UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Interfaz USB de red para acceso seguro basada en Raspberry Pi

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Jornet Calomarde, Raúl

Tutor: Ripoll Ripoll, José Ismael

Curso 2017-2018

Resumen

El proyecto aborda el desarrollo de un dispositivo hardware basado en «Raspberry Pi Zero W» que sirve de punto de acceso a red y que proporciona diversas herramientas orientadas a la seguridad, tales como: *firewall*, control de intrusos, configuración de seguridad inalámbrica, *proxies* o auditoría de nodos.

La conectividad del dispositivo con el ordenador personal se establecerá únicamente mediante un cable USB a micro USB y será compatible con la mayoría de sistemas.

El proyecto estará basado íntegramente en Raspbian (con posibles modificaciones del kernel) y herramientas libres y será publicado con una licencia GPL3.

Palabras clave

Periférico, Seguridad, Red, "Raspberry Pi", Hardware, GNU/Linux

Índice

1. Introducción

Este proyecto pretende aportar una solución personal para el acceso a redes de no confianza; desde redes públicas, hasta redes corporativas cuya monitorización se desee sortear. El resultado debe ser fácilmente distribuible y reproducible, así como sencillo de utilizar para personas sin un conocimiento técnico específico, pero que a su vez sea lo bastante potente para que las avanzadas puedan acceder a aspectos más complejos de la configuración.

1.1. Motivación

Existe una gran oferta y demanda de acceso a red en entornos no familiares, principalmente a redes públicas y corporativas cuyo método de administración se desconoce; esto expone a las personas que hacen uso de ellas a una monitorización no deseada por parte de quienes las administran o incluso de terceras personas que aprovechen la mala configuración de la red para realizar ataques de intermediario –Conocidos en inglés como *Man in the midle*–.

Existen diversas soluciones al alcance del público para protegerse de estas amenazas, pero a menudo no son intuitivas para personas con un nivel básico y elegir la adecuada requiere de un conocimiento concreto de redes y criptografía.

Actualmente no existe un dispositivo de las características de este proyecto. Los dispositivos de acceso a red orientados a seguridad que existen no ofrecen la versatilidad de este y, lo más importante, son privativos; esto significa que no se puede tener la certeza de que no operan de manera maliciosa, de la misma forma que lo puede hacer la red a la que se accede. Existe un pequeño abanico de productos comercializados que utilizan software libre pero son menos versátiles y su coste es elevado en comparación con los 15$ que supone el hardware que sustentará este proyecto.

1.2. Objetivos

El resultado deberá ser un dispositivo físico basado en la placa computadora –En inglés *single-board computer* – Raspberry Pi Zero W, un computador de tamaño reducido y bajo coste. Este se interpondrá entre la red y el computador de la persona usuaria, que tendrá que desactivar todos los medios de acceso a Internet para asegurar que la única vía de salida al exterior pasa por el dispositivo, que será nombrado *Secure-Network Access Device* –S-NAD– a partir de ahora.

La utilización será sencilla, la persona usuaria conectará el S-NAD a su equipo personal –*Host* a partir de ahora– mediante un cable *USB Standart Type A* macho a *USB Micro B* macho, que es un cable muy popular por ser el más utilizado para carga y transmisión de datos en teléfonos inteligentes. En unos segundos el S-NAD habrá iniciado los servicios pertinentes para gestionar la conexión y el sistema operativo del *host* informará de que se ha establecido una nueva conexión de red. A partir de este momento se deberá acceder a la interfaz de configuración, consistente en un servidor web alojado en el propio S-NAD, por medio de cualquier navegador. La primera página que se mostrará será una pantalla inicial que realizará una comprobación de red para determinar que el equipo no tiene acceso a Internet, si es correcta, comenzará a dar servicio de enrutamiento y habilitará toda la configuración al alcance de la persona usuaria.

Con esta visión global se procede a la enumeración concreta de los objetivos de forma jerárquica:

- Garantizar tanto la seguridad y robustez como la apariencia de tales. No basta con que sea seguro, la persona usuaria debe tener la sensación de que lo es.
  - Implementar una solución que no haga uso de ninguna tecnología privativa o de un origen sospechoso de no respetar los derechos de las personas usuarias.
  - No hacer uso de tecnologías que, aunque puedan ser estándar, puedan comprometer la seguridad del proyecto.
  - No implementar algoritmos originales que cubran los aspectos de seguridad. Se detallará a lo largo de la memoria el software confiable en que se delega esta.
- Universalizar el proyecto.
  - Implementar una solución multiplataforma. El dispositivo resultante ha de poder ser utilizado en cualquier sistema operativo sin instalar software adicional.
  - Implementar el frontend de configuración accesible desde cualquier navegador con soporte de HTML, CSS y JavaScript.
  - Implementar el frontend de forma accesible a personas con diversidad funcional visual.
    - Personas con daltonismo.
    - Personas con campo visual reducido.
    - Personas con pérdida de agudeza visual.
  - Asegurar el soporte de las tecnologías seguras y estándar.
  - Permitir que una persona sin conocimientos pueda entender y configurar el *frontend*.
- Obtener el resultado más eficiente posible.
  - Optimizar el consumo de recursos del *frontend* en el cliente.
  - Evitar el aumento de la latencia y la pérdida de ancho de banda en el enrutamiento.
  - Reducir todo lo posible el coste del resultado final.
  - Reducir todo lo posible el tamaño del software.
  - Reducir todo lo posible el consumo eléctrico.

1.3. Estructura

2. Estado del arte: expone el contexto actual de la seguridad en redes inalámbricas y el problema al que se pretende aportar una solución.
   2.1. Soluciones actuales: se detallan las principales soluciones que existe.
   2.2. Crítica al estado del arte: se analizan las soluciones expuestas en el punto anterior y se ponen de manifiesto sus carencias
   2.3. Propuesta: tomando como punto de partida las secciones anteriores, se plantea una solución nueva.
3. Análisis del problema: se realizan una serie de valoraciones con respecto al problema.
   3.1. Análisis de la seguridad
   3.2. Análisis de la eficiencia
   3.3. Análisis del marco legal y ético
   3.4. Análisis de riesgos
   3.5. Solución propuesta: se detalla la solución planteada anteriormente.
   3.6. Presupuesto: se detalla una estimación del presupuesto
4. Diseño de la solución: se pasa a detallar la solución propuesta.
   4.1. Arquitectura del sistema: se detalla la estructura *hardware* y *software*.
   4.2. Diseño detallado: se detalla el funcionamiento del dispositivo que se va a desarrollar.
   4.3. Tecnología: se detallan las tecnologías que se emplearán durante el desarrollo.
5. Desarrollo de la solución propuesta: se detalla la implementación de la solución.
   5.1. Configuración inicial del dispositivo S-NAD: se expone la configuración del dispositivo cuando se inicia.

1.4. Convenciones

Código fuente

El código fuente y el contenido de los archivos de configuración se representará en la línea de texto cuando sean referencias concretas a nombres de variables o métodos y como bloques en el caso de tratarse del contenido de estos. Se representará siempre con una tipografía de ancho fijo en color claro sobre oscuro tal y como se muestra en el `ejemplo`.

Entrecomillado

Se hará uso de las comillas latinas –«»– en lugar de las anglosajonas –“”– y se utilizarán para:

- Nombres propios de proyectos tal y como «GNU/Linux» o «Raspberry Pi».
- Cuando se nombre un término haciendo énfasis en la propia palabra o expresión en lugar de su significado.

Números decimales

Se utilizará el punto como separador decimal por ser más universal y como se expresa en el código.

Palabras extranjeras

Se hará uso del castellano en la medida de lo posible. Cuando se haga uso de un término más utilizado en una lengua extranjera, este se indicará en cursiva entre guiones la primera vez que aparezca en el texto. De esta forma la primera vez que aparece el término «puerta de enlace» se indicará de la siguiente manera: puerta de enlace –Nombrada en inglés *gateway*–; para pasar a ser referida desde ese momento siempre como «puerta de enlace». Palabras como «*frontend*» que no tienen una traducción normalizada se mantendrán en su idioma original y en cursiva.

Referencias bibliográficas

Se utilizará la norma de IEEE, indicándose entre corchetes el número de la entrada cuando se aporte respaldo bibliográfico. En la versión digital estarán enlazadas mediante hipervínculos.

Referencias internas

Cuando se haga referencia a una sección de la memoria se indicará su numeración y en la versión digital enlazará con esta.

Glosario

Se dispone de un glosario al final del documento. Las palabras que incluye este estarán resaltadas en azul y subrayadas de esta forma. En la versión digital estarán enlazadas mediante hipervínculos todas las apariciones.

2. Estado del arte

Se ha planteado parcialmente el estado actual de esta tecnología cuando se ha hablado de la motivación del proyecto.

Las tecnologías de acceso inalámbrico a la red suponen la forma más común de conectarse a Internet por parte del público general.

Como se ha evidenciado recientemente, los protocolos de seguridad que se utilizan en Wifi no son una garantía de seguridad y suponen una vulnerabilidad crítica, ya que al afectar a un estrato bajo de los que conforman los estándares de red, compromete todo lo que queda por encima: datos de navegación, del sistema, de aplicaciones, etc [1][2]. Este fallo no tendrá una solución sencilla al ser necesaria tanto la compatibilidad hacia atrás –hacia una tecnología no segura–, para no perder la comunicación con dispositivos antiguos, como la compatibilidad con el hardware, ya que los puntos de acceso inalámbricos son siempre sistemas embebidos con una electrónica limitada.

Esta endeblez no sólo supone un problema para quien administra la red, que puede perder la capacidad de controlar el acceso a su red, si no que pone en peligro a los equipos clientes de la red.

En la actualidad no se dispone de un protocolo que garantice la seguridad en redes wifi. El protocolo WEP presenta fallos de diseño básicos: la limitación de la profundidad de la clave y el uso continuado de esta sin alterar hacen de esta tecnología una herramienta peligrosa, ya que lejos de aportar una capa de seguridad a la conexión, sugiere una sensación de seguridad que no se corresponde con la realidad de su naturaleza [3].

El protocolo WPA –Llamado WPA2 en la normalización de IEEE [4]– mejora a su predecesor en la implementación de claves temporales –TKIP (*Temporal Key Integrity Protocol*)–, de forma que no se utiliza persistentemente la original, si no que esta es calculada para cada comunicación. No obstante, como es bien conocido, en octubre de 2017 se publicaron los detalles del que ha sido llamado ataque KRACK –*Key Reinstallation AttaCK*– que ofrece acceso total al tráfico a cualquier atacante.

De esta forma, se ha de llegar hasta el nivel de transporte para poder aplicar una capa de cifrado. En el caso de la web es sencillo, accediendo a servidores HTTPS en lugar de HTTP, pero para webs que no ofrecen esta posibilidad o servicios que no cifran su tráfico, la información sigue viajando en claro para cualquiera que haya violado la seguridad del nivel inferior. Y aún en el caso de que se aplique un cifrado, toda la información por debajo del nivel de aplicación sigue estando comprometida, esto es, la información del destinatario. Por ejemplificarlo, si se realizase una conexión HTTPS cifrada por TLS, la información que se facilita al servidor estaría protegida –Nombre de usuario, contraseña, correo electrónico, configuración, búsquedas en el sitio, etc.– pero se conocería las direcciones del sitio destino y del origen.

La amenaza no queda limitada a redes inalámbricas. Si bien suele ser el medio de conexión habitual de las redes públicas, pese a que el uso de una red cableada puede mitigar el problema, se sigue sin disponer de control sobre la infraestructura de la red, que puede estar recabando y almacenando información, tanto legalmente si se nos ha redirigido a una página con los términos de uso de la red y los hemos aceptado, como ilegalmente si no se nos ha informado.

Finalmente, los servidores que son accedidos también pueden obtener información a partir de nuestra IP pública.

2.1. Soluciones actuales

A la hora de valorar cómo afrontar el problema que aborda este proyecto, que es el de la seguridad en redes inalámbricas, se hará un repaso desde bajo nivel hacia arriba de las tecnologías implicadas en los escenarios previsibles. Todos responden al siguiente esquema:

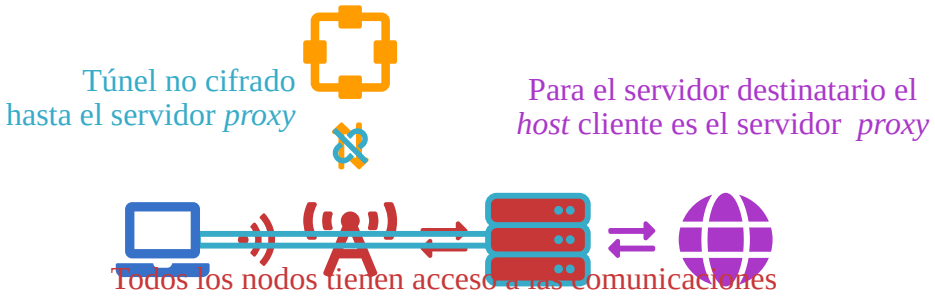
La eficacia de las soluciones se mide tomando en cuenta la privacidad –conocer información del origen y destino: dirección IP, sistema operativo, etc– y la protección frente a cada uno de los dispositivos de enrutamieto a los cuatro ataques descritos por Antonio Villalón Huerta [5]: interrupción –pérdida de los datos–, interceptación –acceso no autorizado a los datos–, modificación –además del acceso no autorizado se logra alterar los datos– y fabricación –envío de datos al servidor con apariencia similar a la del elemento atacado, haciéndolo difícilmente distinguible del elemento atacante–. Se supondrá en cada nivel que no existe más seguridad por encima, ya que si se está analizando por ejemplo el nivel de red y no existe ningún tipo de seguridad en esta capa pero por encima de este existe una conexión TLS punto a punto, la conexión sí sería segura.

Soluciones en el nivel de enlace –Seguridad Wifi–

Las primeras tecnologías que se pueden encontrar orientadas a seguridad pertenecen a la capa de enlace y son propias de Wifi. En el mejor de los casos se tratará de un protocolo WPA2 y en el peor ninguno. De cualquier forma, como se ha explicado en el apartado «Estado del arte», ya no se puede confiar en la seguridad de WPA2, por lo que en cualquier caso, la solución al problema no pasa por atajar la capa de enlace. Además, por el escenario en que se desarrolla el proyecto, se ha de suponer que no tenemos acceso a la configuración Wifi, por lo que se deberá trabajar con el protocolo que haya decidido la administración de la red. Por todo esto se considerará a nivel conceptual en el contexto de este trabajo que la seguridad a nivel de enlace es equivalente a una comunicación en claro quedando la persona usuaria expuesta de los ataques antes descritos.

Solución en el nivel de transporte –*proxy*–

A partir de aquí no existe diferencia con las redes cableadas. Aún por debajo de la capa de transporte, en la de enlace y la de red, se encuentra una solución parcial ya comentada anteriormente, los servidores *proxy*. Estos permiten ocultar al *host* como el origen de la comunicación. No obstante, los servidores *proxy* no aportan cifrado, por lo que, al igual que en el nivel de red, no existe protección contra los ataques.



Solución en el nivel de red –Red VPN–

La red VPN es una solución que se adecua mejor al problema que aborda este proyecto. Establece un túnel en el nivel de red, a diferencia del servidor *proxy*, este sí es cifrado. El protocolo para el cifrado más utilizado en VPN es Ipsec, aunque puede funcionar sobre cualquiera de la capa de red.

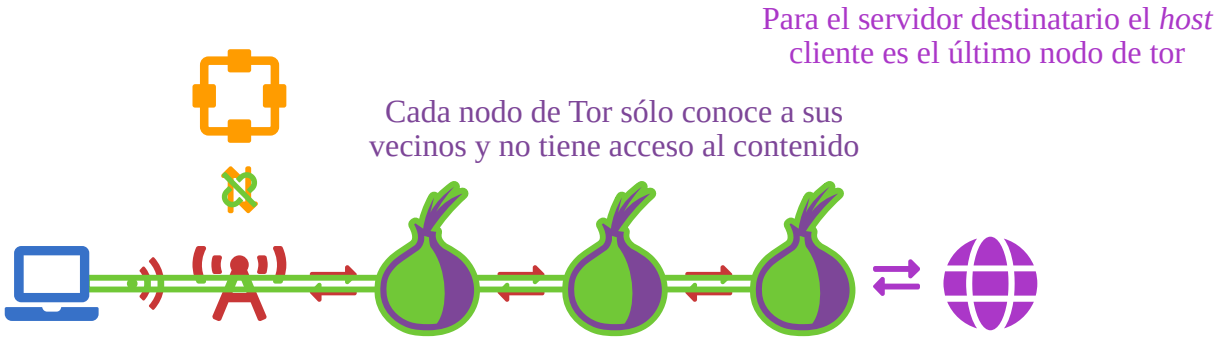
Establecer un túnel a través de una red VPN asegura los protocolos por encima, utilicen o no cifrado, haciendo sus paquetes incomprensibles para los nodos intermedios y, por tanto, protegiéndolos de los ataques de interceptación y modificación. Al establecerse una comunicación cifrada punto a punto entre el *host* y el servidor no es posible para un nodo intermediario simular tráfico del nodo original al estar firmado. No obstante sí puede interrumpirse el tráfico, aunque al estar cifrado no es posible conocer el contenido ni el tipo de comunicación, por lo que no es posible filtrarlo en base a ellos. Todo esto se aplica entre el *host* y el servidor VPN, una vez el tráfico sale de este último, el enrutamiento hasta el servidor destino viaja en claro. Al igual que el *proxy*, ocultar el origen, ya que el servidor destino recibe la petición desde la red virtual en lugar del *host*, lo que ofrece privacidad respecto al servidor destino.

La principal desventaja que ofrece una red VPN es que esta tiene acceso al contenido de las comunicaciones, por lo que elegirla correctamente es crítico, delegando así en una decisión del usuario su privacidad, ya que el servidor escogido siempre podrá llevar a cabo cualquiera de los cuatro ataques.

Solución en el nivel de aplicación –*proxy* Tor–

Existe otra alternativa, que es el uso de servidores *proxy* que operen sobre la red «Tor».

De esta forma, el *host* queda protegido de todos los elementos de enrutamiento y es anónimo al servidor destino



2.2. Crítica al estado del arte

En la siguiente tabla se muestran los ataques a los que es vulnerable cada tecnología y los datos que un elemento atacante podría obtener en rojo y en verde los que sí están protegidos:

	WPA2	*Proxy*	VPN	Tor
Elementos de enrutamiento	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino
Servidor del túnel	–No existe túnel–	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino	Interrupción Interceptación Modificación Fabricación Datos del *host* Datos del destino
Servidor destino	Datos del *host* Datos del destino	Datos del *host* Datos del destino	Datos del *host* Datos del destino	Datos del *host* Datos del destino

Visto lo anterior, la solución de que se dispone es el uso de servidores *proxy* y redes VPN, que aseguran una capa sobre todas las comunicaciones, protegiendo así los datos de las aplicaciones que no lo

hacen nativamente, y ocultando a los dispositivos de enrutamiento de la red el destino y al destino el origen.

Si bien técnicamente esta solución es correcta, la configuración adecuada para llevarla a cabo puede ser complicada, depender del sistema o requerir privilegios de que no disponga la persona usuaria y un error en esta puede desembocar en un fallo de seguridad.

Existen métodos más sencillos, como el uso de un servidor *proxy* en el navegador, que puede ser más asequible para personas sin un conocimiento específico, pero esa seguridad se limita al entorno del navegador, quedando fuera de su protección el resto de comunicaciones.

La mejor solución es aplicar la configuración del servidor *proxy* o de la VPN en un sistema robusto que redireccione todo el tráfico saliente de forma apropiada, la solución más obvia es disponer de un dispositivo externo que realice esta función. Un punto de acceso que conecte el *host* con el exterior.

Como se ha explicado anteriormente, las soluciones comerciales que existen no son versátiles, se especializan en una única tecnología y su coste es muy elevado.

2.3. Propuesta

La solución que se aporta es el desarrollo de un dispositivo seguro de acceso a red sencillo y completo.

La interposición de un dispositivo físico aporta una simplicidad y seguridad superior a la que se conseguiría mediante una configuración en el *host*. Separar el dispositivo que se conecta a la red del que representa y ejecuta la web lo protege de posibles ataques JavaScript. Aleja también del dispositivo de enrutamiento del entorno del equipo de la persona usuaria, que puede ser susceptible de malware –Software malicioso–, configuraciones inseguras, etc.

La simplificación en un sólo dispositivo que ejecuta únicamente el software imprescindible lo hace más robusto y seguro que un sistema personal. Además, al tratarse de un procesador con una arquitectura menos común puede minimizar su exposición a determinados ataques de bajo nivel.

3. Análisis del problema

3.1. Análisis de la seguridad

Las tecnologías orientadas a la seguridad se implementan mediante módulos, bibliotecas y software existentes y libres, ya que sería una negligencia pretender implementar de forma original estándares de seguridad. El uso de software libre tiene una implicación crítica y directa en la seguridad, ya que la implementación puede ser supervisada y auditada por una amplia comunidad, citando a Eric S. Raymond [6] «*Given enough eyeballs, all bugs are shallow*» –Con suficientes ojos, todos los errores son descubiertos–.

Las tecnologías que se requieren y el tipo de solución que se implementa se describen en la siguiente tabla:

Tarea	Tecnología	Software
Capa TLS del servidor web	TLS	stunnel4
Conectividad Wifi	WEP WPA WPA2 EAP PSK translation	wpa_suplicant wpa_cli
Enrutamiento y *Firewall*	*Firewall*	PEAP
Auditoría y análisis de red	Mixta –Nivel transporte–	nmap

3.2. Análisis de eficiencia

Como queda dicho en el apartado «Objetivos» en la «Introducción» uno de los objetivos es obtener un resultado eficiente en todos los aspectos.

Consumo energético

El consumo de la placa computadora Raspberry Pi Zero W oscila entre 100 mA cuando el procesador no está siendo usado –expresado *Idle* en inglés– y un máximo de 350 mA a máximo rendimiento y con periféricos conectados.

En el peor de los casos el consumo se mantendría por debajo de los 250 mA, ya que no se requiere un procesamiento gráfico y su único puerto USB se usará como esclavo, siendo entonces el consumo inferior a 1.25 W.

Eficiencia del sistema

Una forma de reducir el consumo del sistema es eliminando los módulos que no sean necesarios para la implementación del proyecto recompilando el núcleo –en inglés y más concretamente en el entorno Linux, *kernel*–. Además, según Dave Wreski [7], la revisión del las opciones de compilación dentro del grupo «Networking Options» pueden aportar seguridad en las comunicaciones.

Eficiencia en las comunicaciones

El *frontend* de configuración está compuesto de HTML, CSS y JavaScript –JS a partir de ahora–. Los archivos de cada lenguaje están separados en distintos archivos, es decir, los archivos HTML enlazan a los estilos CSS y los *scripts* JS en lugar de incluir el código embebido. Para reducir la carga en las comunicaciones, la información real viaja en el ficheros XML mediante peticiones AJAX, siendo el resto de lenguajes los encargados de presentar y tratar la información. De esta forma, el único tráfico será el de la actualización del contenido, ya que el servidor HTTP indica al cliente web que los documentos no caducan, siendo innecesario solicitarlos cada vez que se enlazan.

3.3. Análisis del marco legal y ético

Análisis de la protección de datos

Dado que en todo momento la persona usuaria es propietaria y administradora de los servicios, nadie salvo ella será la responsable de los datos.

Propiedad intelectual

El proyecto finalizado se distribuirá libre bajo una licencia GPL3[a] con el fin de hacer llegar esta tecnología al mayor público posible y contribuir al desarrollo colaborativo y horizontal.

Se emplean únicamente dos bibliotecas no nativas. Estas son «ping.js», desarrollada por Alfred Gutierrez y distribuida con una licencia MIT[b]; y «ua-parser-js», desarrollada por Faisal Salman y distribuida con una licencia dual GPL2[c] y MIT[b]. Ambas bibliotecas tienen licencias compatibles con GPL3[a]

Además de software, se emplean las fuentes de iconos y el CSS de Font Awesome[d]. Esta herramienta tiene un conjunto de licencias compatibles con GPL3[a]. Los iconos, empleados también en esta memoria, tienen una licencia CC-BY 4.0[e]. Las tipografías, empleadas en el *frontend* están licenciadas bajo una OFL[f]

Ética

Proporcionar anonimato siempre supone una implicación ética. Muchas de las tecnologías que se utilizan en este proyecto son utilizadas para burlar los controles de los cuerpos policiales y cometer actos ilícitos. No obstante, son utilizadas de la misma forma por quienes tratan de sortear el seguimiento que llevan a cabo muchas empresas, por periodistas en zonas de conflicto, por quienes llevan a cabo denuncia social o quienes no pueden hacer uso de su libertad de expresión. Son tecnologías que se enfrentan a la dicotomía entre la seguridad y la libertad.

a https://www.gnu.org/licenses/gpl-3.0.en.html
b https://opensource.org/licenses/MIT
c https://www.gnu.org/licenses/gpl-2.0.en.html
d https://fontawesome.com/
e https://creativecommons.org/licenses/by/4.0/
f http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL

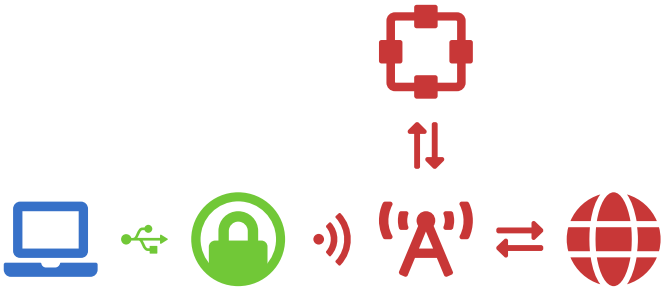
Si bien esta polémica puede estar relacionada con el proyecto, este no ofrece una tecnología nueva, sólo acerca los medios que ya existen a un público que estaba excluido de ella por su nivel técnico.

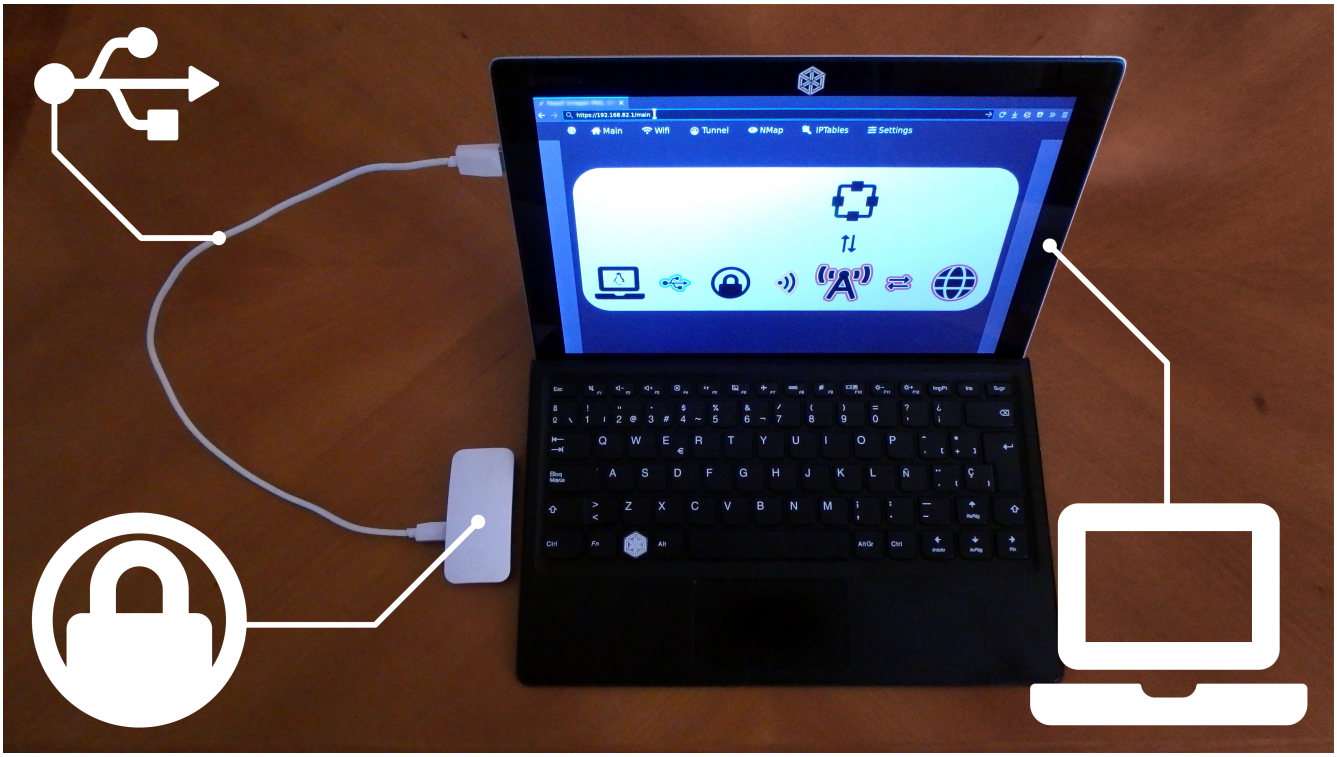
3.4. Análisis de riesgos

De acuerdo con la licencia de distribución –GPL3–, se indicará a la persona usuaria que no existe garantía, eximiendo a la parte desarrolladora de la responsabilidad del funcionamiento del proyecto.

3.5. Solución propuesta

Todas las soluciones propuestas pasan por una configuración avanzada en el equipo *host* y privilegios de administrador. La introducción de un dispositivo físico independiente puede aportar una solución a este problema. Si se interpone un punto de acceso a la red sobre el que se tenga control –Diseñado ex profeso–.





3.6. Presupuesto

El presupuesto es variable según la calidad, capacidad de la tarjeta y gastos de envío. Suponiendo una tarjeta de 2GB –Más que suficiente para albergar el proyecto– y obviando los gastos de transporte, el presupuesto estimado es de unos 15USD –Dólares estadounidenses– a fecha de Junio de 2018.

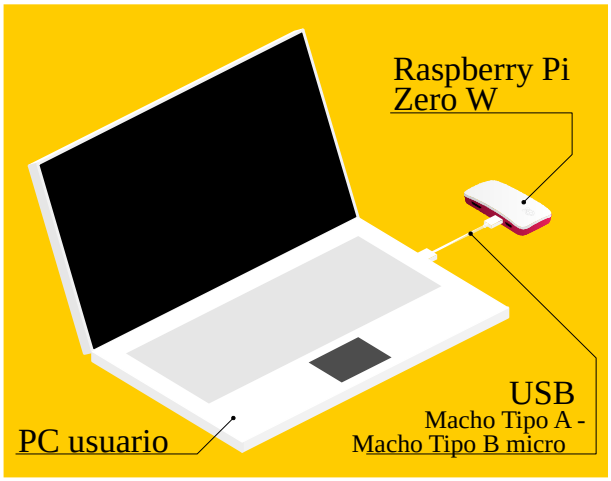
Producto	#	Precio
Raspberry Pi Zero W	1	10.00 USD
2GB microSD	1	2.50 USD
Cable USB Standart Type A macho a USB Micro B macho	1	2.00 USD
Total		14.50 USD

4. Diseño de la solución

4.1. Arquitectura del sistema

Elementos hardware

El proyecto se compone de una placa computadora «Raspberry Pi Zero W» que se conecta mediante un cable USB macho tipo A –*Host*– macho tipo B micro –«Raspberry Pi Zero W»–.



El estándar USB establece una topología en árbol, con un sólo dispositivo maestro –Incorrecta traducción del inglés *master*– y uno o varios esclavos –Del inglés *slave*–. El rol del esclavo se ajusta a los dispositivos que actúan como periféricos, siendo el maestro el dispositivo que hace uso de ellos. En la conexión de un cable USB que conecte un equipo de sobremesa con un teclado es la computadora la que actúa como maestro, aceptando incluso más periféricos en el mismo puerto USB si se utiliza un *hub*, siendo todos ellos esclavos.

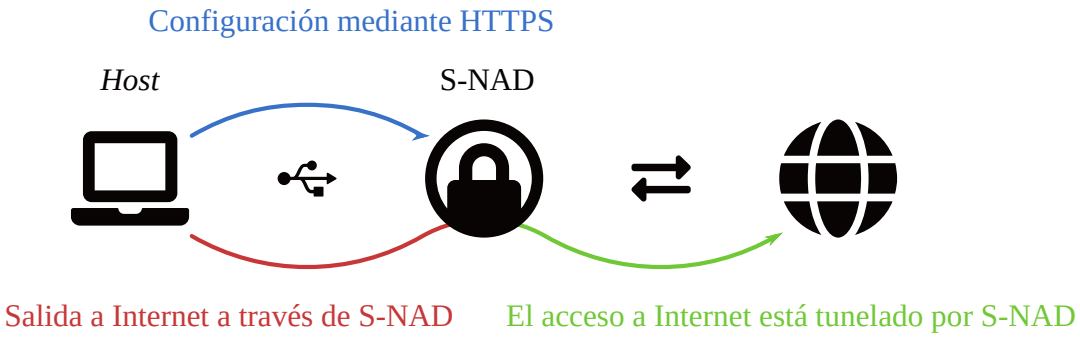
El puerto tipo A es el más común, todos los equipos personales disponen de él y tiene siempre un rol maestro. El puerto tipo B micro es el más común en los teléfonos inteligentes y puede, según la electrónica detrás de él, actuar como maestro o esclavo, aunque lo más común es que pueda actuar como ambos según el protocolo, esta tecnología recibe el nombre de OTG. En el caso del puerto de la «Raspberry Pi Zero W» soporta OTG, tal como se especifica en el manual de la placa [8] de la misma. Los teléfonos inteligentes por ejemplo pueden actuar como maestros si se les conecta un periférico como un ratón o un teclado o como esclavos si se utilizan conectados a otro dispositivo –Que actuará como maestro– para compartir red o permitir acceder a su almacenamiento interno.

Al igual que los teléfonos inteligentes, la «Raspberry Pi Zero W» puede alternar el rol de su único puerto USB –Una hembra tipo B micro– para operar como maestro o como esclavo. En este proyecto, la placa computadora actuará siempre como dispositivo esclavo ya que cumple el rol de dispositivo periférico, concretamente de adaptador *ethernet*, y la maestra es la computadora de la persona usuaria. La tecnología utilizada para que la «Raspberry Pi Zero W» pueda actuar como un adaptador ethernet es *Ethernet over USB* – Literalmente *ethernet* sobre USB– y el protocolo concreto es RNDIS.

Una de las limitaciones del estándar USB es indispensable para garantizar la seguridad del sistema. Esto implica que sólo puede existir un dispositivo maestro, de esta forma se garantiza que una conexión RNDIS levantada por la «Raspberry Pi Zero W» sólo podrá tener un maestro, haciendo físicamente imposible que un dispositivo común pueda acceder a la comunicación aunque se utilice un *hub* –Nada recomendable igualmente–. Podría utilizarse algún dispositivo intermedio para registrar el tráfico pero no sería sutil y la persona usuaria lo percibiría claramente. Además, el tráfico con el *frontend* está cifrado.

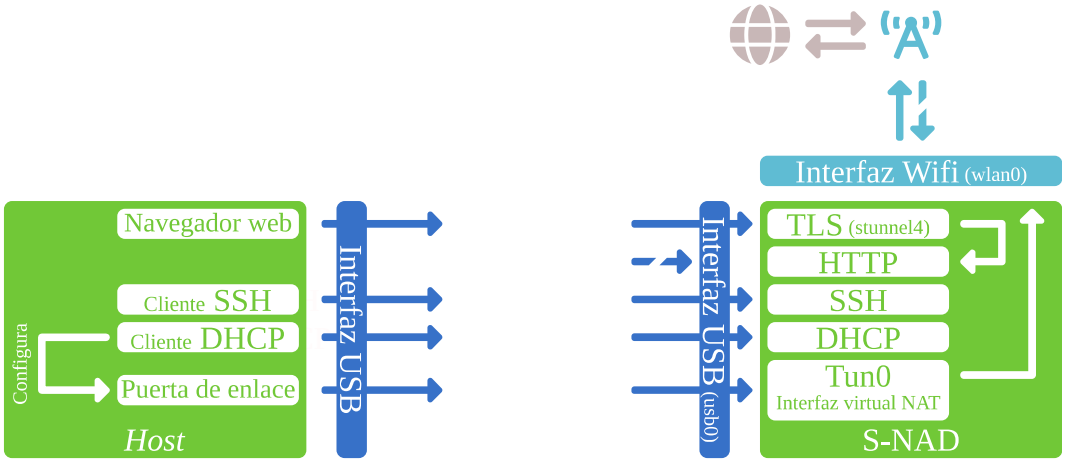
Estructura del software

El software desarrollado para este proyecto consiste en un servidor web que permite configurar el comportamiento de S-NAD

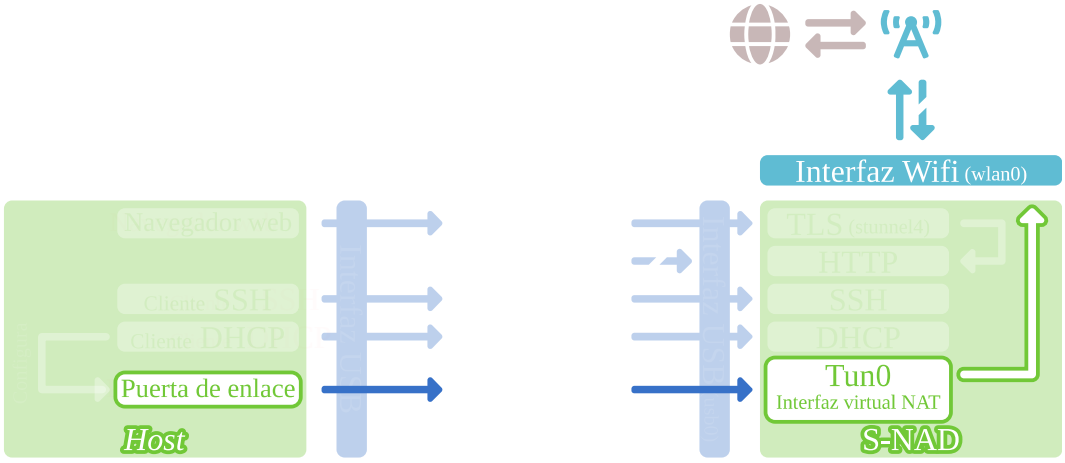


4.2. Diseño detallado

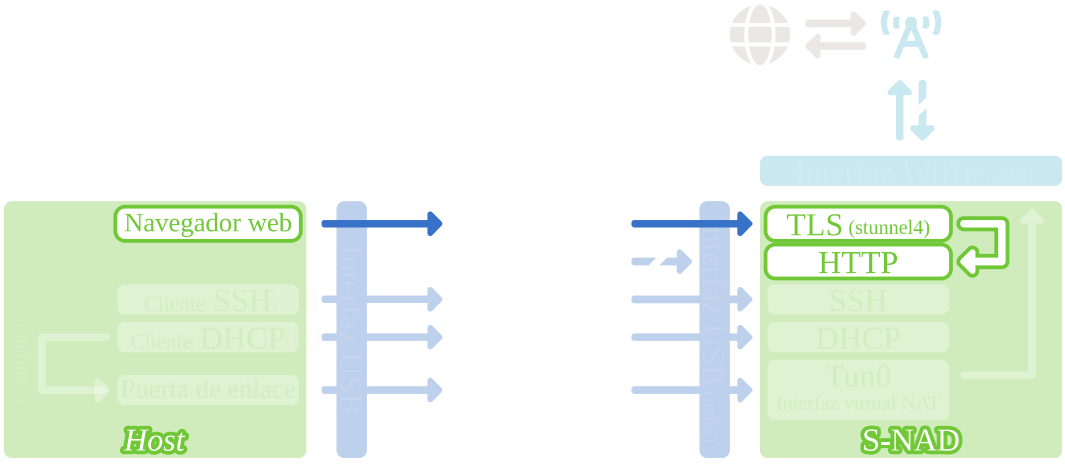
S-NAD actúa como un entutador –*router* en inglés– tradicional con los servicios propios de este que se describen a continuación por orden de uso en un primer contacto de una máquina clienta.



En primer lugar actúa RNDIS, que permite tanto a S-NAD como al *host* identificar sus puertos USB como dispositivos de red. En ese momento, si la persona usuaria del equipo no ha configurado la interfaz de red, esta debería por defecto tratar de configurarse por medio de DHCP, para lo que S-NAD dispone de un servidor que le proporciona una dirección IP y se establece a sí mismo como puerta de enlace. Para gestionar las peticiones del *host* al servidor se requiere una traducción de dirección IP privada, desde la que se realiza la petición, a la dirección IP –Presumiblemente privada también– que tiene S-NAD en la red a la que se conecta mediante Wifi. Esta tarea la lleva a cabo un servidor NAT alojado en el S-NAD.



Aquí termina la parte automática y será cuando la persona usuaria comience a solicitar los posibles servicios. El principal es el *frontend* de configuración, que se accede mediante HTTPS. Para garantizar la seguridad se delega la parte del túnel TLS en el servicio nativo de «Linux» «stunnel4» por lo que el servidor web consta realmente de dos servicios: por un lado un servidor HTTP desarrollado en «Python» íntegramente para este proyecto y una instancia de «stunnel4» que tunela las peticiones que recibe en el puerto 443 hacia el puerto 80 en *localhost* –Alias de 127.0.0.1 que dirige a la máquina local–.



4.3. Tecnología

Protocolos utilizados

Los protocolos que se emplean en las comunicaciones entre el *host* y el dispositivo S-NAD son, por orden de uso desde que se inicia: RNDIS –utilizado para que S-NAD actúe como un adaptador ethernet– y DHCP.

Aunque el soporte RNDIS está muy extendido, algunos sistemas pueden ocasionar problemas:

- GNU/Linux: todos los sistemas de escritorio lo incluyen. Algunas distribuciones de servidor u orientadas a tareas específicas pueden no incluir el soporte por defecto, en este caso debe instalarse el controlador desde los repositorios.
- Mac OS: las versiones actuales incluyen el controlador, si no, debe instalarse manualmente.
- Windows: puede no incluirse según el fabricante, en este caso debe instalarse el controlador a mano.

El acceso al *frontend* de lleva a cabo por medio de HTTPS, que se implementa de forma separada como un servidor HTTP ordinario y un túnel TLS.

Por último, los protocolos desde el dispositivo S-NAD hacia el exterior y que son ordenado y configurados desde el *frontend* por el *host* son los que implementan la seguridad Wifi, tales como: WPA, WEP, PEAP, etc. Y los protocolos para el tunelado posterior se emplea OpenVPN, *Proxies* y el servicio «Tor».

Lenguajes empleados en el desarrollo

Se emplea Python –ejecutado por el intérprete Python 3.7.5– para el desarrollo del servidor web. Se utilizan algunos módulos nativos de Python como `os`, `re`, `time`, `urllib`, etc.; aunque el más destacable es `http.server`, que permite la gestión de las peticiones HTTP.

Por otra parte el *frontend* se desarrolla utilizando únicamente las tecnologías más estándar y soportadas por todos los navegadores: XHTML, CSS y JavaScript.

Todos los navegadores modernos soportan estas tecnologías, aunque se recomienda el uso de Mozilla Firefox u otro libre por motivos de seguridad y privacidad.

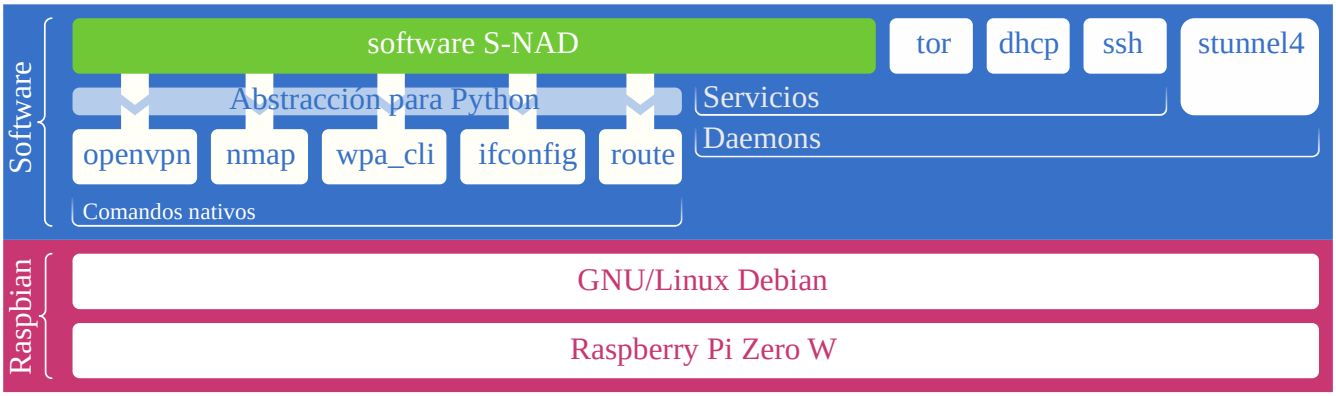
5. Desarrollo de la solución propuesta

La solución propuesta permite, por medio del *frontend*, conectar el dispositivo S-NAD a una red Wifi, configurar distintas opciones de tunelado y realizar análisis de red contra otros dispositivos.

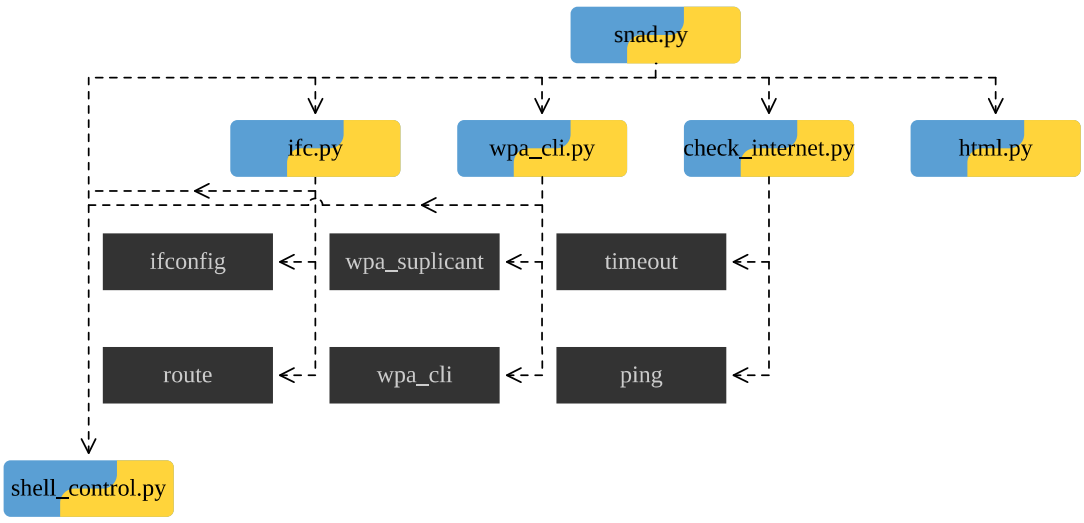
5.1. Configuración inicial del dispositivo S-NAD

El sistema se construirá sobre el sistema «Raspbian», un sistema basado en «Debian» preparado para operar en computadoras «Raspberry Pi».

Se hará uso de distintos *daemons* y comandos nativos del sistema. El programa principal en «Python» –Servidor web del *frontend*– interactuará con estos por medio de unos módulos diseñados ex profeso para el proyecto. En el siguiente diagrama se puede apreciar la estructura del sistema operativo en el dispositivo S-NAD:



Le estructura del software desarrollado –representado en verde en el diagrama anterior– se detalla en el siguiente diagrama:



El dispositivo comienza con todos sus servicios levantados y con una configuración que le permite operar totalmente, no obstante, algunas capacidades se bloquean temporalmente a la espera de instrucciones de la persona usuaria o comprobaciones de red por motivos de seguridad. Este es el caso del enrutamiento, que no inicia hasta que no se recibe una señal del *host* que indica que está aislado de la red; o el caso también de la conexión Wifi en que se limita intencionadamente el funcionamiento de «wpa_suplicant» para que no inicie conexiones de forma automática y que tenga que ser la persona usuaria la que las active.

Configuración IP

Para la configuración de red se toman en cuenta las únicas dos interfaces de red que se encuentran en el dispositivo.

Se configura la interfaz USB con parámetros estáticos. Se decide que la dirección de la red local que se establece entre el *host* y S-NAD será 192.168.82.1/24

```
allow-hotplug usb0
iface usb0 inet static
        address 192.168.82.1
        netmask 255.255.255.0
        network 192.168.82.0
        broadcast 192.168.82.255
```

A la interfaz Wifi se le indica recibir la configuración por medio de DHCP.

```
auto wlan0
iface wlan0 inet dhcp
```

Servicios nativos

DHCP y DNS

Se utiliza el software nativo «isc-dhcp-server» para proporcionar los servicios DHCP y DNS [9]. Se asignan IPs en el rango entre 192.168.82.2 y 192.168.82.254. Establece la puerta de enlace como su propia dirección. Se utilizan los DNS de «OpenDNS» en lugar de los que sugiera la red Wifi.

```
subnet 192.168.82.0 netmask 255.255.255.0 {
    option routers 192.168.82.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.82.255;
    option domain-name-servers 208.67.222.222 208.67.220.220 208.67.222.220 208.67.220.222;
    range 192.168.82.2 192.168.82.254;
}
```

Cortafuegos y enrutamiento

Marcus J. Ranum define un cortafuegos –*firewall*– como un sistema para establecer una política de control de acceso entre dos redes [10]. En este caso, es importante señalar el carácter de cada una de estas. Por un lado se tiene la que se establece entre las interfaces USB del *host* y el dispositivo S-NAD que sólo puede tener conectados estos dos nodos y la red inalámbrica cuya configuración y topología son totalmente desconocidas. Esta última es la que entraña más dificultad

Se hace uso del *framework* –que se puede traducir como «entorno de trabajo»– «Netfilter», propio del núcleo Linux. Se utilizan para este propósito reglas de filtrado mediante el comando nativo «Iptables» en las que no se va a profundizar, para más detalles [11]. La política general es de ACCEPT.

Configuración de seguridad

Se sigue la máxima descrita por D. Brent Chapman y Elizabeth D. Zwicky [12] de mínima complejidad y máxima seguridad.

Puesto que la conexión USB tiene como único cliente el *host*, se confía en esta totalmente. No pueden darse ataques de intermediario, sólo existen en la red el *host* y S-NAD, por lo que no se aplican reglas de ningún tipo.

Por otra parte, en el caso de la interfaz Wifi existen múltiples peligros potenciales. Dado que en ningún caso es deseable que ninguna máquina externa pueda solicitar servicios a S-NAD, se bloquean todas las peticiones entrantes. Las respuestas a las solicitudes que haga S-NAD como cliente sí serán atendidas a pesar del bloqueo ya que una regla de enrutamiento posterior permite la entrada de conexiones previamente iniciadas.

```
# iptables -A INPUT -i wlan0 -j DROP
```

Con esta configuración, S-NAD es invisible a la red Wifi a la que se conecta. Al realizar un escaneo Nmap contra él no se obtiene ninguna respuesta, obteniendo el mismo informe que se obtendría contra una IP no asignada. El informe recomienda utilizar la opción -Pn –Según la documentación de NMap [13] «*Treat all hosts as online -- skip host discovery*», en castellano «Tratar todos los hosts como conectados -- Saltar detección de hosts»–, el informe sigue siendo el mismo que se obtendría contra un host inexistente. El único inconveniente es que el escaneo tarda mucho más en finalizar que contra un *host* ficticio, esto puede ocurrir porque los nodos de la red recuerden las peticiones de S-NAD. En los anexos pueden encontrarse ambos informes.

También se prohíbe el acceso al puerto 80 desde el *host*, ya que debe emplear la conexión HTTPS.

```
# iptables -A INPUT -i usb0 -p tcp --dport 80 -j REJECT
```

La política es de rechazo –*REJECT*– en lugar de ignorar –*DROP*– ya que no se pretende ocultar el servicio al *host*.

Enrutamiento

Para permitir el reenvío –del inglés *forwarding*– de las peticiones ha de modificarse la configuración de dos archivos. Sobrescribir el contenido de `/proc/sys/net/ipv4/ip_forward` por `1` y descomentar la línea `net.ipv4.ip_forward=1` en el archivo `/etc/sysctl.conf`.

Para que las peticiones del *host* sean encaminadas al exterior a través de la red Wifi se incluyen las siguientes reglas en «Iptables».

```
# iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
# iptables -A FORWARD -i wlan0 -o tun0 -j ACCEPT
# iptables -A FORWARD -i tun0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Túnel TLS para el servidor HTTP

Los parámetros para tunelar el servicio HTTP por TLS son sencillos y se resume con el archivo de configuración de «stunnel4».

```
cert = server.pem

setuid = 0
setgid = 0

foreground = yes

[https]
accept = 192.168.82.1:443
connect = 127.0.0.1:80
```

5.2. Métodos remotos de S-NAD desde el *frontend*

La persona usuaria puede, por medio del *frontend*, interactuar puede interactuar con S-NAD mediante determinados métodos que le permiten obtener información y cambiar su comportamiento.

Esto se lleva a cabo mediante peticiones AJAX que envían los parámetros requeridos por medio de peticiones GET y recibe el resultado del servidor por medio de respuestas XML.

Métodos para la obtención de información

Comprobar conexión a Internet

Se puede solicitar a S-NAD información sobre el estado de la conexión a Internet. La respuesta consta de un único campo que es el estado codificado como un sólo carácter numérico: «0» para indicar que S-NAD no está conectado, «1» para indicar que está conectado a una red inalámbrica pero no hay acceso a Internet y «2» para indicar qué está conectado y tiene acceso a Internet.

El formato de la respuesta tiene esta estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<STATUS>0<STATUS>
```

Comprobar configuración de enrutamiento

El *host* puede preguntar a S-NAD si está preparado el servicio de enrutamiento, ya que por defecto no está habilitado. La respuesta utiliza códigos similares a los de respuesta HTTP.

Código	Significado en HTTP	Significado en S-NAD para esta petición
503	*Service Unavailable* –Servicio no disponible–	No se está enrutando.
102	*Processing* –Procesando–	Se está llevando a cabo la configuración de enrutamiento.
200	OK	La configuración de enrutamiento se ha llevado a cabo.

El formato de la respuesta tiene esta estructura.

```
<?xml version="1.0" encoding="UTF-8"?>
<STATUS>200<STATUS>
```

Estado Wifi

S-NAD puede solicitar a «wpa_suplicant» el estado de la conexión Wifi, a su vez, desde el *frontend* se puede demandar esta información. La respuesta a la petición es un único elemento `STATUS` con los campos del estado como atributos.

Ejemplo –Desconectado–

```
<?xml version="1.0" encoding="UTF-8"?>
<STATUS
  uuid="01234567-89ab-cdef-0123-456789abcdef"
  wpa_state="INACTIVE"
  p2p_device_address="ff:ee:dd:cc:bb:aa">
</STATUS>
```

Si el dispositivo S-NAD no está conectado a ninguna red la respuesta tendría un aspecto similar a este:

Escaneo Wifi

Desde el *frontend* se puede solicitar un escaneo de Wifi. Cuando el servidor recibe una petición solicita a «wpa_suplicant» por medio de «wpa_cli», abstraído por el módulo «Python» wpa_cli.py desarrollado para este proyecto, un escaneo que se presenta en formato XML.

En la raíz existen dos nodos: uno `SCAN`, que contiene el resultado del escaneo, y uno `STATUS`, que contiene el estado Wifi de «wpa_suplicant» en atributos. El elemento `SCAN` contiene un elemento `NETWORK` por cada SSID distinto en las redes detectadas, que a su vez contiene todos los puntos de acceso detectados que se identifican con ese SSID en subelementos `AP`. Los elementos `AP` tienen como atributos: el bssid –`bssid`–, la frequencia –`freq`–, intensidad de la señal –`level`–, las etiquetas decodificadas –`flags`–, el ssid –`ssid`– y si está recordada el psk o la clave del punto de acceso –`saved`–.

Ejemplo

Si S-NAD no está conectado y en el alcance de su antena Wifi se encuentran estos 4 puntos de acceso:

BSSID	Freq.	Señal	Seguridad	SSID	Record.
`01:23:45:67:89:ab`	2.422 GHz	82%	WEP	\x00\x00\x00\x00 –Oculto–	No
`02:46:8a:ce:02:46`	2.422 GHz	34%	WPA2	\x00\x00\x00 –Oculto–	No
`00:11:22:33:44:55`	2.427 GHz	83%	WPA/WPA2	Red de casa	Sí
`00:ff:11:ee:22:dd`	2.462 GHz	22%	Ninguna	FREEWIFI	No

La respuesta a la petición sería similar a esta:

```
<?xml version="1.0" encoding="UTF-8"?>
<SCAN>
  <NETWORK ssid="">
    <AP
      bssid="01:23:45:67:89:ab"
      freq="2422"
      level="0.82"
      flags="WEP ESS"
      ssid="\x00\x00\x00\x00"
      saved="0"></AP>
    <AP
      bssid="02:46:8a:ce:02:46"
      freq="2422"
      level="0.34"
      flags="WPA2-PSK-CCMP ESS"
      ssid="\x00\x00\x00"
      saved="0"></AP>
  </NETWORK>
  <NETWORK ssid="Red de Casa">
    <AP
      bssid="00:11:22:33:44:55"
      freq="2427"
      level="0.83"
      flags="WPA-PSK-CCMP WPA2-PSK-CCMP WPS ESS"
      ssid="Red de Casa"
      saved="0"></AP>  </NETWORK>
  <NETWORK ssid="FREEWIFI">
    <AP
      bssid="00:ff:11:ee:22:dd"
      freq="2462"
      level="0.22"
      flags="ESS"
      ssid="FREEWIFI"
      saved="0"></AP>
<STATUS
  uuid="01234567-89ab-cdef-0123-456789abcdef"
  wpa_state="INACTIVE"
 p2p_device_address="ff:ee:dd:cc:bb:aa">
</STATUS>
```

Métodos para ordenar cambios en la configuración

Por último pueden enviarse órdenes para que S-NAD las lleve a cabo.

Indicar aislamiento

Al inicio del servicio se comprueba en el *frontend* que el *host* está aislado de Internet. Una vez confirmado que no parece que existan otras redes, se informa a S-NAD, que es respondida con códigos similares a los códigos de respuesta HTTP.

Código	Significado en HTTP	Significado en S-NAD para esta petición
102	*Processing* –Procesando–	Iniciando la configuración de enrutamiento
200	OK	S-NAD estaba enrutando antes de la solicitud

El formato de la respuesta tiene esta estructura.

```
<?xml version="1.0" encoding="UTF-8"?>
<STATUS>102<STATUS>
```

Conectar a una red Wifi

Desde el *frontend* se puede solicitar la conexión a una red Wifi. En este caso, se requieren algunos parámetros para indicar la información necesaria para llevarla a cabo. Estos son:

ssid		
<passphrase en claro>	**saved**=psk	
	saved=wep_key	
	passphrase=<passphrase en claro>	[**save**=true]
	wpa_key=<clave wep en claro>	

De esta forma, accediendo a la URL `/wifi?action=connect&ssid=Wifi01saved=psk`, S-NAD trataría de conectar con la red «Wifi01» con el psk que tuviese almacenado.

Si se solicitase la dirección `/wifi?action=connect&ssid=Wifi01&passphrase=pass012345678`, intentaría conectarse con la misma red pero con la contraseña indicada –pass012345678–.

Por último, si se realiza una petición igual a la anterior pero incluyendo el parámetro «save» `/wifi?action=connect&ssid=Wifi01&passphrase=pass012345678&save=true` S-NAD realizará la misma conexión, la diferencia es que se almacenará el PSK para poder conectarse posteriormente mediante el atributo «saved»

La respuesta no es inmediata, se envía progresivamente el estado conforme avanza el proceso de sincronización con el punto de acceso: 102 –iniciada la sincronización–, 403 –contraseña incorrecta– o la dirección IP de S-NAD cuando finaliza la sincronización.

Desconectar la red Wifi

Junto con la orden anterior existe otro comando para desconectar la red.

Configuración del tunelado

La orden para modificar la configuración de tunelado requiere de bastantes parámetros que se detallan a continuación.

tecnology	type	
none		
proxy	tor	
	custom	**http**=<ip:puerto>
		ssl=<ip:puerto>
		ftp=<ip:puerto>
		socks=<ip:puerto>
vpn	tls	**ac_cert**=<certificado de la AC>
		user_cert=<certificado de usuario>
		priv_key=<clave privada>
		pass=<contraseña de la clave privada>
	key	**key**=<clave>
		addess=<dirección>
		local=<IP local>
	pass	**ac_cert**=<certificado de la AC>
		user=<nombre de usuario>
		pass=<contraseña de usuario>
	pass+tls	**ac_cert**=<certificado de la AC>
		user_cert=<certificado de usuario>
		priv_key=<clave privada>
		pass=<contraseña de clave privada>
		user=<nombre de usuario>
		pass=<contraseña de usuario>

Ejecutar «NMap»

Existen tres peticiones que se pueden realizar para ejecutar y obtener los informes de «NMap».

La primera no está diseñada para ser accedida mediante AJAX sino que lo hace el navegador, ya que devuelve la página XHTML que muestra el progreso del comando y que llama mediante JS a la orden que invoca realmente a «NMap». Para ello se proporciona únicamente un parámetro, que es el comando «NMap» que se debe ejecutar. Se filtra la entrada en el servidor para evitar inyecciones.

La segunda orden se invoca por medio de AJAX automáticamente desde la página respondida por la primera orden y ejecuta el comando en S-NAD para posteriormente recibir su salida. La respuesta a

esta petición no es instantánea, sino que se envía progresivamente conforme la ejecución del comando devuelve su salida. Esta respuesta es tratada y representada por la página que invoca la petición, que fue generada en la llamada descrita en el párrafo anterior.

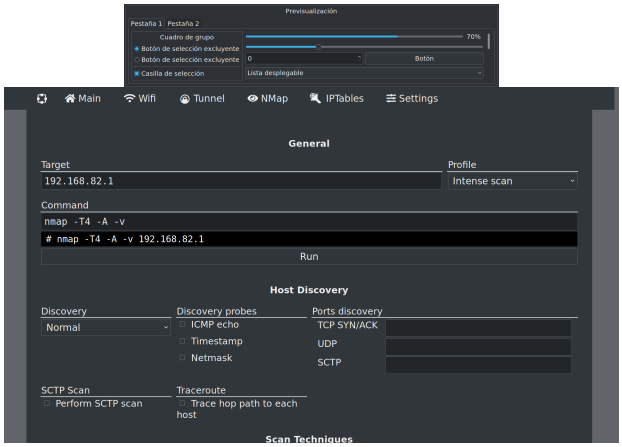
Finalmente, cuando termina la ejecución del comando, la página redirige automáticamente a la dirección del informe que ha generado «NMap», en formato XML, con una hoja de estilo para hacerla comprensible para la persona usuaria.

5.3. Frontend

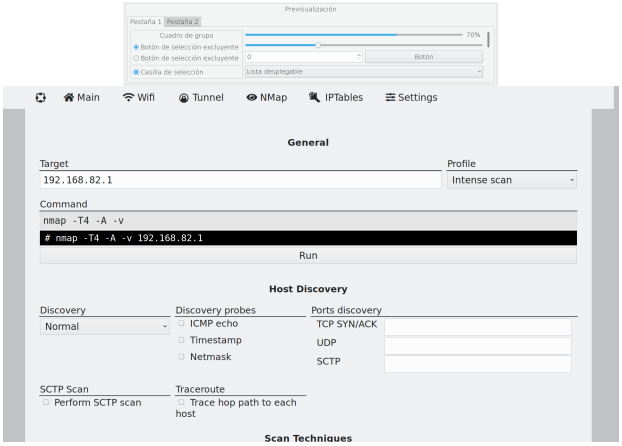
Como ya se ha expuesto, las tecnologías utilizadas son exclusivamente XHTML, CSS y JS. Como regla general, a lo largo de todo el desarrollo del *frontend* se utilizarán los colores y las tipografías definidas en el sistema operativo del *host* en lugar de decidir unos fijos en el desarrollo para garantizar la accesibilidad. De esta forma, si la persona usuaria tiene una configuración de alto contraste, con una tipografía específica configurada en su sistema, todo el *frontend* responderá a dicha configuración.

Estos son los temas por defecto de los escritorios KDE –«Breeze»–:

Tema oscuro

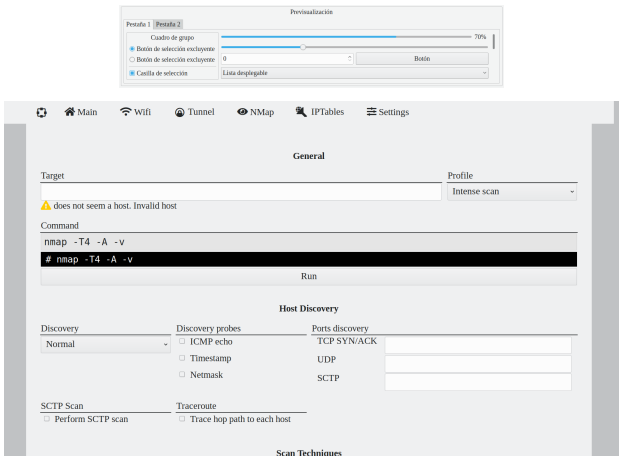


Tema claro

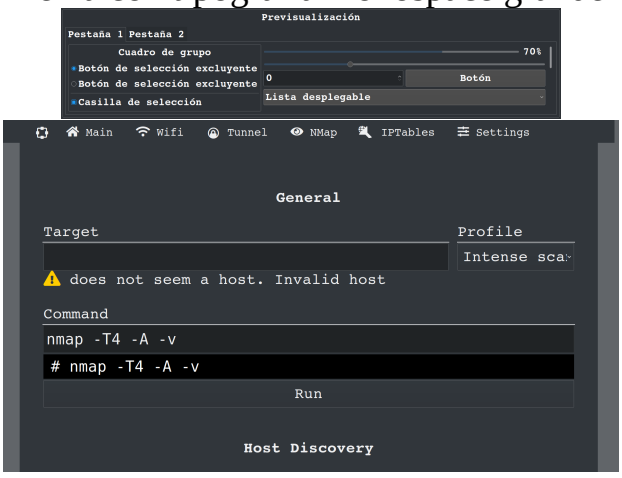


Estos son otros dos ejemplos con temas menos comunes. Por un lado uno utilizando una tipografía «serif» y otro con una monospace de 24px:

Tema con tipografía serif



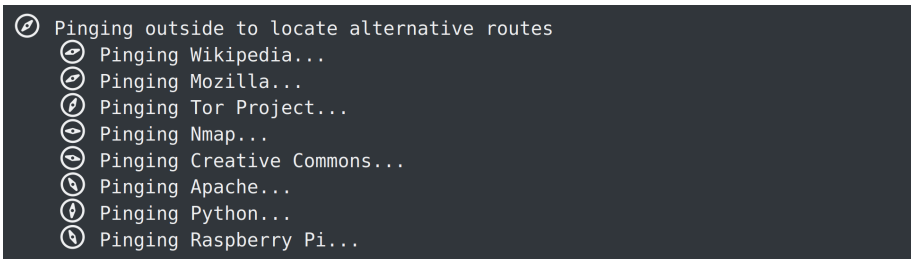
Tema con tipografía monospace grande



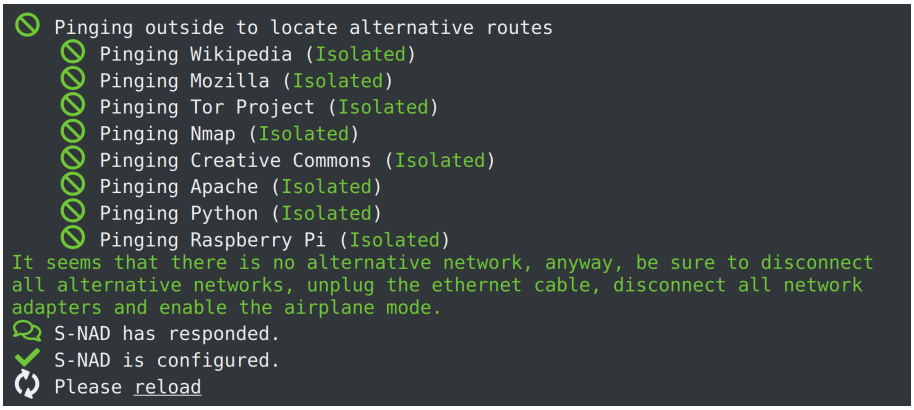
Comprobación del aislamiento de la red

La primera vez que se accede el *frontend* en cada arranque del servicio «http-server» se muestra una página especial que sirve para comprobar el aislamiento de la red y recordar a la persona usuaria que debe estar aislada de esta.

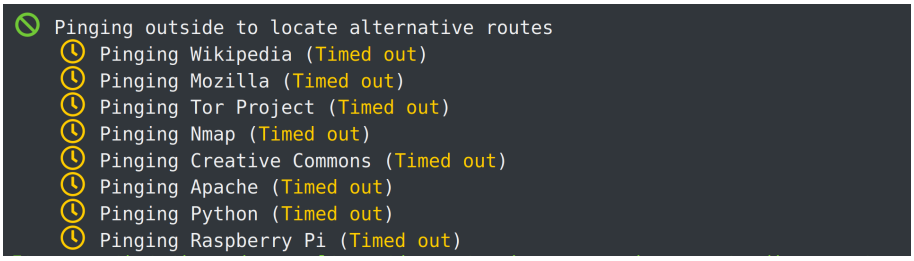
En esta primera pantalla se indican los pasos que se van llevando a cabo. En primer lugar se realizan llamadas a ocho servidores externos para comprobar si se recibe respuesta. Puesto que el servidor S-NAD aún no ha comenzado el enrutamiento, si no hay más redes activas no debería poder acceder a los servidores, obteniendo errores en todos ellos y confirmando –no de forma absoluta– que el *host* está aislado de otras redes.



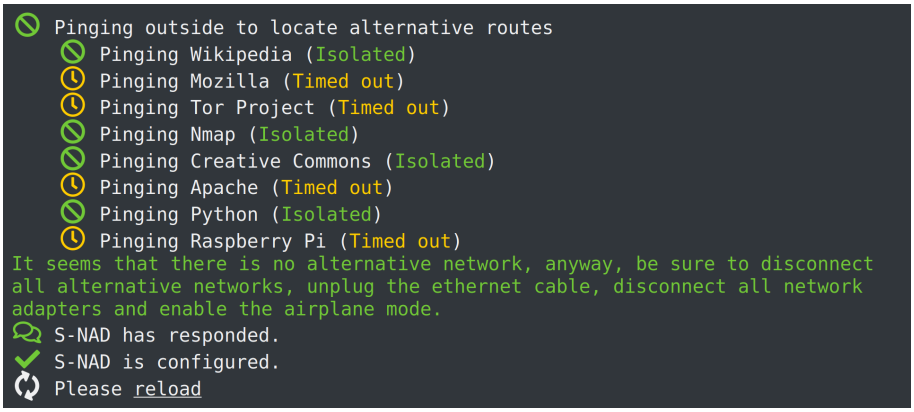
Si en efecto no existen otras redes, según la configuración pueden obtenerse errores instantáneos indicando entonces que el *host* parece estar aislado y recuerda a la persona usuaria que debe desconectar todas las redes alternativas, desconectar el cable ethernet, desconectar todos los adaptadores de red y conectar el modo avión.



Si no se obtiene ningún error inmediatamente, existe un *timeout* –Tiempo límite– de ocho segundos a partir de los cuales se supone que no hay red.

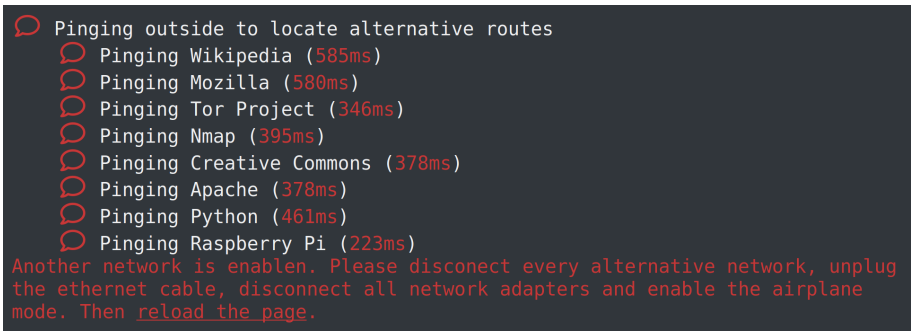


Después de superar el tiempo límite aún se espera a los errores, por si la persona usuaria prefiere esperar a obtener errores explícitos.



Una vez terminado este proceso, la persona usuaria debe recargar la página o presionar «reload».

En caso de que exista conexión a Internet por otra vía se muestra un mensaje y no se inician los servicios de enrutamiento.



Para realizar las peticiones a los servidores externos se utiliza una biblioteca de Alfred Gutierrez[g] modificada para adaptarse a las necesidades del proyecto. El código modificado puede encontrarse en los anexos.

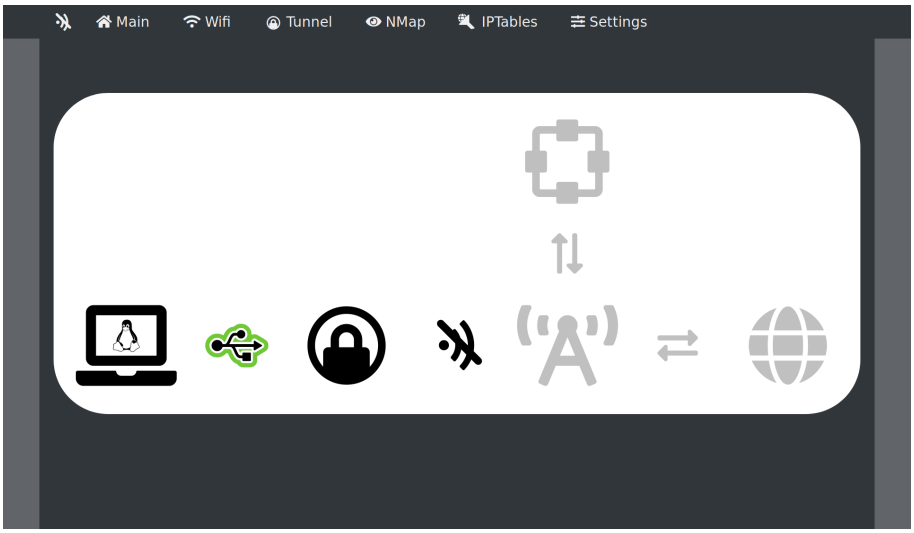
g https://github.com/alfg/ping.js

Página principal

La página principal muestra un diagrama con una visión general de la red, mostrando de izquierda a derecha por proximidad los dispositivos estimados de la red. Cada uno de estos elementos tiene un panel informativo asociado en el que se detalla el estado del mismo.

Diagrama

Según el estado de la conexión puede mostrar distinto aspecto. Cuando no existe una conexión Wifi el aspecto es similar a este:



Icono del *host*

El *host* se representa con distintos aspectos según el *user agent*. Estos son algunos ejemplos

 Computadora personal GNU/Linux

 Computadora personal Mac OS X

 Comandos como Wget o Curl

 Computadora personal Windows

 Teléfono inteligente Apple iPhone

 Teléfono inteligente Android

Para interpretar el user agent se utiliza una biblioteca de Faisal Salman. Esta genera, a partir del user agent, un objeto JS a partir del cual se escoge la combinación de iconos que representarán al dispositivo *host*. Este código se puede encontrar en los anexos.

Icono de la conexión USB

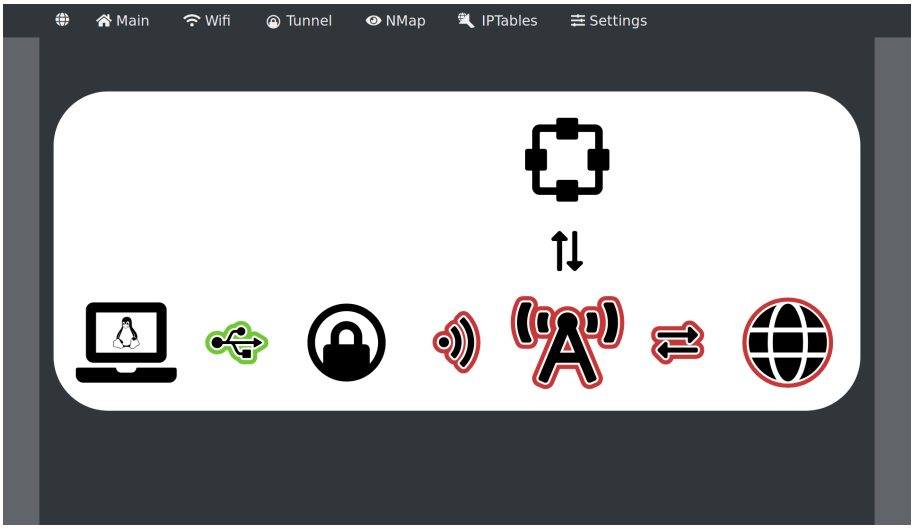
La conexión USB –  – siempre se muestra en verde y etiquetada como «Physically isolated (USB)» ya que es considerada segura.

Iconos de la ruta al servidor destino

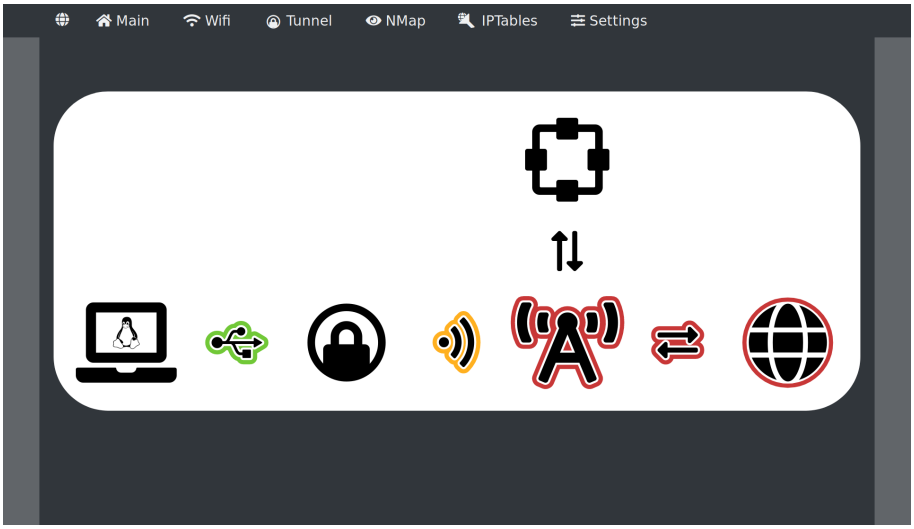
El resto de elementos hasta los posibles servidores destino son: la conexión Wifi ––, el punto de acceso –– y el propio servidor destino ––.

Si se ha establecido una conexión Wifi, el color y etiqueta contextual del icono depende de la seguridad de esta. El resto de elementos se destacan en rojo para señalar que todo mensaje en claro podrá ser leído por los dispositivos de enrutamiento. El servidor destino se resalta en rojo cuando puede obtener la IP y otros datos como la geolocalización o el *user agent* del *host*.

Cuando no se utiliza cifrado –red abierta– se etiqueta como «Unencrypted» y si se utiliza WEP, como «Weakly encrypted (WEP-104)» –el método de cifrado que se indica entre paréntesis puede variar según el protocolo, se obtiene dinámicamente en el S-NAD por medio de «wpa_suplicant»–. En ambos casos se representa en rojo.



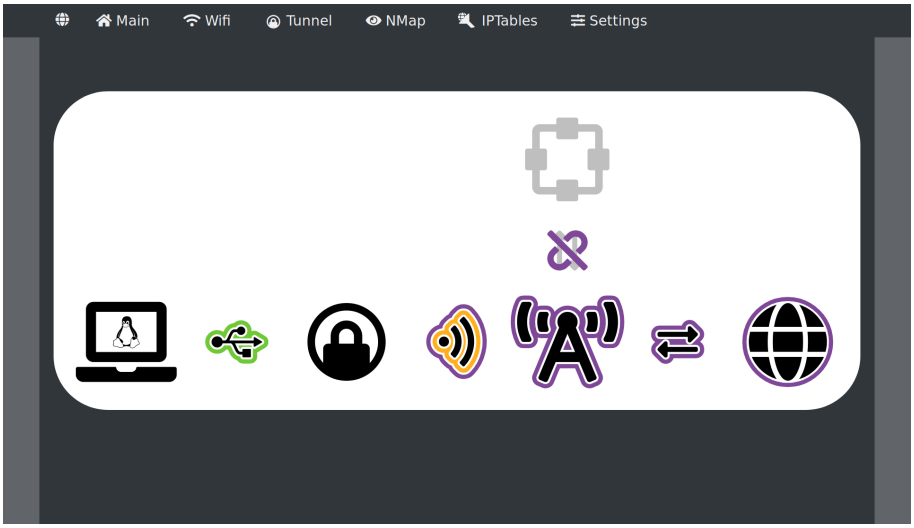
Cuando se utiliza WPA2 se etiqueta como «Encrypted (CCMP)» –el método de cifrado que se indica entre paréntesis puede variar según el protocolo, se obtiene dinámicamente en el S-NAD por medio de «wpa_suplicant»– y se muestra así:



En un futuro, cuando se implemente una solución a WPA2, tal vez se actualice la vista representando el Wifi en verde. Por el momento, se representa en naranja para indicar que no es garantía de seguridad.

Representación del tunelado

En esta vista se representa también, si está configurado, el tunelado. En el caso de utilizar, por ejemplo, el *proxy* «Tor», el aspecto sería este:



En este caso se incluye en la etiqueta de la conexión Wifi, a continuación del texto que le correspondiese a la seguridad de la misma, «[…] but the inner traffic is tunneled by Tor» –en caso de utilizar por ejemplo WPA2 la etiqueta sería esta: «Encrypted (CCMP) but the inner traffic is tunneled by Tor».

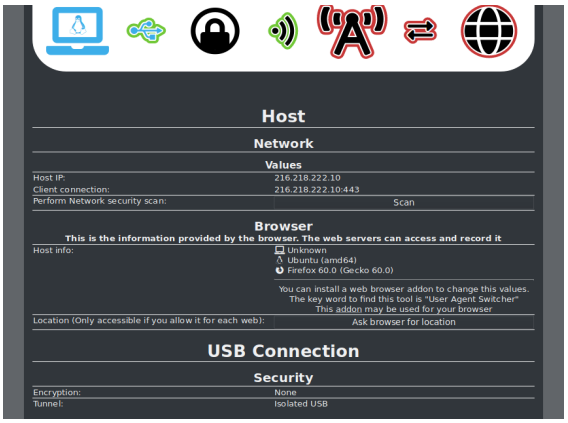
El punto de acceso muestra la etiqueta «Traffic is tunneled by Tor and inaccessible for the access point and the rest of routig elements».

Paneles informativos

Los paneles informativos se hacen visibles al hacer clic sobre cualquiera de los elementos del diagrama. Muchos de ellos engloban a la vez dispositivos y conexiones, por ejemplo, el panel del *host* muestra también la información de la conexión USB.

Panel informativo del *host* y conexión USB

En este panel se abarca tanto el *host* como la conexión USB.



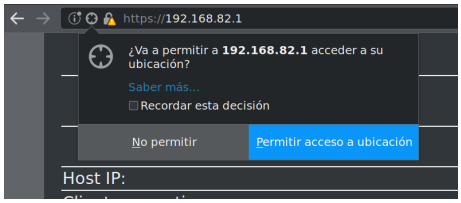
Se proporciona la IP y se ofrece la posibilidad de hacer un análisis mediante «NMap», que genera un informe del aspecto del *host* desde una red LAN a la que se conecte directamente. Cualquier vulnerabilidad descubierta aquí no afectaría a la seguridad, ya que S-NAD filtrará todas las comunicaciones, no obstante, es recomendable solventarlas, ya que cuando se acceda directamente a una red sin utilizar S-NAD serán efectivas.

También se muestra la información del equipo, sistema y navegador facilitada por este último y que no tiene por qué ser real, ya que la propia persona usuaria

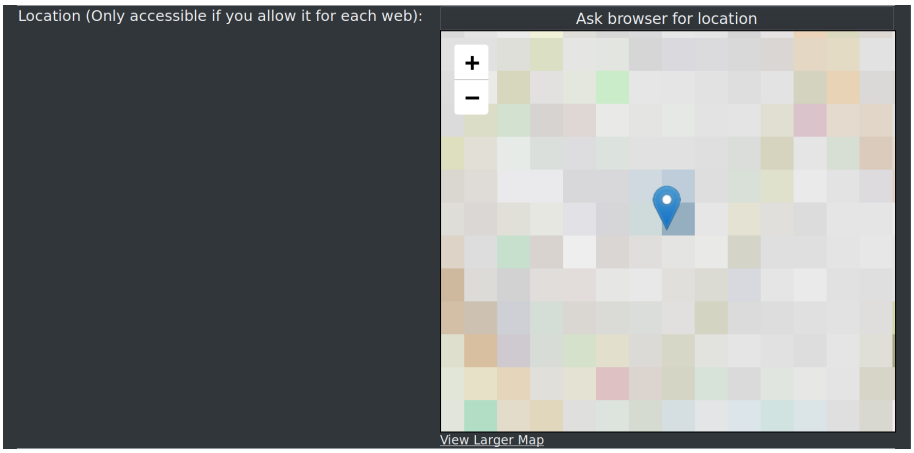
puede falsear su *user agent* mediante distintas técnicas. De hecho, se recomienda alterarlo para garantizar el anonimato, en esta vista se incluye un enlace a un addon que permite hacerlo.

Por último, permite solicitar al navegador la localización para ubicarte en un mapa –se utiliza el servicio «Open Street Maps[h]» una alternativa libre y respetuosa con los derechos de los usuarios a «Google maps»–. Esta ubicación no se obtiene mediante la IP, si no que la aporta cada navegador utilizando distintos servicios de localización y es mucho más precisa. Esta información se solicita desde las páginas web mediante el método nativo JS `navigator.geolocation.getCurrentPosition` y no es accesible por los sitios web ni por terceros –al menos en el caso de «Mozilla Firefox»– a menos que la persona usuaria conceda el permiso explícito. Además, consta de medidas de seguridad para garantizar que la información de la geolocalización no viaja sin cifrar. En navegadores privativos no se puede tener la certeza de que esto suceda así.

Algunos navegadores pueden no soportarlo, en este caso no se mostrará la geolocalización y la persona usuaria no debe preocuparse por ella, ya que no será accesible por este método para ninguno de los servidores a los que se acceda.



Si se permite el acceso se muestra el mapa con la ubicación:



Para generar el mapa en el punto adecuado se utiliza el método JS `get_osm()` desarrollado para este proyecto y que genera un elemento iframe a un mapa de «Open Street Maps» con la ubicación y el aumento que se pasan como parámetros. Puede encontrarse este método en los anexos.

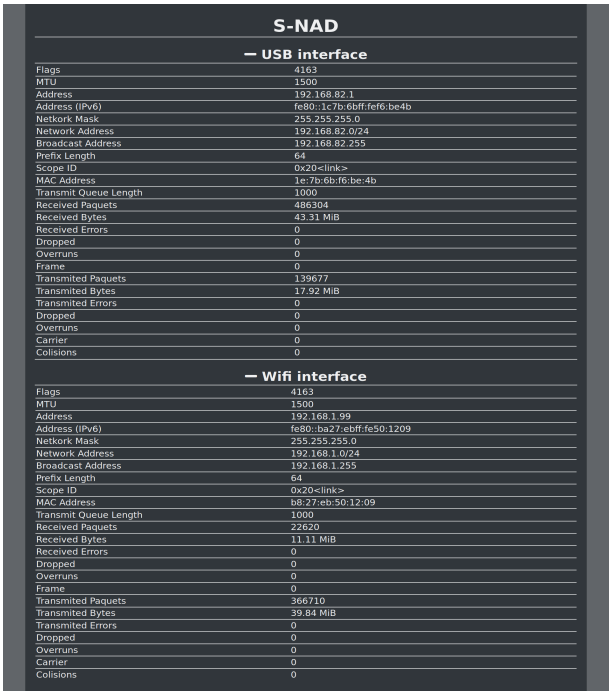
A su vez, la ubicación se solicita y actualiza con el método `update_browser_loc()` que se invoca mediante el botón «Ask browser for location».

La información de la conexión USB es estática ya que no puede variar. A menos que la persona usuaria configure un tunelado propio –no recomendable– no existe ninguna capa de cifrado y la seguridad reside en las características físicas de la conexión: la limitación maestro-esclavo explicada en la sección 4.1.1. Elementos hardware.

h https://www.openstreetmap.org

Panel informativo de S-NAD

Puesto que el servicio se ejecuta en S-NAD, es el dispositivo del que más información se puede obtener. El informe muestra un resumen tanto de la información que proporciona «ifconfig» como «route».



S-NAD

— USB interface

Flags	4163
MTU	1500
Address	192.168.82.1
Address (IPv6)	fe80::1c7b:6bff:fef6:be4b
Netkork Mask	255.255.255.0
Network Address	192.168.82.0/24
Broadcast Address	192.168.82.255
Prefix Length	64
Scope ID	0x20<link>
MAC Address	1e:7b:6b:f6:be:4b
Transmit Queue Length	1000
Received Paquets	486304
Received Bytes	43.31 MiB
Received Errors	0
Dropped	0
Overruns	0
Frame	0
Transmited Paquets	139677
Transmited Bytes	17.92 MiB
Transmited Errors	0
Dropped	0
Overruns	0
Carrier	0
Colisions	0

— Wifi interface

Flags	4163
MTU	1500
Address	192.168.1.99
Address (IPv6)	fe80::ba27:ebff:fe50:1209
Netkork Mask	255.255.255.0
Network Address	192.168.1.0/24
Broadcast Address	192.168.1.255
Prefix Length	64
Scope ID	0x20<link>
MAC Address	b8:27:eb:50:12:09
Transmit Queue Length	1000
Received Paquets	22620
Received Bytes	11.11 MiB
Received Errors	0
Dropped	0
Overruns	0
Frame	0
Transmited Paquets	366710
Transmited Bytes	39.84 MiB
Transmited Errors	0
Dropped	0
Overruns	0
Carrier	0
Colisions	0

Panel informativo de la conexión Wifi

En el informe de la conexión Wifi se muestra toda la información que proporciona «wpa_suplicant» y se ofrece además la posibilidad de realizar un escaneo Nmap. Este escaneo se realiza desde la red Wifi, obviamente, por lo que, como quedó explicado en el apartado 5.3.3.2. Cortafuegos y enrutamiento –Netfilter/IPTables– y si la persona usuaria no ha alterado las tablas de «Netfilter», el informe debería indicar que no se ha localizado el *host*.

Wifi

Wifi network

WPA state:	[redacted]
SSID:	[redacted]
Frequency:	[redacted]
Mode:	[redacted]
BSSID:	[redacted]
UUID:	[redacted]
P2P device address:	[redacted]

S-NAD

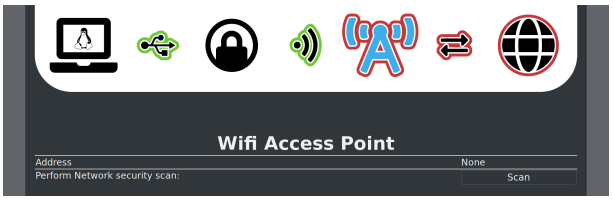
MAC address:	[redacted]
IP address:	[redacted]

Security

Pairwise cipher:	CCMP
Group cipher:	TKIP
Key management:	WPA2-PSK
Perform Network security scan:	Scan

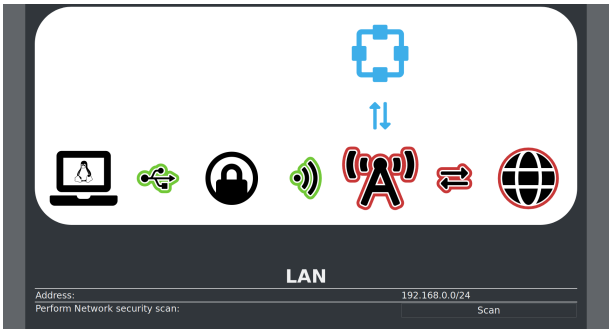
Panel informativo del punto de acceso

No existe mucha información que aportar sobre el punto de acceso inalámbrico. Puede consultarse su dirección y realizar un escaneo «NMap» contra él.



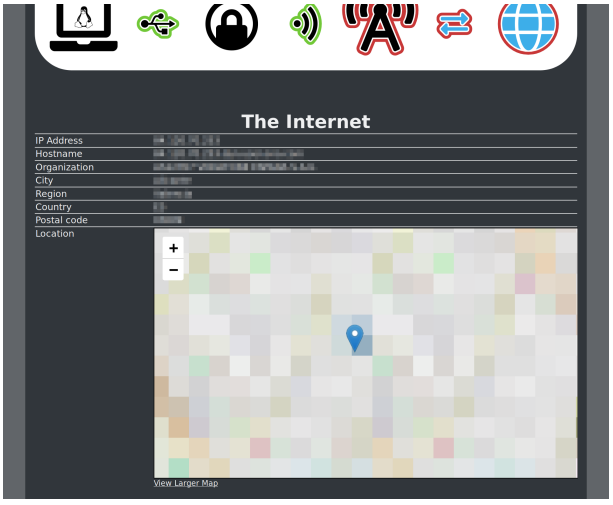
Panel informativo de la red LAN

Al igual que la información disponible en el panel del punto de acceso Wifi, aquí se muestra la dirección de la red LAN en que se encuentra S-NAD y se ofrece la opción de realizar un escaneo de la misma.



Panel informativo del servidor destino

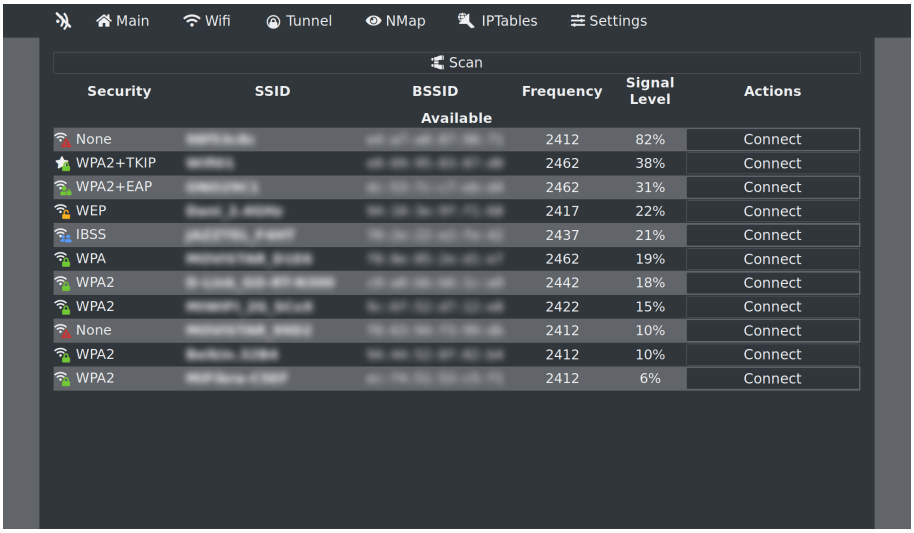
Consultando a un servicio — externo mediante su API se obtiene la información visible para los servidores a los que se accedan, incluida la geolocalización, cuya precisión depende del proveedor de servicios de Internet –a partir de ahora ISP del inglés *Internet Service Provider*– de la persona usuaria.



Página de la configuración de la conexión Wifi

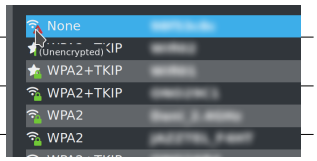
Esta página es sencilla e imita las interfaces gráficas de los sistemas operativos para gestionar la conexión Wifi.

Al cargar realiza una primera petición de escaneo y muestra el resultado en una tabla.



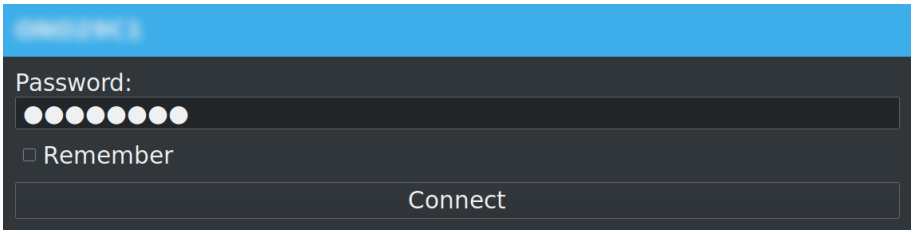
Los iconos que se muestran a la izquierda del campo «security» muestran un aspecto y una etiqueta contextual distinta según las propiedades de la red:

Icono de la red			
Red Wifi no recordada			
Red Wifi recordada			
Icono de seguridad			
	Descripción	Etiqueta contextual	Texto del campo
	Ninguna	(Unencrypted)	None
	WEP	(Weakly encrypted)	WEP
	WPA	(Encrypted)	WPA
	WPA+TKIP		WPA+TKIP
	WPA2		WPA2
	WPA2+TKIP		WPA2+TKIP
	WPA2+EAP	(Encrypted + EAP auth)	WPA2+EAP
	IBSS	(Personal P2P infrastructure)	IBSS



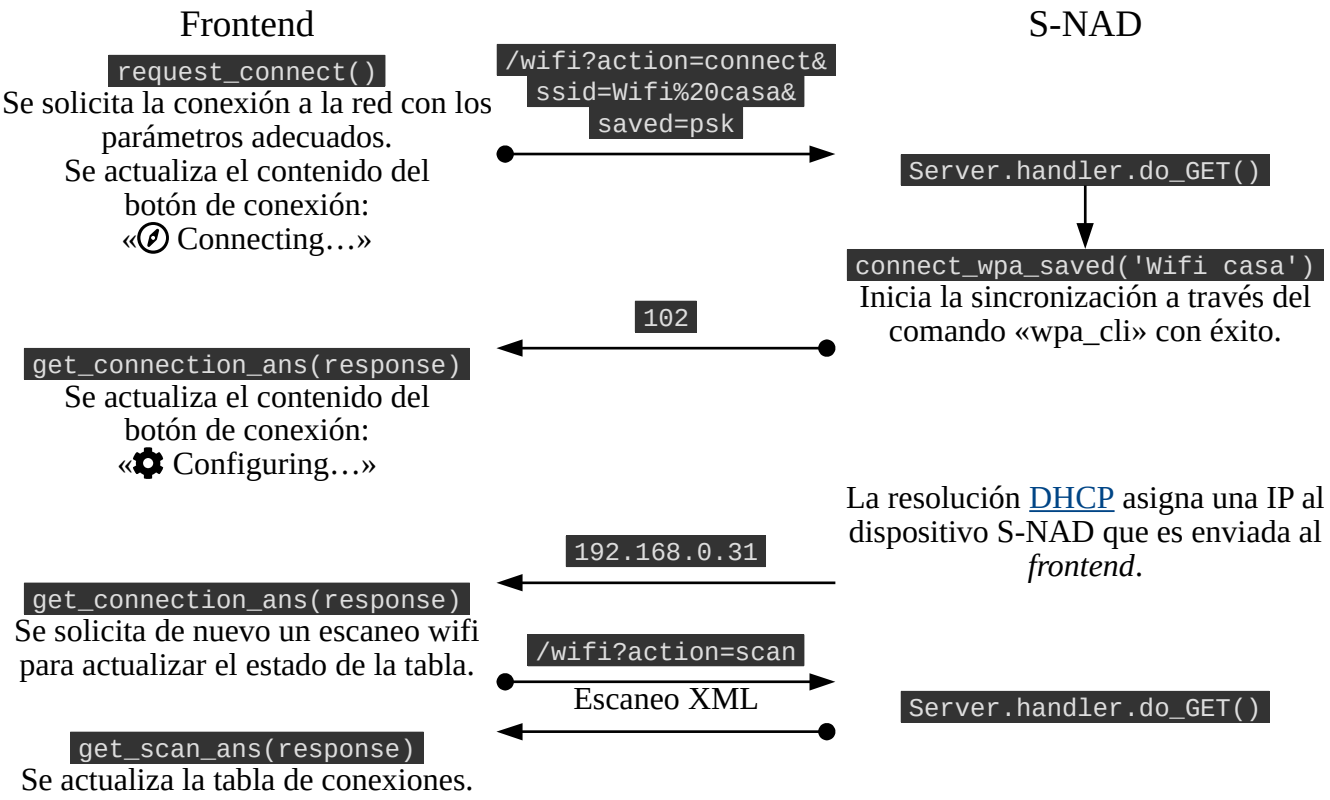
Aunque cada icono se represente en un color distinto, el color no aporta ninguna información añadida, ya que de otra manera podría no ser accesible a personas daltónicas.

Cuando se selecciona una red protegida con contraseña se muestra el siguiente diálogo:

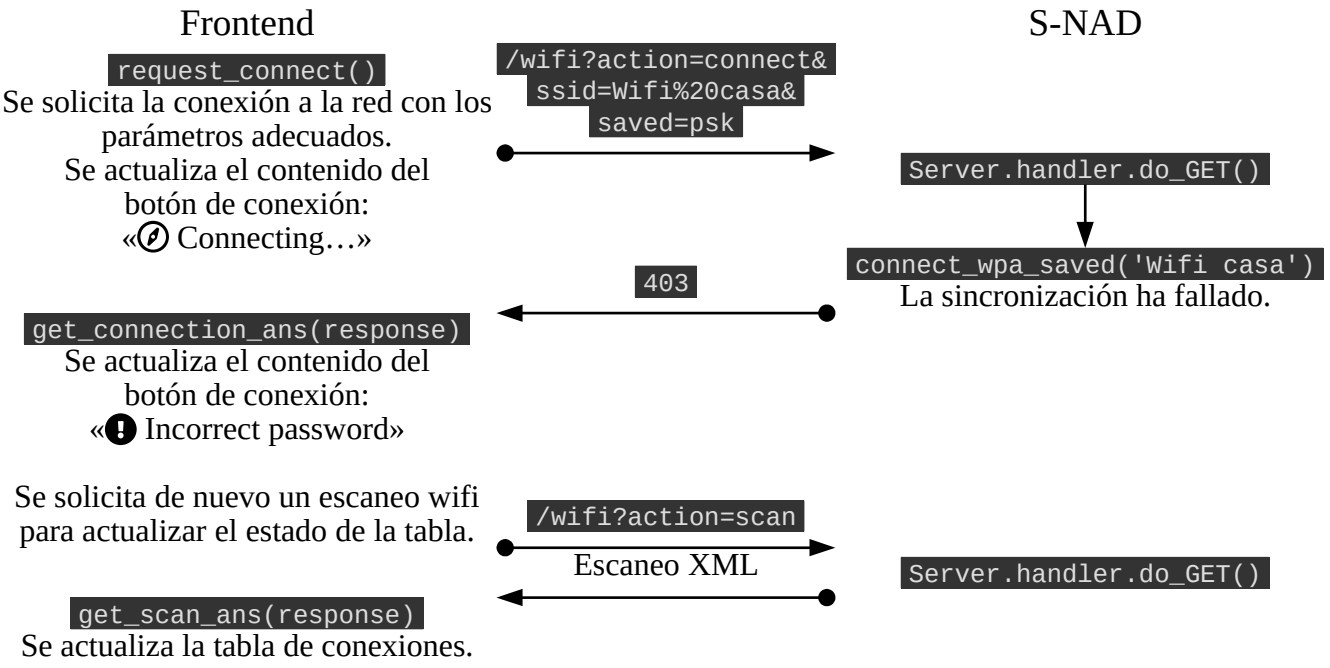


En él, la persona usuaria introduce la contraseña y puede escoger si S-NAD la recordará, por defecto esta opción no está marcada. El botón «connect» está deshabilitado hasta que la clave tiene una longitud válida –mayor o igual a 8 para WPA o 5, 13 o 16 para WEP–.

Una vez se solicita la conexión, comienza una comunicación AJAX entre S-NAD y el *frontend*. Los mensajes y métodos que se envían dependen de la seguridad de la red y de si la clave está recordada. Para el ejemplo se supondrá una red Wifi WPA cuya clave está recordada.



En caso de que la contraseña haya cambiado desde que se guardó o que se haya facilitado una incorrecta, las comunicaciones se sucederán de la siguiente forma:

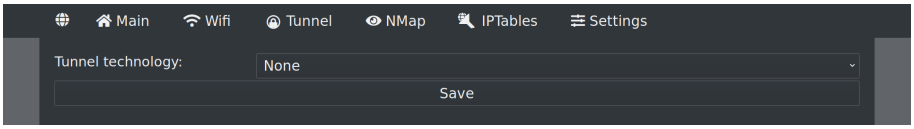


Página de configuración del tunelado

La página de tunelado es muy sencilla, simplemente ofrece la capacidad de decidir la combinación de configuraciones para el túnel descritas en la sección 5.2.3.4. Configuración del tunelado. El formulario añade y elimina campos según se seleccionan.

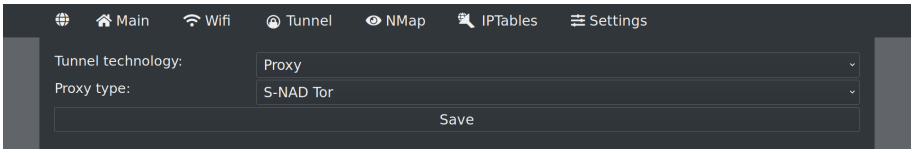
Sin tunelado

El botón «Save», situado en la parte inferior, informaría a S-NAD de la nueva configuración mediante una petición AJAX a la URL `/tunnel?action=tunnel&technology=none`.



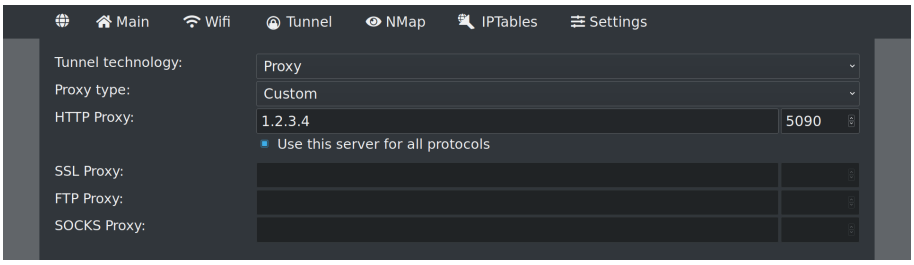
Proxy «Tor» en S-NAD

Para el *proxy* «Tor» no se requiere configuración manual. El botón «Save» envía una petición AJAX contra la dirección `/tunnel?action=tunnel&technology=proxy&type=tor`.



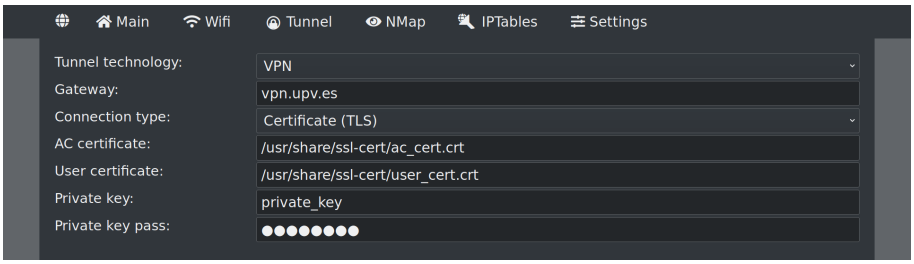
Proxy personalizado

El *proxy* personalizado requiere más configuración. El *frontend* se encarga de reproducir el servidor cuando se selecciona «Use this server for all protocols», de forma que la petición incluye el servidor especificado como «HTTP Proxy» en el resto. La petición del botón «Save» se efectúa contra la dirección `/tunnel?action=tunnel&technology=proxy&type=custom&http=1.2.3.4:5090&ssl=1.2.3.4:5090&ftp=1.2.3.4:5090&socks=1.2.3.4:5090`.

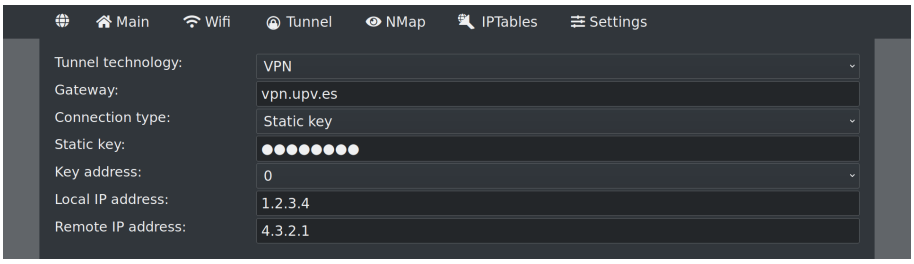


VPN Certificado –TLS–

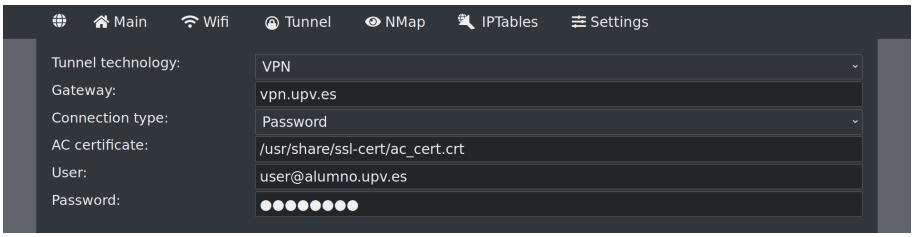
Algunas configuraciones VPN requieren certificados, pueden enviarse propios desde el *frontend* o utilizar los disponibles en el sistema.



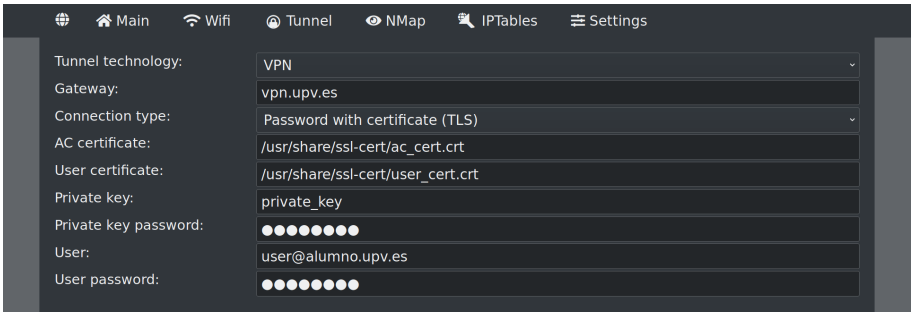
VPN Clave estática



VPN Contraseña



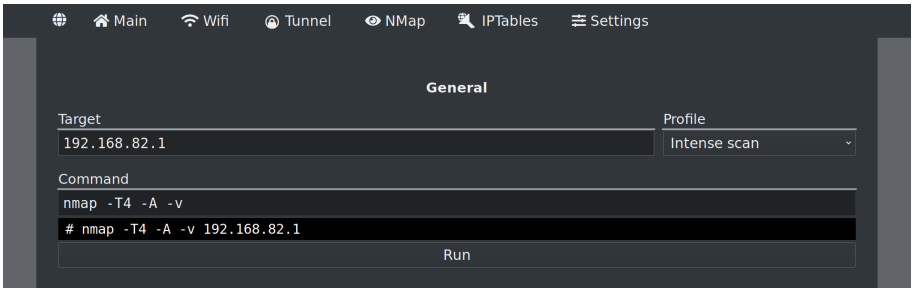
VPN Contraseña con certificado –TLS–



Página para realizar escaneos de red de «NMap»

La página «NMap» permite a la persona usuaria disponer del comando «NMap» de S-NAD desde el *frontend*.

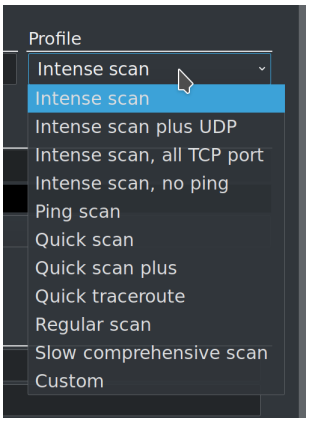
El campo «Target» permite introducir cualquier cantidad de objetivos para el análisis separados por espacios y en el «Command» se completan el comando y las opciones. Bajo el campo «Command» se muestra la composición del comando con los datos aportados y que se enviará a S-NAD cuando se presione el botón «Run».



Existen tres formas de componer el comando «NMap» que se enviará a S-NAD mediante esta interfaz.

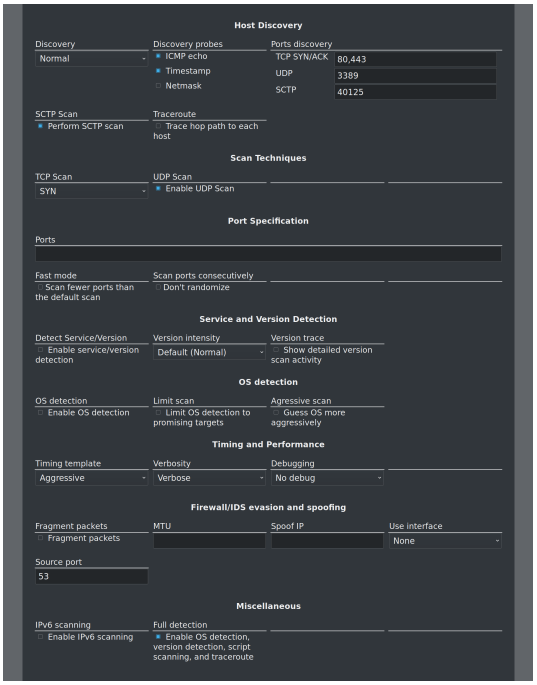
La más sencilla es utilizar alguno de los modos predefinidos. Cuando se selecciona uno de estos perfiles en el desplegable «Profile» se completa el campo «Command» con las opciones pertinentes.

Intense scan	`nmap -T4 -A -v`
Intense scan plus UDP	`nmap -sS -sU -T4 -A -v`
Intense scan, all TCP	`nmap -p 1-65535 -T4 -A -v`
Intense scan, no ping	`nmap -T4 -A -v -Pn`
Ping scan	`nmap -sn`
Quick scan	`nmap -T4 -F`
Quick scan plus	`nmap -sV -T4 -O -F --version-light`
Quick traceroute	`nmap -sn --traceroute`
Regular scan	`nmap`
Slow comprehensive scan	`nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 --script "default or (discovery and safe)"`
Custom	



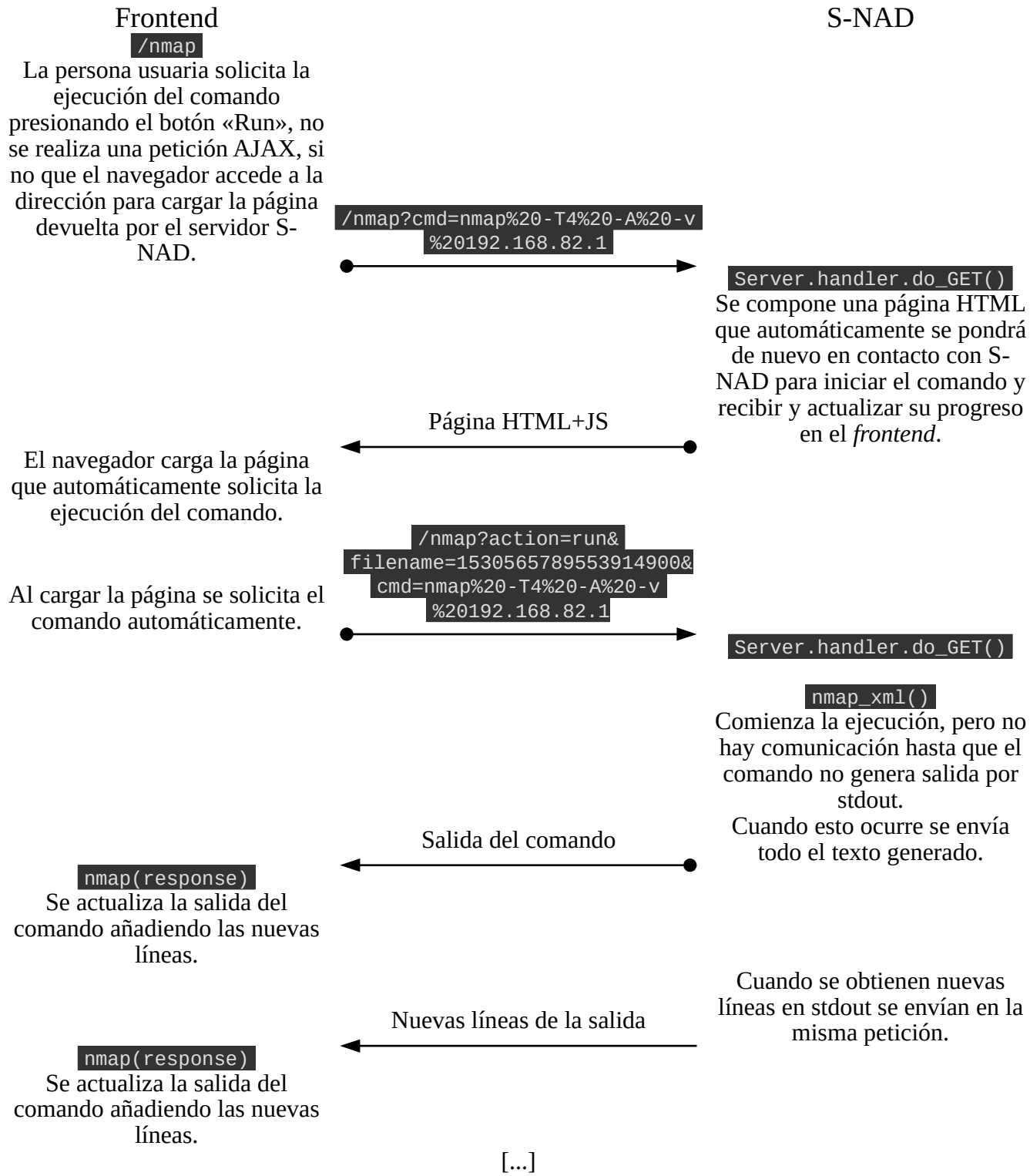
La segunda opción por orden de sencillez es la elección individual de opciones mediante los elementos de formulario que se encuentran por debajo del botón de envío:

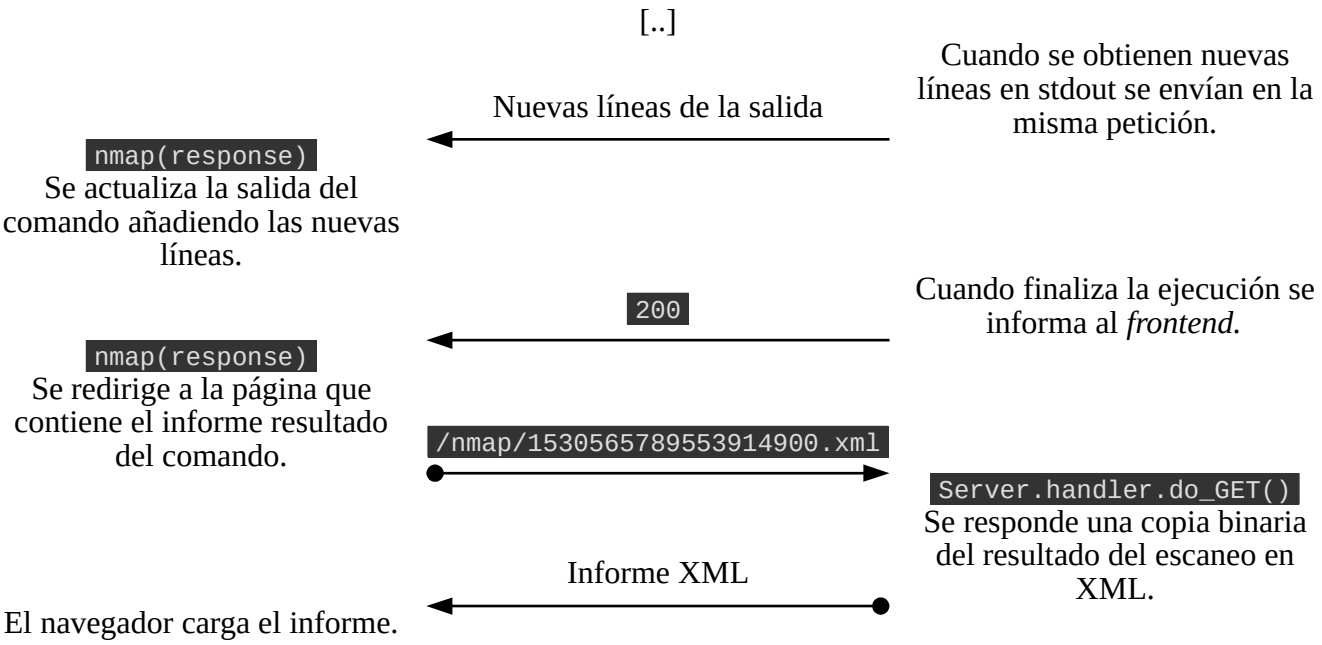
De esta manera, conforme se modifican los distintos campos, se compone el campo «Command» de forma automática. No se muestran todas las opciones del comando, ya que existen demasiadas como para abarcarlas en una sola interfaz. Puede consultarse el manual de «NMpap» para más detalle.



Por último, siempre existe la posibilidad, si se desea utilizar una opción no incluida, de emplear la tercera opción para componer el comando, que consiste simplemente en completar las opciones manualmente editando el campo «Command». Existen una serie de filtros para evitar la inyección de código en el campo, tales como prohibir el carácter «;», evitar el uso de variables de entorno –$*cadena*– o evitar llamadas a otros comandos –$(*comando*) o `*comando*`–.

La comunicación que se lleva a cabo a la hora de solicitar el escaneo ha sido tratada superficialmente en la sección 5.2.2.5. Ejecutar «NMap».




[..]
Cuando se obtienen nuevas líneas en stdout se envían en la misma petición.
Nuevas líneas de la salida
nmap(response)
Se actualiza la salida del comando añadiendo las nuevas líneas.
Cuando finaliza la ejecución se informa al *frontend.*
200
nmap(response)
Se redirige a la página que contiene el informe resultado del comando.
/nmap/1530565789553914900.xml
Server.handler.do_GET()
Se responde una copia binaria del resultado del escaneo en XML.
Informe XML
El navegador carga el informe.

6. Implantación

La implantación del proyecto utiliza el mismo sistema que utiliza la distribución «Raspbian». Consiste en una imagen de disco con la instalación del software y la configuración pertinente.

El hardware necesario para reproducir este proyecto es una placa «Raspberry Pi Zero W», un cable *USB Standart Type A* macho a *USB Micro B* y una tarjeta «MicroSD» de un mínimo de 1.7 GiB –una tarjeta de mayor capacidad no mejorará el rendimiento–.

El proceso para implantarlo es muy sencillo. En primer lugar se obtiene un archivo con extensión «img» que es una imagen binaria del sistema. Contiene dos particiones, una de arranque –*boot*– y otra con el sistema.

El siguiente paso es grabar la imagen en una tarjeta «MicroSD». En sistemas GNU/Linux se puede usar el comando nativo `dd` para grabarla. En otros sistemas existen también soluciones sencillas.

Finalmente, sólo queda introducir la tarjeta en la placa «Raspberry Pi Zero W». En la primera ejecución ampliará el tamaño de la partición del sistema –que por defecto ocupa el mínimo necesario– hasta que cubra todo el espacio disponible.

7. Pruebas

Se realizan distintas pruebas de rendimiento para determinar el impacto en las prestaciones que supone el uso de S-NAD.

7.1. Impacto sobre la conexión

Para ello se han realizado pruebas contra el servidor de la Universidad de Valencia –que es el que mejor latencia ofrecía desde la red en que se realizaron las pruebas–. La información que se recopila es el tiempo de Ping –tiempo en milisegundos que tarda el equipo en recibir una respuesta del servidor– y el ancho de banda tanto de bajada como de subida.

Para cada configuración de túnel soportado se han realizado varias pruebas en el mismo equipo, tanto conectado directamente a la red como a través del dispositivo S-NAD.

Las pruebas de VPN se han realizado a través la red VPN de la Universidad Politécnica de Valencia. En la prueba en el equipo se ha configurado la red VPN en el sistema operativo.

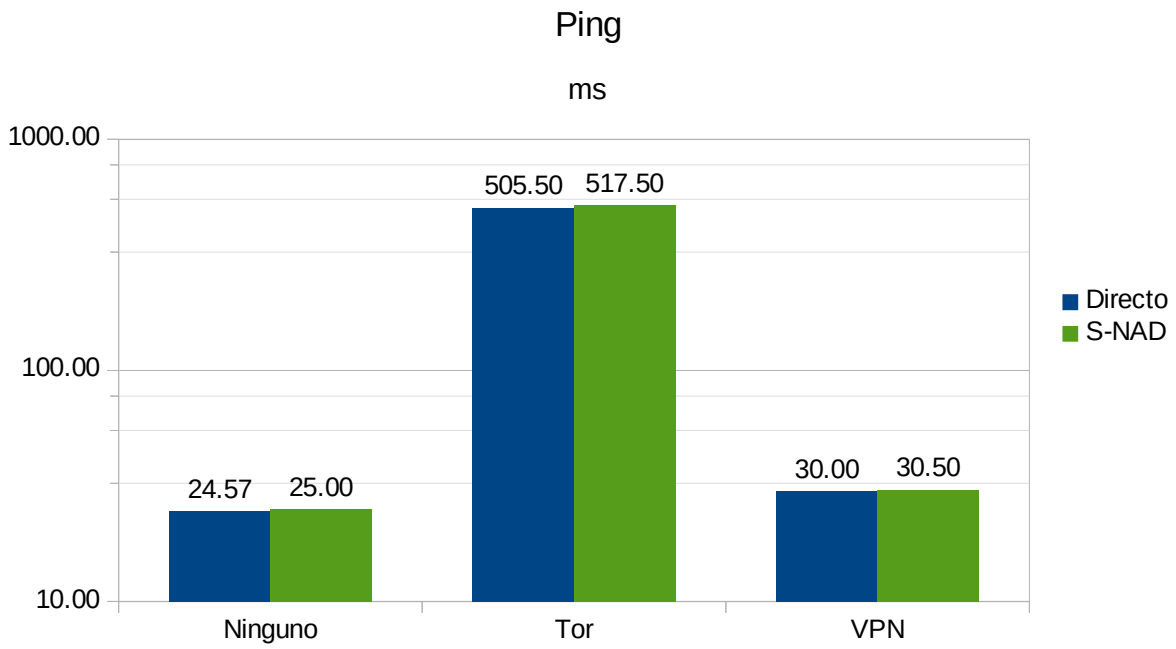
Las pruebas de *proxies* se han realizado sobre «Tor». Para asegurar un escenario lo más parecido posible entre el modo directo y a través del dispositivo S-NAD, se ha configurado en el *host* la misma ruta de «Tor» que la escogida por S-NAD. Estas pruebas se han realizado contra un servidor cercano al nodo de salida con el fin de reducir el impacto de la red de *proxies* sobre las pruebas.

Los resultados obtenidos son los siguientes:

Túnel	Enrutamiento	Ping	Descarga	Subida
Ninguno	Directo	24.57 ms	40.48 Mbps	15.71 Mbps
	S-NAD	25.00 ms	13.30 Mbps	10.14 Mbps
VPN UPV	Directo	30.00 ms	15.36 Mbps	7.60 Mbps
	S-NAD	30.50 ms	13.10 Mbps	7.98 Mbps
«Tor»	Directo	505.50 ms	1.67 Mbps	0.71 Mbps
	S-NAD	517.50 ms	1.63 Mbps	0.67 Mbps

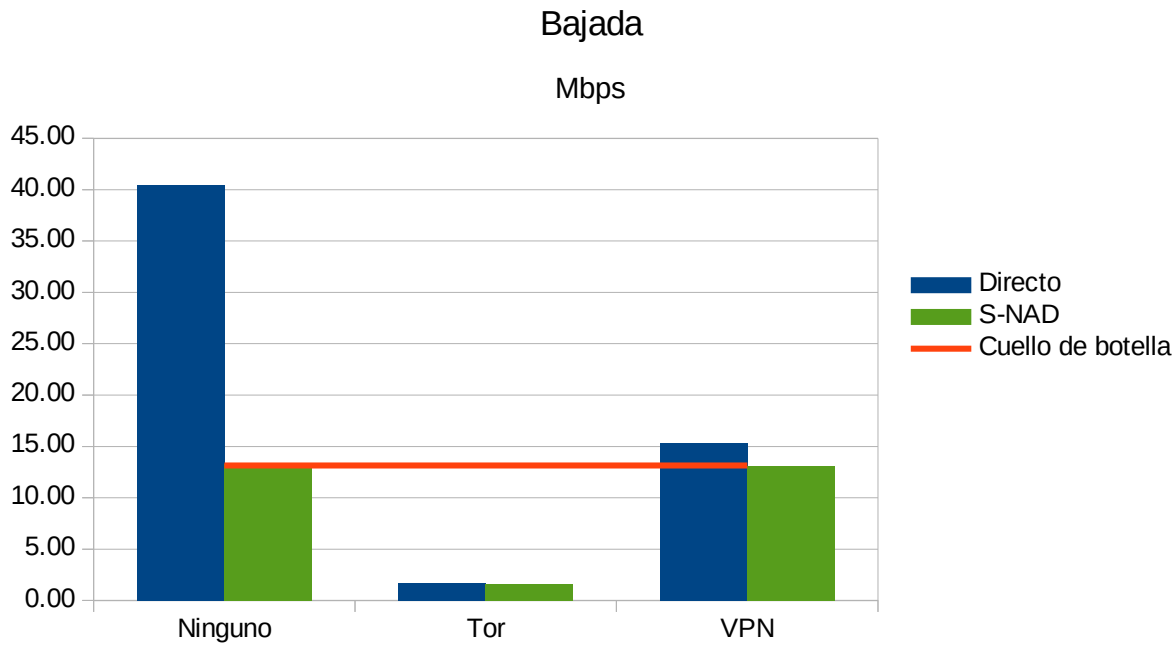
Repercusión en la latencia

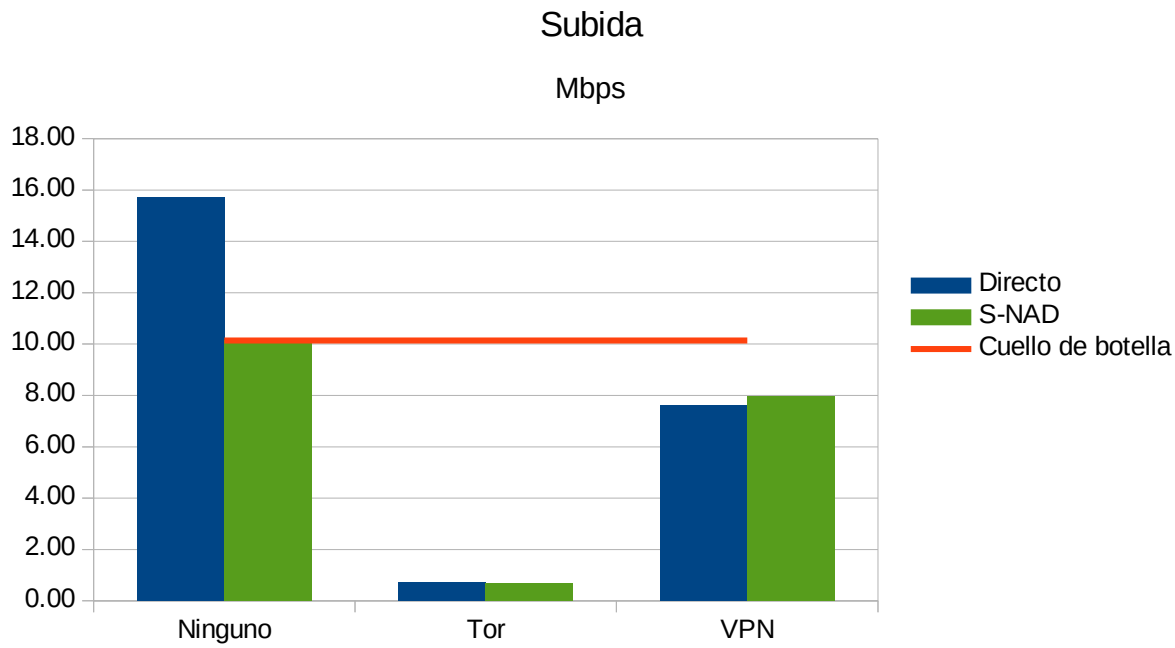
La repercusión sobre el Ping es mínima. Esto es muy positivo ya que implica que el tiempo que tarda la respuesta del servidor en llegar al *host* es prácticamente igual al que se obtendría mediante una conexión directa. Se garantiza de esta forma una navegación tan fluida como la que se experimentaría sin utilizar S-NAD.



Repercusión en el ancho de banda

En el ancho de banda se puede apreciar una repercusión clara del uso del dispositivo S-NAD. Este provoca un cuello de botella que trunca la bajada a unos 13.15 Mbps y la subida a unos 10.14 Mbps.





La conclusión a la que podemos llegar, observando estos resultados, es que en la navegación web no se apreciaría ningún impacto. La gran limitación es el ancho de banda, pero esto no tiene por qué afectar necesariamente a la experiencia de la persona usuaria, sólo percibiría esta limitación cuando requiriese hacer un gran uso de ancho de banda, al subir o bajar archivos de gran tamaño o al emitir o recibir vídeo a tiempo real –*streaming*– de muy alta calidad –para el visionado de vídeos o el establecimiento de video-llamadas no debería percibirse–. Por lo general existen otros cuellos de botella que repercuten más gravemente en la conexión, los propios servidores suelen limitar los anchos de banda que utilizan los clientes. Una conexión ADSL de 13.15 Mbps de bajada y 10.14 Mbps de subida debería ser suficiente para un único *host*.

7.2. Consumo energético

Se emplea un multímetro USB –un dispositivo físico que se conecta entre el puerto USB del *host* y el periférico e indica el voltaje y amperaje– para medir el consumo del dispositivo S-NAD. Las mediciones se realizan mientras se llevan a cabo las distintas pruebas de red con cada configuración.

Túnel	Voltaje		Amperaje		Consumo
	Rango	Media	Rango	Media	
Ninguno	4.93 V - 4.95 V	4.94 V	102 mA-221 mA	152 mA	750.88 mW
VPN UPV	4.93 V - 4.95 V	4.94 V	99 mA-232 mA	159 mA	785.46 mW
«Tor»	4.93 V - 4.95 V	4.94 V	112 mA-251 mA	171 mA	844.74 mW

Estos resultados son muy positivos ya que revelan un consumo muy reducido. El resultado no varía significativamente entre las distintas configuraciones de tunelado. El consumo en reposo –S-NAD está conectado pero el *host* no emplea la red– es de unos 103 mA –508.82 mW–.

8. Conclusiones

El resultado final de este proyecto ha sido muy positivo y ha resultado en un dispositivo perfectamente funcional y fácilmente reproducible. Se han alcanzado todos los objetivos y se han explorado áreas de conocimientos más allá de las impartidas en el grado.

8.1. Cumplimiento de objetivos

Objetivos de seguridad

En lo que a seguridad respecta, se han alcanzado los objetivos. Se emplean tecnologías estándares, libres y seguras. Nunca se puede garantizar la seguridad, pero las herramientas utilizadas aportan la máxima garantía que se puede alcanzar en un proyecto de estas características.

Objetivos de acceso universal

Respecto a la universalización también se han obtenido buenos resultados. La compatibilidad del producto final es muy elevada. Según las estadísticas de uso de navegadores en equipos personales de entre el comienzo del año 2018 y junio del mismo año publicado por «statcounter[i]», el producto final sería compatible con el 99.94% de los *hosts* que navegan por Internet. El 0.06% restante corresponde a navegadores desconocidos para la empresa, por lo que no se puede saber si son compatibles.

Los objetivos relacionados con aspectos de accesibilidad han sido solventados satisfactoriamente respetando la configuración del sistema –tanto color como tamaño de fuente y tipografía– y soportando resoluciones de pantalla pequeñas –que es la mejor forma de facilitar el uso para personas con campo visual reducido–.

Objetivos de eficiencia

Se ha alcanzado una eficiencia muy elevada. La única repercusión en la red es el ancho de banda, que no tiene por qué ser perceptible en un uso personal de Internet.

El mejor resultado de eficiencia que se ha obtenido ha sido en el consumo eléctrico, que es realmente bajo.

8.2. Perspectiva crítica

Tras el comienzo de la implementación se han revaluado decisiones de implementación que no han resultado óptimas.

El decisión con mayor repercusión ha sido en lo referente a las comunicaciones entre el JS del *frontend* y el servidor web en S-NAD. Originalmente estas comunicaciones se llevaban a cabo mediante websockets, se abría una comunicación al cargar cada página y se utilizaba para todas las llamadas y respuestas. Gestionar un socket, con los riesgos que entraña, teniendo que garantizar que queda cerrado cuando el navegador libera la página y sorteando las limitaciones que los navegadores les imponen suponían una dificultad muy elevada y no aportaba ninguna ventaja sobre el uso de peticiones AJAX.

Otra decisión que varió tras el inicio del desarrollo fue el uso de HTTPS en lugar de HTTP. La conexión USB está protegida de ataques de intermediario por la limitación de su tecnología, por lo que no es necesario cifrar el tráfico, pero al comenzar con las pruebas con los distintos navegadores se percibieron limitaciones derivadas del uso de HTTP. La primera era que el navegador indica a la persona

i http://gs.statcounter.com

usuaria que la comunicación no es segura, ya que no puede saber el tipo de conexión que hay por debajo. Esto transmite una sensación de inseguridad a la persona usuaria. La otra limitación era que el navegador no permite el uso de ciertas funcionalidades –como la geolocalización– por no confiar en la página.

8.3. Relación del trabajo desarrollado con los estudios cursados

Se pueden establecer relaciones claras con determinadas asignaturas y más veladas con otras. Las relaciones más significativas son las que se pueden establecer entre las tecnologías de cifrado y la asignatura «Seguridad en los Sistemas Informáticos», en que se estudiaron conceptos imprescindibles para este trabajo, tales como los algoritmos de cifrado asimétrico, los túneles TLS o los certificados. Los conocimientos sobre la configuración de los servicios de enrutamiento tales como DNS, DHCP, Iptables, etc. corresponden a la asignatura de «Configuración, Administración y Gestión de Redes» cuyo contenido ha sido imprescindible para el desarrollo de este proyecto.

8.4. Ampliación de conocimientos

Para el desarrollo de este proyecto se han explorado áreas que no se han tratado en la carrera –al menos no en la especialidad de Ingeniería de computadores–, sobretodo en relación con el desarrollo web y el diseño de interfaces accesibles para personas con diversidad funcional.

9. Trabajos futuros

9.1. Ampliar la accesibilidad a personas ciegas

Para alcanzar un mayor grado de accesibilidad se podría implementar una solución amigable para personas ciegas. La solución actual no cumple esta funcionalidad ya que los navegadores de texto no procesan JS, tal y como está concebido el proyecto no es posible dar una solución para este tipo de navegadores. Una posible solución sería sustituir AJAX por enlaces dentro del cuerpo de la página, pero requeriría una nueva implementación.

9.2. Tecnologías VPN

para dar soporte más redes VPN pueden implementarse otras tecnologías de VPN.

9.3. Configuración «Tor»

Para aportar más versatilidad al servicio «Tor» podrían incluirse campos en el *frontend* que permitiesen a la persona usuaria configurar el comportamiento de este, permitiendo escoger la localización geográfica del nodo de salida, la ruta de los nodos, etc.

10. Referencias

[1]: M. Vanhoef (2017). Key Reinstallation Attacks: Breaking the WPA2 Protocol. London: Black Hat Briefings Europe.
[2]: M. Vanhoef & F. Piessens (2017). Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. Dallas, USA: Proceedings of the 24th ACM Conference on Computer and Communication Security.
[3]: Kevin Benton (2010). The Evolution of 802.11 Wireless Security.
[4]: IEEE (2004). 802.11i-2004.
[5]: Villalón Huerta A. (2002). Seguridad en Unix y redes.
[6]: Raymond E. (2001). The Cathedral & the Bazaar. O'Reilly Media, Inc.
[7]: Wreski D. (1998). Linux Security Administrator´s Guide.
[8]: Broadcom Corporation (2012). BCM2835 ARM Peripherals. Science Park Milton, Road Cambridge CB4 0WW.
[9]: Kolesnikov, O., & Hatch, B. (2003). Redes Privadas Virtuales Con Linux : Guía Avanzada.
[10]: Ranum M. J. (1995). Firewalls Frequently Asked Questions.
[11]: Purdy, G. (2004). Linux Iptables : Pocket Reference.
[12]: Chapman D. B. and Zwicky E. D. (1995). Building Internet Firewalls. : O´Reilly &Associates.
[13]: Insecure.Com LLC (2005). Manual NMap.

11. Glosario

HTTP	HTTP –*Hypertext Transfer Protocol*– es un protocolo de comunicación de Internet diseñado originalmente para transferir páginas web y que actualmente se utiliza para en el envío de todo tipo de archivos en la navegación web.
HTTPS	HTTPS –*Hypertext Transfer Protocol Secure*– no es más que el mismo protocolo HTTP sobre una capa de cifrado que impide que los ataques de intermediario obtengan la información.
En claro	Se utiliza esta expresión para referirse al tráfico o a los datos que no están cifrados y pueden ser interpretados por personas distintas al destinatario.
TLS	TLS –*Transport Layer Security*– es un protocolo de comunicación de Internet del nivel de transporte que aplica una capa de cifrado punto a punto con el destino a todos los protocolos por encima de él. Se utiliza con protocolos de capas superiores para aportarles seguridad. El ejemplo más conocido es HTTP, llamado HTTPS al combinarse con TLS.
Servidor *proxy*	Un *proxy* es un servidor intermediario entre la comunicación de una máquina clienta y otra servidora, la clienta envía una petición al *proxy*, que la realiza al segundo servidor. Este responde al *proxy* y hace llegar la respuesta a la máquina que realizó la petición en primera instancia.
Servicio Tor	Tor es un servicio orientado al anonimato que establece un camino de varios proxies con túneles cifrados entre ellos, de forma que ningún nodo independiente puede acceder a los datos ni conoce al resto.
VPN	VPN –*Virtual Private Network*– es una tecnología de red que simula una red se área local segura sobre una infraestructura distinta, en el caso de este proyecto, desde Internet.
NAT	NAT –*Network Address Translation*– es un procedimiento mediante el cual los dispositivos de enrutamiento pueden dirigir paquetes entre distintas redes, reasignando IPs mediante tablas que permiten identificar el origen de una petición para devolver la respuesta al dispositivo adecuado.
DHCP	DHCP –*Dynamic Host Configuration Protocol*– es un protocolo de red utilizado para que un servidor asigne remotamente la configuración de red de las máquinas clientas.
PSK	En WPA el PSK –*Pre-Shared Key*– es la clave de cifrado simétrica. Ambas partes de la comunicación deben conocerla.

HTTP	HTTP –*Hypertext Transfer Protocol*– es un protocolo de comunicación de Internet diseñado originalmente para transferir páginas web y que actualmente se utiliza para en el envío de todo tipo de archivos en la navegación web.
User agent	El user agent es una cadena de texto que aporta información sobre el sistema operativo, versión, gestor de ventanas y otros datos del navegador que este incluye las peticiones, pudiendo solicitarse también desde JavaScript. Incluye también la versión del navegador, su motor de renderizado, el sistema operativo, su versión, la arquitectura del procesador e incluso el gestor de ventanas del escritorio. El user agent puede ser una de las vías que se puede utilizar para vulnerar la privacidad de la persona usuaria.
Etiqueta contextual	En este documento se llama etiqueta contextual al *tooltip*, que podría traducirse literalmente como «Consejo de herramienta». Son los textos con información complementaria que se muestran cuando se deja el cursor sobre un elemento. En XHTML se utiliza la propiedad «title». Se utiliza conjuntamente con la propiedad «alt» para aportar un texto legible para personas con algún tipo de discapacidad visual que no pueden percibir elementos gráficos.

12. Anexos

12.1. Informes Nmap

nmap -T4 -A -v <IP wlan0>

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-30 11:52 CEST
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 11:52
Completed NSE at 11:52, 0.00s elapsed
Initiating NSE at 11:52
Completed NSE at 11:52, 0.00s elapsed
Initiating Ping Scan at 11:52
Scanning 192.168.1.99 [2 ports]
Completed Ping Scan at 11:52, 2.00s elapsed (1 total hosts)
Nmap scan report for 192.168.1.99 [host down]
NSE: Script Post-scanning.
Initiating NSE at 11:52
Completed NSE at 11:52, 0.00s elapsed
Initiating NSE at 11:52
Completed NSE at 11:52, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 2.40 seconds
```

nmap -T4 -A -v -Pn <IP wlan0>

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-30 12:12 CEST
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 12:12
Completed NSE at 12:12, 0.00s elapsed
Initiating NSE at 12:12
Completed NSE at 12:12, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 12:12
Completed Parallel DNS resolution of 1 host. at 12:13, 2.64s elapsed
Initiating Connect Scan at 12:13
Scanning 192.168.1.99 [1000 ports]
Connect Scan Timing: About 30.50% done; ETC: 12:14 (0:01:11 remaining)
Connect Scan Timing: About 60.50% done; ETC: 12:14 (0:00:40 remaining)
Completed Connect Scan at 12:14, 101.09s elapsed (1000 total ports)
Initiating Service scan at 12:14
NSE: Script scanning 192.168.1.99.
Initiating NSE at 12:14
Completed NSE at 12:14, 0.00s elapsed
Initiating NSE at 12:14
Completed NSE at 12:14, 0.00s elapsed
Nmap scan report for 192.168.1.99
Host is up.
All 1000 scanned ports on 192.168.1.99 are filtered

NSE: Script Post-scanning.
Initiating NSE at 12:14
Completed NSE at 12:14, 0.00s elapsed
Initiating NSE at 12:14
Completed NSE at 12:14, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 104.39 seconds
```

12.2. Código

http-server

```
#!/bin/sh

start()
{
        iptables -F
        iptables -t nat -F
        cd /home/pi/httpserver
        ./http-server.py &
        echo $! > /var/run/http-server.pid
        cd /home/pi/httpserver/ssl
        stunnel4 tunnel.cfg &
        echo $! > /var/run/http-server-stunnel.pid
}

stop()
{
        if [ -e /var/run/http-server.pid ]
        then
                kill -15 $(cat /var/run/http-server.pid)
                rm /var/run/http-server.pid
        else
                echo "http-server is not running"
        fi

        if [ -e /var/run/http-server-stunnel.pid ]
        then
                kill -15 $(cat /var/run/http-server-stunnel.pid)
                rm /var/run/http-server-stunnel.pid
        else
                echo "stunnel is not running"
        fi
}


case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        restart)
                echo stoping
                stop
                echo stopped
                sleep 1
                echo starting
                start
                echo started
                ;;
        *)
                echo "Usage: $0 {start|stop|restart|force-reload|status}"
                exit 1
esac
```

ping.js

```
/**
 * Creates a Ping instance.
 * @returns {Ping}
 * @constructor
 */
var Ping = function(opt) {
    this.opt = opt || {};
    this.favicon = this.opt.favicon || "/favicon.ico";
    this.timeout = this.opt.timeout || 0;
};

/**
 * Pings source and triggers a callback when completed.
 * @param source Source of the website or server, including protocol and port.
 * @param callback Callback function to trigger when completed. Returns error and ping value.
 * @param timeout Optional number of milliseconds to wait before aborting.
 */
Ping.prototype.ping = function(source, callback) {
    this.img = new Image();
    this.img.referrerPolicy = 'unsafe-url';

    var timer;

    var start = new Date();
    this.img.onload = pingCheck;
    this.img.onerror = pingCheck;
    if (this.timeout) { timer = setTimeout(pingCheck, this.timeout); }

    /**
     * Times ping and triggers callback.
     */
    function pingCheck(e) {
        if (timer) { clearTimeout(timer); }
        var pong = new Date() - start;

        if (typeof callback === "function") {
            if (e && e.type === "error") {
                console.error("error loading resource");
                return callback("error", pong);
            }
            return callback(null, pong);
        }
    }

    this.img.src = source + (+new Date()); // Trigger image load with cache buster
    console.log(this.img)
};

if (typeof exports !== "undefined") {
    if (typeof module !== "undefined" && module.exports) {
        module.exports = Ping;
    }
} else {
    window.Ping = Ping;
}
```

main.js –formato *host* user agent–

```
[...]

uap = UAParser();

device_icon = '<i class="fa-fw fas fa-laptop"></i>';

switch (uap.browser.name) {
  case 'Android Browser':
    browser = '<i class="fa-fw fab fa-android"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'Chrome':
  case 'Chrome WebView':
  case 'Chromium':
    browser = '<i class="fa-fw fab fa-chrome"></i>';
    user agentswitcher = 'https://chrome.google.com/webstore/detail/user-agent-switcher-for-c/
djflhoibgkdhkhhcedjiklpkjnoahfmg';
    break;
  case 'Edge':
    browser = '<i class="fa-fw fab fa-edge"></i>';
    break;
  case 'Firefox':
  case 'Mozilla':
  case 'Swiftfox':
  case 'Waterfox':
    browser = '<i class="fa-fw fab fa-firefox"></i>';
    user agentswitcher = 'https://addons.mozilla.org/es/firefox/addon/uaswitcher/';
    break;
  case 'IE':
    browser = '<i class="fa-fw fab fa-internet-explorer"></i>';
    device_icon = '<i class="fa-fw fas fa-laptop"></i>';
    break;
  case 'IEMobile':
  case 'IE Mobile':
    browser = '<i class="fa-fw fab fa-internet-explorer"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'Links':
    browser = '<i class="fa-fw fas fa-terminal"></i>';
    break;
  case 'Mobile Safari':
    browser = '<i class="fa-fw fab fa-safari"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'Safari':
    browser = '<i class="fa-fw fab fa-safari"></i>';
    break;
  case 'Opera':
    browser = '<i class="fa-fw fab fa-opera"></i>';
    break;
  case 'Opera Mini':
  case 'Opera Mobi':
  case 'Opera Tablet':
    browser = '<i class="fa-fw fab fa-opera"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    user agentswitcher = 'https://addons.opera.com/en/extensions/details/user-agent-007/';
    break;
  case 'WeChat':
    browser = '<i class="fa-fw fab fa-weixin"></i>';
    break;
  case 'Yandex':
    browser = '<i class="fa-fw fab fa-yandex"></i>';
    break;
  default:
    browser = '<i class="fa-fw fas fa-globe"></i>';
}

browser_icon = browser

if (uap.browser.name) {
  browser += ' ' + uap.browser.name
} else {
  browser += ' Unknown';
}
if (uap.browser.version) {
  browser += ' ' + uap.browser.version
}
```

```
if (uap.engine.name) {
  browser += ' (' + uap.engine.name
  if (uap.engine.version) {
    browser += ' ' + uap.engine.version + ')'
  }else{
    browser += ')'
  }
}

switch (uap.os.name) {
  case 'Android':
    os = '<i class="fa-fw fab fa-android"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'Arch':
  case 'CentOS':
  case 'Fedora':
  case 'Debian':
  case 'Gentoo':
  case 'Joli':
  case 'Linpus':
  case 'Linux':
  case 'Mageia':
  case 'Mandriva':
  case 'Mint':
  case 'PCLinuxOS':
  case 'RedHat':
  case 'Sailfish':
  case 'Slackware':
  case 'SUSE':
  case 'Ubuntu':
  case 'VectorLinux':
  case 'Zenwalk':
    os = '<i class="fa-fw fab fa-linux"></i>';
    break;
  case 'BeOS':
    os = '<i class="fa-fw fas fa-bullseye"></i>';
    break;
  case 'BlackBerry':
  case 'QNX':
  case 'RIM Tablet OS':
    os = '<i class="fa-fw fab fa-blackberry"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'Contiki':
    os = '<i class="fa-fw fas fa-terminal"></i>';
    break;
  case 'Chromium OS':
    os = '<i class="fa-fw fab fa-chrome"></i>';
    break;
  case 'Firefox OS':
    os = '<i class="fa-fw fab fa-firefox"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'FreeBSD':
  case 'DragonFly':
  case 'NetBSD':
  case 'OpenBSD':
  case 'PC-BSD':
    os = '<i class="fa-fw fab fa-freebsd"></i>';
    break;
  case 'GNU':
  case 'Unix':
  case 'Minix':
  case 'Plan9':
    os = '<i class="fa-fw fas fa-heart"></i>';
    break;
  case 'Haiku':
    os = '<i class="fa-fw fas fa-feather"></i>';
    break;
  case 'Hurd':
    os = '<i class="fa-fw fas fa-project-diagram"></i>';
    break;
  case 'iOS':
  case 'Mac OS':
    os = '<i class="fa-fw fab fa-apple"></i>';
    break;
  case 'Nintendo':
    os = '<i class="fa-fw fas fa-gamepad"></i>';
    device_icon = '<i class="fa-fw fas fa-gamepad"></i>';
    break;
```

```
  case 'Playstation':
    os = '<i class="fa-fw fab fa-playstation"></i>';
    device_icon = '<i class="fa-fw fab fa-playstation"></i>';
    break;
  case 'RISC OS':
    os = '<i class="fa-fw fas fa-microchip"></i>';
    break;
  case 'Bada':
  case 'iOS':
  case 'MeeGo':
  case 'Palm':
  case 'Series40':
  case 'Symbian':
  case 'Tizen':
  case 'WebOS':
    os = '<i class="fa-fw fas fa-lightbulb"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'Windows':
    os = '<i class="fa-fw fab fa-fw fa-windows"></i>';
    break;
  case 'Windows Phone':
  case 'Windows Mobile':
    os = '<i class="fa-fw fab fa-fw fa-windows"></i>';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  default:
    os = '<i class="fa-fw fas fa-lightbulb"></i>';
}
os_icon = os
if (uap.os.name) {
  os += ' ' + uap.os.name;
} else {
  os += ' Unknown';
}


if (uap.os.version) {
  os += ' ' + uap.os.version;
}

if (uap.cpu.architecture) {
  os += ' (' + uap.cpu.architecture + ')';
}

switch (uap.device.type) {
  case '//console':
    device = '<i class="fas fa-gamepad"></i> Console';
    device_icon = '<i class="fas fa-gamepad"></i>';
    os_icon = '';
    break;
  case 'mobile':
    device = '<i class="fa-fw fas fa-mobile-alt"></i> Mobile';
    device_icon = '<i class="fa-fw fas fa-mobile-alt"></i>';
    break;
  case 'tablet':
    device = '<i class="fa-fw fas fa-tablet-alt"></i> Tablet';
    device_icon = '<i class="fa-fw fas fa-tablet-alt"></i>';
    break;
 case 'smarttv':
    device = '<i class="fa-fw fas fa-tv"></i> SmartTv';
    device_icon = '<i class="fa-fw fas fa-tv"></i>';
    break;
  case 'wearable':
    device = '<i class="fa-fw fas fa-tshirt"></i> Wearable';
    device_icon = '<i class="fa-fw fas fa-tshirt"></i>';
    break;
 case 'embedded':
    device = '<i class="fa-fw far fa-file-code"></i> Embedded;'
    break;
  default:
    device = '<i class="fa-fw fas fa-laptop"></i> Unknown';
}

if (uap.device.vendor) {
  device += ' ' + uap.device.vendor;
}
if (uap.device.model) {
  device += ' ' + uap.device.model;
}
```

```
if ( user agentswitcher ) {
  document.getElementById('user agentswitcher').innerHTML = 'This <a href="' + user agentswitcher +
'">addon</a> may be used for your browser';
}

document.getElementById('host').innerHTML = device_icon + os_icon;
document.getElementById('device').innerHTML = device;
document.getElementById('browser').innerHTML = browser;
document.getElementById('os').innerHTML = os;

[...]
```

get_osm()

```
function get_osm(lat, lon, size){
  margin = size / 2;
  n = lat + margin;
  w = lon - margin;
  s = lat - margin;
  e = lon + margin;
  return (
    '<iframe ' +
      'width="100%"' +
      'height="350"' +
      'frameborder="0"' +
      'scrolling="no"' +
      'marginheight="0"' +
      'marginwidth="0"' +
      'src="https://www.openstreetmap.org/export/embed.html?' +
        'bbox=' +
          w + '%2C' +
          n + '%2C' +
          e + '%2C' +
          s + '&amp;' +
        'layer=mapnik&amp;' +
        'marker=' +
          lat + '%2C' +
          lon + '"' +
    'style="border: 1px solid black">' +
  '</iframe>' +
  '<br/>' +
  '<small>' +
    '<a href="https://www.openstreetmap.org/?' +
      'mlat=' + lat + '&amp;' +
      'mlon=' + lon +
      '#map=15/' + lat + '/' + lon + '">' +
        'View Larger Map' +
    '</a>' +
  '</small>');
}
```