloquea la sesión poniendo la pantalla en negro y no se es capaz de recuperar la sesión.

  Solución: No se pudo recuperar la sesión, pero para que no vuelva a suceder, desinstalamos light-locker e instalamos xscreensaver.

  $sudo apt-get install xscreensaver

- Openvas o gvm. La Raspberry Pi 3B no es capaz de ejecutar este programa. Si en Kali se instala mediante apt-get no aparece la opción openvas-setup imprescindible para su configuración. Se intentó instalar de varias maneras y en el resto de Raspberry sin éxito.

- Temperatura Raspberry Pi 3B sin ventilador. La caja de la Raspberry Pi 3B no dispone de ventilador y las necesidades de Kali sobre este dispositivo son tan elevadas al realizar escaneo de vulnerabilidades junto con el software de grabación que salta el sensor de temperatura por encima de los 80º.

  Solución: Se le retira la tapa y no se abusa del tiempo de escaneo unido al tiempo de grabación, sino que se graba en momentos puntuales y necesarios.

- Como el router nos interesa que arranque solo, pero no está instalado interfaz gráfico, no se produce el autoarranque y no es capaz de ejecutar lo que se encuentra en el archivo "*.bashrc*" del usuario Pi.

  Solución: Hay que realizar el proceso del anexo del Router donde configuramos la parte del **autologin**.

- Kali consume demasiados recursos al lanzar ciertas herramientas de pentesting como Armitage, Scripts de NMAP, etc.

  Solución: Pasar ese rol a la Raspberry Pi 4 de 2Gb y realizar el enrutamiento con la RPI 3.

- En los escaneos de vulnerabilidades, hay que saltarse los que usan UDP, esto se debe a que no es posible realizarlos en Tor.
- Curl refresca la página web cacheada de ipinfo.io con lo que al realizar escaneos no veíamos nuestra nueva IP proporcionada por Tor, sino la anterior.

  Solución: Ejecutar curl con el siguiente comando para que no use la caché:

  $curl -H 'CacheControl: no-cache' ipinfo.io

  o

  $curl -s --max-time 60 https://api4.ipify.org

Referencias y bibliografía

Alcocer, Manuel. “Crear punto de acceso con 'hostapd.'” Hostapd, 2016, https://github.com/manuelalcocer/hostapd. Fecha de acceso marzo 2021.

Aparicio, Pablo. “Simple Screen Recorder, una nueva opción para grabar la pantalla de tu PC.” 2018, https://ubunlog.com/simple-screen-recorder-grabar-pantalla/. Fecha de acceso marzo 2021.

Aplura. “Tango honeypot intelligence with Splunk.” 2018, https://github.com/aplura/Tango. Fecha de acceso mayo 2021.

ArchLinux. “Tor. Iptables.” 2021, https://wiki.archlinux.org/title/tor#iptables. Fecha de acceso abril 2021.

“Arrestado el "hacker" sueco que reveló contraseñas de embajadas.” 16 noviembre 2007, https://www.ccn-cert.cni.es/gl/gestion-de-incidentes/lucia/23-noticias/667-arrestado-el-qhackerq-sueco-que-revelo-contrasenas-de-embajadas.html. Fecha de acceso mayo 2021.

Astudillo B, Karina. “Instalando Golismero en Kali 2020.” 22 julio 2020, https://www.youtube.com/watch?v=Mit5gU9UomU. Fecha de acceso abril 2021.

Baert, Maarten. “SSR SimpleScreenRecorder.” 2021, https://github.com/MaartenBaert/ssr. Fecha de acceso marzo 2021.

Bakker, Paul. “Why use Ephemeral Diffie-Hellman.” *ARM MBED*, 10 diciembre 2013, https://tls.mbed.org/kb/cryptography/ephemeral-diffie-hellman. Fecha de acceso marzo 2021.

Bhat, Umesh .A. “How does Tor actually work?” 2019, https://hackernoon.com/how-does-tor-really-work-5909b9bd232c. Fecha de acceso abril.

Byte Mind. “Escaneando la red con Nmap en Kali Linux.” 30 septiembre 2017, https://byte-mind.net/escaneando-la-red-con-nmap/. Fecha de acceso abril 2021.

Castillo, Pedro. “Servicios ocultos en Tor, ¿cómo consiguen pasar desapercibidos?” 29 junio 2015, https://securityinside.info/servicios-ocultos-en-tor-como-pasan-desapercibidos/. Fecha de acceso marzo 2021.

Castro Astroz, Dinael Antonio. “Honeypot de dispositivos IoT usando una Raspberry Pi 3.” 31 diciembre 2018, http://openaccess.uoc.edu/webapps/o2/bitstream/10609/89885/3/dcastroaTFM1218memoria.pdf. Fecha de acceso marzo 2021.

CCN-CERT. “Arrestado el "hacker" sueco que reveló contraseñas de embajadas.” 16 noviembre 2007, https://www.ccn-cert.cni.es/gl/gestion-de-incidentes/lucia/23-noticias/667-arrestado-el-qhackerq-sueco-que-revelo-contrasenas-de-embajadas.html. Fecha de acceso junio 2021.

CCN CERT. “Glosario términos. AES, CBC, SHA, Algoritmo Diffie Hellman.” https://www.ccn-cert.cni.es/publico/seriesCCN-STIC/series/400-Guias_Generales/401-glosario_abreviaturas/index.html?n=21.html. Fecha de acceso marzo 2021.

CJ Click Jurídico. “¿ES LEGAL EN ESPAÑA NAVEGAR POR LA DEEP WEB?” 30 agosto 2017, https://clickjuridico.es/es-legal-navegar-por-la-deep-web/. Fecha de acceso marzo 2021.

Comunidad Element14. “Figura Raspberry Pi 3 Modelo B.” 2016, https://www.element14.com/community/docs/DOC-81294/l/raspberry-pi-3-model-b-with-1gb-of-ram-with-wifi-and-bluetooth-low-energy. Fecha de acceso marzo 2021.

Cots Sanfeliu, Jordi. “Un paseo por la Deep Web.” 31 diciembre 2018, http://openaccess.uoc.edu/webapps/o2/bitstream/10609/88545/7/jcotssTFM0119memoria.pdf. Fecha de acceso marzo 2021.

Crespo, Adrian. “Loopix, una nueva red anónima que quiere robar la hegemonía a Tor.” 18 septiembre 2017, https://www.redeszone.net/2017/09/18/loopix-una-nueva-red-anonima-quiere-robar-la-hegemonia-tor/. Fecha de acceso junio 2021.

CristianRus4 y Xataca. “Alguien se está haciendo con Tor: ya controla más del 10% de los nodos de salida que permiten interceptar el tráfico de la red.” 12 agosto 2020, https://www.xataka.com/seguridad/alguien-se-esta-haciendo-tor-controla-10-nodos-salida-que-permiten-interceptar-trafico-red. Fecha de acceso mayo 2021.

Daladim. “Umtskeeper.” 2018, https://github.com/daladim/umtskeeper. Fecha de acceso marzo 2021.

Datasoft. “Honeyd.” 2013, https://github.com/DataSoft/Honeyd. Fecha de acceso marzo 2021.

Dest-unreach.org. “Socats.” 2021. Fecha de acceso abril 2021.

Díaz, Jesús e Incibe. “Redes anónimas: más allá de Tor.” 14 abril 2015, https://www.incibe-cert.es/blog/redes-anonimas-mas-alla-de-tor. Fecha de acceso junio 2021.

Dietze, Josua y Draisberghof. “Usb-Modeswitch.” 2019, https://www.draisberghof.de/usb_modeswitch/#download. Fecha de acceso marzo 2021.

Dingledine, Roger, et al. “Tor protocol Specification.” 2009, https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt. Fecha de acceso abril 2021.

DinoTools. “Dionaea.” 2021, https://github.com/DinoTools/dionaea. Fecha de acceso abril 2021.

Dragonjar. “Manual de Armitage en Español.” 2014, https://www.dragonjar.org/manual-de-armitage-en-espanol.xhtml. Fecha de acceso mayo 2021.

EC-Council. “CEH.” 2021, https://www.eccouncil.org/programs/certified-ethical-hacker-ceh-es/. Fecha de acceso abril 2021.

Echeverri, Daniel, e Incibe. “Hackeando TOR y Freenet.” 12 marzo 2018, https://www.youtube.com/watch?v=V1msORieS4I. Fecha de acceso marzo 2021.

Emmet y Pimylife. “Installing Nagios on the Raspberry Pi.” 3 junio 2020, https://pimylifeup.com/raspberry-pi-nagios/. Fecha de acceso mayo 2021.

Estremadoyro, Pablo y Altronics. “Tutorial PPP.” 2019, https://altronics.cl/uploads/TutorialPPP_Rev1.pdf. Fecha de acceso marzo 2021.

Evdokimov, Leonid. “RedSocks.” 2018, https://github.com/darkk/redsocks. Fecha de acceso abril 2021.

Færøy, Alexander. “TransparentProxy.” 2014, https://gitlab.torproject.org/legacy/trac/-/wikis/doc/TransparentProxy. Fecha de acceso abril 2021.

Fajardo, Jay y Jayfajardo. “RaspTor implements a Wi-Fi router and a TOR Proxy on a Raspberry Pi.” 2016, https://github.com/jayfajardo/rasptor. Fecha de acceso abril 2021.

Fire1ce. “Raspberry Pi 3 TOR Access Point Router.md.” 2018, https://gist.github.com/fire1ce/bc8d0ab9e1aeb4c83b46a22df7846625. Fecha de acceso abril 2021.

Fuentes Iglesias, https://gist.github.com/fire1ce/bc8d0ab9e1aeb4c83b46a22df7846625. “TOR Hidden Service Discovery.” enero 2018, http://openaccess.uoc.edu/webapps/o2/handle/10609/74425. Fecha de acceso mayo 2021.

Fundación Raspberry PI. “GPIO.” 2014, https://www.raspberrypi.org/documentation/usage/gpio/. Fecha de acceso marzo 2021.

Fundación Raspberry PI. “Raspberry Pi 3 Model B.” https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. Fecha de acceso marzo 2021.

Fundación Raspberry PI. “Raspberry Pi 4 Tech Specs.” 2021, https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/. Fecha de acceso marzo 2021.

Gago Padreny, Ignacio. “Sistema de detección de ataques DDoS en Tor.” junio 2015, https://eprints.ucm.es/id/eprint/33415/1/memoria_tfg.pdf. Fecha de acceso marzo 2021.

Gaitatzis, Tony. “Browse Anonymously with a DIY Raspberry Pi VPN/TOR Router.” 24 abril 2015, https://makezine.com/projects/browse-anonymously-with-a-diy-raspberry-pi-vpntor-router/. Fecha de acceso abril 2021.

Garcia, Ruben y Alexander Færøy. “Torsocks.” 2020. Fecha de acceso abril 2021.

García Lopez, Roberto. “Red de anonimización Tor y cibermercados negros.” diciembre 2017, http://openaccess.uoc.edu/webapps/o2/bitstream/10609/72568/6/rgarcialopez012TFM0118memoria.pdf. Fecha de acceso marzo 2021.

Gil Gil, Alicia y Roberto Hernández Berlinches. *Cibercriminalidad*. 1 ed., Editorial Dykinson, 2019, https://www.librosuned.com/LU26674/Cibercriminalidad.aspx. Fecha de acceso mayo 2021.

Golismero. “Golismero.” 2020, https://github.com/golismero/golismero. Fecha de acceso mayo 2021.

GoVanguard. “Legion.” 2021, https://govanguard.com/legion/. Fecha de acceso junio 2021.

Gus y PiMyLifeUp. “How to setup a Raspberry Pi TOR Access Point.” 2019, https://pimylifeup.com/raspberry-pi-tor-access-point/. Fecha de acceso marzo 2021.

Gus y PiMyLifeUp. “Raspberry Pi Wireless Access Point.” 2019, https://pimylifeup.com/raspberry-pi-wireless-access-point/. Fecha de acceso marzo 2019.

Guy Bruneau y SANS ISC. “RaspberryPiInstall. Installing the DShield Honeypot on a Raspberry Pi.” 11 noviembre 2019, https://isc.sans.edu/honeypot.html. Fecha de acceso abril 2021.

Guy Fawkes y Vpn Mentor. “VPNs anónimas.” 17 mayo 2021, https://es.vpnmentor.com/blog/las-mejores-vpn-que-no-guardan-registros-de-tu-actividad/. Fecha de acceso marzo 2021.

Hamsik, Adam. “Proxychains.” 2021, https://github.com/haad/proxychains. Fecha de acceso marzo 2021.

Hiroshiman. “Anonym8.” 2016, https://github.com/HiroshiManRise/anonym8. Fecha de acceso marzo 2021.

Honeytrap. “Honeytrap.” 2021, https://github.com/honeytrap/honeytrap. Fecha de acceso marzo 2021.

Insecure.Com LLC. “NMap Security.” 2021, https://nmap.org/man/es/. Fecha de acceso mayo 2021.

Isaac. “Linus Adictos.” *Alternativas a red Tor*, https://www.linuxadictos.com/i2p-y-freenet-alternativas-a-la-red-tor.html. Fecha de acceso mayo 2021.

J. Menéndez, Julian y Una al Día. Hispasec. “Más del 25% de los nodos de salida Tor espiaron a sus usuarios.” 11 mayo 2021, https://unaaldia.hispasec.com/2021/05/mas-del-25-de-los-nodos-de-salida-tor-espiaron-a-sus-usuarios.html. Fecha de acceso mayo 2021.

Julian, Guillermo. “Si Tor es seguro ¿cómo se llevó a cabo la Operación Onymous?” 18 noviembre 2012, https://www.genbeta.com/seguridad/si-tor-es-seguro-como-se-llevo-a-cabo-la-operacion-onymous. Fecha de acceso mayo 2021.

Junquera Sánchez, Javier y XII Jornadas STIC CCN-CERT. “Ciberseguridad. Técnicas avanzadas de descubrimiento y análisis de la Dark Net.” 2018. Fecha de acceso mayo 2021.

Kali org. “Kali.” 2021, https://www.kali.org/. Fecha de acceso abril 2021.

Katmagic. “Shallot.” 2012, https://github.com/katmagic/Shallot. Fecha de acceso mayo 2021.

Krishnan, Susmith. “TorGhost.” 2020, https://github.com/SusmithKrishnan/torghost/blob/master/torghost.py. Fecha de acceso marzo 2021.

La Vanguardia. “Nuevos detalles sobre el espionaje de EE.UU. a Merkel comprometen a Dinamarca.” 31 mayo 2021, https://www.lavanguardia.com/internacional/20210531/7493157/nuevos-detalles-espionaje-eeuu-merkel-dinamarca.html. Fecha de acceso junio 2021.

Lachance, William y Wlach. “Wvdial.” 2009, https://github.com/wlach/wvdial. Fecha de acceso marzo 2021.

Lady Ada y Adafruit. “Onion Pi. Make a Raspberry Pi into an Anonymizing Tor Proxy!” 14 julio 2013, https://learn.adafruit.com/onion-pi. Fecha de acceso marzo 2021.

Lewis, Sarah Jamie, y MIT. “Onionscan.” 25 febrero 2017, https://onionscan.org. Fecha de acceso junio 2021.

Luzthedev, et al. “TorghostNG.” 2021, https://github.com/GitHackTools/TorghostNG. Fecha de acceso marzo 2021.

MushMush. “Conpot.” 2021, https://github.com/mushorg/conpot. Fecha de acceso abril 2021.

MushMush. “Glastopf.” 2019, https://github.com/mushorg/glastopf. Fecha de acceso abril 2021.

NTop Company. “NTopng.” 2021, https://www.ntop.org/products/traffic-analysis/ntop/. Fecha de acceso abril 2021.

Null byte. “Set Up a Vulnerable Target Computer with DV-Pi (Damn Vulnerable Pi) [Tutorial].” 4 octubre 2018, https://www.youtube.com/watch?v=fKOX4InkaGc. Fecha de acceso junio 2021.

Open Observatory of Network Interference (OONI). “Uncover evidence of internet censorship worldwide.” 2021, https://explorer.ooni.org. Fecha de acceso abril 2021.

Palabra de Hacker. “¿Cómo funciona Tor? - Entendiendo la red TOR.” 25 agosto 2020, https://www.youtube.com/watch?v=89p_4lsF5mo. Fecha de acceso abril 2021.

Pérez Esteso, Mario. “Que es y cómo funciona la red Tor.” 2016, https://geekytheory.com/que-es-y-como-funciona-la-red-tor. Fecha de acceso mayo 2021.

Pinero, G. “TOR: Primeros pasos con los Hidden Service.” 28 diciembre 2017, https://www.error509.com/2017/12/tor-primeros-pasos-con-los-hidden-service/ .

Px Mx y Foospidy. “Honeypy.” 2020, https://github.com/foospidy/HoneyPy. Fecha de acceso abril 2021.

Ramiro, Ruben. “Introducción a la comunicación segura - TOR, HTTPS, SSL.” 16 mayo 2018, https://ciberseguridad.blog/la-biblia-de-la-ciberseguridad/. Fecha de acceso abril 2021.

Raspap. “The easiest, full-featured wireless router setup for Debian-based devices.” 2019, https://raspap.com. Fecha de acceso marzo 2021.

Raspberry valley. “Raspberry Pi Tor Access Point.” 2015, https://raspberry-valley.azurewebsites.net/Raspberry-Pi-Tor-Access-Point/. Fecha de acceso Marzo 2021.

Reclaim Your Privacy (Cloak). “Eschalot.” 2019, https://github.com/ReclaimYourPrivacy/eschalot. Fecha de acceso mayo 2021.

Rodrigo, Alonso. “Las mejores alternativas a Raspberry Pi 4 que puedes comprar.” 2020, https://hardzone.es/tutoriales/reparacion/pc-fecha-hora-resetean-encender/. Fecha de acceso junio 2021.

Rodríguez, Francisco, et al. “Taller: Cambiando el CuenTOR.” 2 diciembre 2016, https://www.youtube.com/watch?v=PYu9Zkwmhw0. Fecha de acceso marzo 2021.

RSA Security LLC. 2021, https://www.rsa.com. Fecha de acceso abril 2021.

Sánchez Alés, José Miguel y Linuxnomicon. “6.1.3. DHCP con dnsmasq.” https://sio2sio2.github.io/doc-linux/06.infraestructura/02.dhcp/03.dnsmasq.html. Fecha de acceso marzo 2021.

Sayrafi, Mahrud y Cloudflare. “Presentamos el Resolutor de DNS para Tor.” 5 junio 2018, https://blog.cloudflare.com/es-es/welcome-hidden-resolver-es-es/. Fecha de acceso abril 2021.

Scott, Brenton. “Sakis3g-source.” 2014, https://github.com/Trixarian/sakis3g-source. Fecha de acceso marzo 2021.

Skavngr. “Rapidscan.” 2021, https://github.com/skavngr/rapidscan. Fecha de acceso mayo 2021.

Smith, Travis y TravisFSmith. “Sweetsecurity.” 2017, https://github.com/TravisFSmith/SweetSecurity. Fecha de acceso marzo 2021.

Sullo. “Nikto.” 2021, https://github.com/sullo/nikto. Fecha de acceso mayo 2021.

Tails non-profit organization. “Sistema operativo anonimizado.” 2021, https://tails.boum.org/index.es.html. Fecha de acceso abril 2021.

Tamminen, Upi y Desaster. “Kippo.” 2016, https://github.com/desaster/kippo. Fecha de acceso marzo 2021.

Telekom-security. “Tpotce.” 2021, https://github.com/telekom-security/tpotce. Fecha de acceso marzo 2021.

Test de velocidad. “Diferencias entre Deeb Web y Dark Web.” 7 octubre 2016, https://www.testdevelocidad.es/2016/10/07/deep-web-vs-dark-web/. Fecha de acceso mayo 2021.

Tor Project. “Torrc.” 2019, https://2019.www.torproject.org/docs/tor-manual.html.en. Fecha de acceso abril 2021.

Tor Project. “Tor: The Second-Generation Onion Router.” https://svn-archive.torproject.org/svn/projects/design-paper/tor-design.pdf. Fecha de acceso mayo 2021.

Tor Project Inc. “Manual.” 2019, https://2019.www.torproject.org/docs/tor-manual-dev.html.en. Fecha de acceso mayo 2021.

Tor Project Inc. “Services.” 2019, https://2019.www.torproject.org/docs/tor-onion-service. Fecha de acceso marzo 2021.

Tor Project Inc. “Tor Project Inc.” marzo 2021, https://www.torproject.org/es/.

Tor Project Lnc. “Tor Browser.” *https://www.torproject.org/download/*. Fecha de acceso marzo 2021.

Torre, Roman. “Configurando THERO: Crear un Hotspot wifi y conectarlo a la red TOR.” 5 noviembre 2016, https://www.romantorre.net/v3/2016/11/05/configurando-thero-crear-un-hotspot-wifi-y-conectarlo-a-la-red-tor/. Fecha de acceso abril 2021.

Ubuntu. “Manuals. Pppd.” 2019, http://manpages.ubuntu.com/manpages/bionic/man8/pppd.8.html. Fecha de acceso marzo 2021.

Ubuntu help. “Isc-dhcp-server.” 2015, https://help.ubuntu.com/community/isc-dhcp-server. Fecha de acceso marzo 2021.

UNIR | La Universidad en Internet. “¿cómo navegar en ella? | UNIR OPENCLASS.” 28 junio 2018. Fecha de acceso mayo 2021.

Velasco, Rubén y Redes zone. “Uniscan: conoce esta herramienta para buscar vulnerabilidades en cualquier aplicación web.” 23 febrero 2019, https://www.redeszone.net/2019/02/23/uniscan-buscar-vulnerabilidades/. Fecha de acceso mayo 2021.

Web ciencia. “EXPLICACIÓN DEL PROCESO INGENIERIL.” 2018, http://www.webciencia.es/index.php/articulos/212-explicacion-del-proceso-ingenieril. Fecha de acceso junio 2021.

WikiHow. “Cómo establecer un país en específico en el navegador Tor.” 2018, https://es.wikihow.com/establecer-un-país-en-específico-en-el-navegador-Tor. Fecha de acceso abril 2021.

Wireshark Foundation. “Wireshark.” 2021, https://www.wireshark.org. Fecha de acceso junio 2021.