INVESTIGACIÓN

Herramientas para la detección de redes WiFi falsas en la Empresa de Telecomunicaciones de Cuba

Tools for the detection of fake WiFi Telecommunications Company of Cuba networks

Ing. Sebastián Rodríguez Méndez[1]

Recibido: 11/2017 | Aceptado: 02/2018

PALABRAS CLAVE
Wifi
Android
Seguridad
Protección

RESUMEN

Esta investigación surge debido al hurto de cuentas de Internet que sufren los clientes de la Empresa de Telecomunicaciones de Cuba SA.(ETECSA) al conectarse a redes WiFi públicas falsas, lo que implica que la empresa tenga que reponer dichas cuentas. El objetivo de esta investigación es crear un sistema que permita eliminar el robo de cuentas de Internet mediante herramientas que determinen las conexiones verdaderas. Por esta razón se implementó una aplicación móvil que corre sobre sistema operativo Android, un analizador de tramas Ethernet y dos servidores web desplagados en Nginx, que detectan y evitan que el usuario se conecte a puntos de acceso de ETECSA falsos, brindando un mayor grado de seguridad a la hora de usar el servicio, aumentando considerablemente el grado de satisfacción del cliente, además de la total eliminación del robo de saldos de las cuentas nauta de clientes que se conectan en dichas zonas wifi, fomentando así una de las primicias fundamentales¸ la cual es la protección al consumidor.

KEYWORDS
Wifi
Android
Security
Protection

ABSTRACT

This investigation arises due to the theft of Internet accounts suffered by ETECSA clients when connecting to fake public WiFi networks, which implies that the company has to replenish those accounts. So the goal of this research is to eliminate the theft of Internet accounts using tools to determine the true connections. For this reason, a mobile application was implemented that runs on Android operating system, an Ethernet frame analyzer and two web servers deployed in Nginx, which detect and prevent the user from connecting to false ETECSA access points, providing a greater degree of security at the time of using the service, significantly increasing the degree of customer satisfaction, also completely eliminates the theft of balances from the client accounts that are connected in these WiFi zones, thus promoting one of the fundamental scoops¸ which is consumer protection.

Empresa de Telecomunicaciones de Cuba SA., Cuba. sebastian.rodriguez@etecsa.cu

ISSN: 1813-5056 Vol. 14, No. 1 , enero-julio, pp. 42 - 46, 2018 tono

Introducción

La Empresa de Telecomunicaciones de Cuba S.A. (ETECSA), es una Asociación Económica Internacional, bajo la modalidad de Empresa Mixta, constituida e inscrita en el año 1994 en el Registro de Asociaciones Económicas, hoy Registro Mercantil Central. (ETECSA. Vicepresidencia de Economía, 2010). Tiene la misión de brindar servicios de telecomunicaciones que satisfagan las necesidades de los clientes y la población, así como respaldar los requerimientos del desarrollo socio-económico del país.

ETECSA, además de los servicios tradicionales de telefonía que ofrece, también está orientada a la comercialización, operación, mantenimiento, gestión, desarrollo tecnológico e inversiones de los productos y servicios relacionados con el acceso y utilización de Internet y la provisión de servicios de transmisión de datos a través de sus redes públicas, que se ofrecen en las salas de navegación de ETECSA, Joven Club, oficinas de correo, hoteles, aeropuertos y en las áreas WiFi (ETECSA, 2017).

Dichas áreas WiFi son las más empleadas por los clientes, y estas a su vez son más propensos al robo de sus cuentas de navegación, debido a la creación de redes falsas cuyo nombre es similar a las empleadas por ETECSA. Por lo tanto, es necesario brindar a los usuarios una herramienta que evite la conexión a dichas redes falsas, para esto se implementaron una aplicación móvil y un conjunto de herramientas web.

Materiales y métodos

Para la realización de la aplicación móvil se utilizó el editor de Google, Android Studio en su versión 2.3, dicho software libre le brinda al desarrollador numerosas ventajas, entre las cuales se encuentra el completamiento de código mediante Intellisense, también posee un editor de interfaces visuales WYSIWYG (lo que ves es lo que obtienes).

Fue empleado el popular lenguaje de programación Java en su versión 8 para la funcionalidad de la aplicación móvil y para la creación de la interfaz visual el lenguaje XML, dicha interfaz fue diseñada siguiendo la normativa Material Design desarrollada por Google, el cual permite un diseño más limpio y brinda un efecto de profundidad que ofrece más atractivo visual al usuario final, lo que implica el uso de fuentes y colores específicos.. Tanto para la implementación del frontend, sniffer y backend se empleó el IDE P y Charm de Jetbrains, utilizando como lenguajes de programación Javascript y Python respectivamente, en el frontend fue empleado el framework React 16 (Facebook, 2018a) y en el backend el framework Django 2.0 (Django, 2017). El backend será desplegado en el servidor web Nginx (Nginx, 2018), debido a que el sistema debe recibir un gran número de peticiones simultáneas y dicho servidor permite manejar eficientemente una alta concurrencia de usuarios.

El proceso para la detección de si el AP seleccionado pertenece a ETECSA es mostrado en la figura 1.

Al realizar la conexión se envía al servidor una petición HTTP (enviando datos en JSON) la cual incluye, la mac del AP el cual se desea conectar, la mac de la interfaz WiFi del terminal móvil y el listado de aps detectados cuyo SSID sea WiFi_ETECSA.

Detector de tramas Ethernet

Primeramente el programa sniffer o detector de tramas Ethernet que detecta automáticamente peticiones HTTP obtiene de la trama Ethernet, la dirección

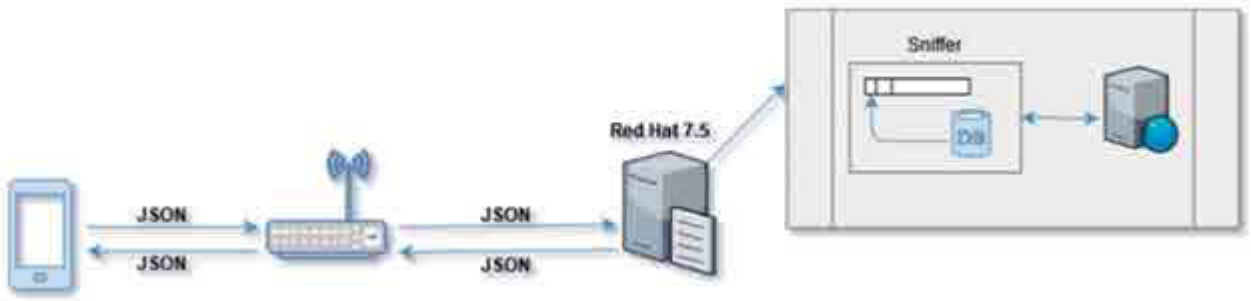

Figura 1. Diagrama del proceso de detección de ap verdadero de ETECSA

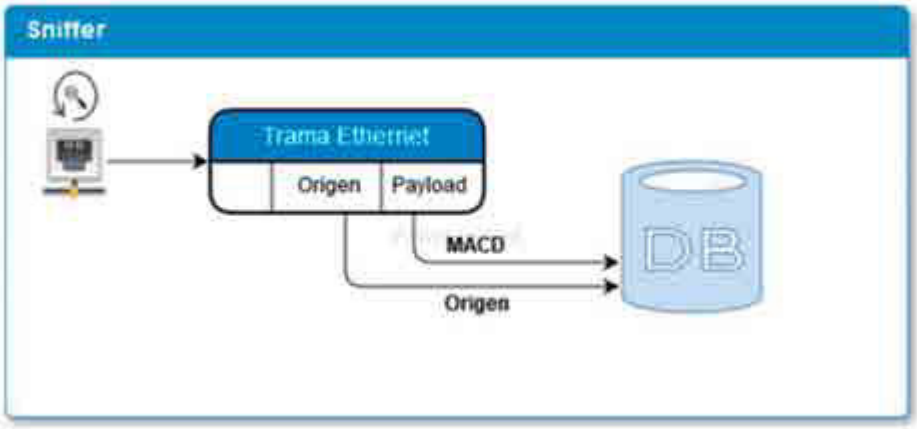

Figura 2. Diagrama del programa sniffer.

mac origen y la almacena en base de datos cuyo identificador es la mac de la interfaz del dispositivo móvil, el objetivo es poder almacenar la dirección de la mac origen para su posterior comparación en el *backend* para detectar si la señal se está replicando (Biondi, 2018), el proceso del sniffer se puede observar en la figura 2.

Servidor web que analiza las peticiones de la aplicación móvil

El *backend* verifica en primera instancia si la mac del AP conectado pertenece a la empresa, en caso de no ser así, retorna un código de error perteneciente a los códigos de estado de las API REST; en caso de que pertenezca, se comprueba la mac de la interfaz del dispositivo con la de la trama Ethernet almacenada; en caso de ser diferentes, es que se está replicando la señal haciendo *spoofing*, y también es retornado un código de error, si todas las condiciones son cumplidas significa que el AP donde el cliente se ha conectado pertenece a ETECSA, almacenando así todos las macs de los AP detectados en dicha zona WiFi para su posterior análisis.

En dependencia del código devuelto la aplicación móvil decide si permitir la conexión o no, posteriormente se permite que el usuario ingrese sus credenciales nauta, en caso de que el certificado del portal nauta sea inválido se genera un error y la aplicación se desconecta automáticamente del ap seleccionado.

En el backend desarrollado utilizando el framework Django 2.0 se encuentran todos los *endpoints*, del API Restful (Christie, 2017), que devuelven todos los valores en formato JSON permitiendo que la aplicación se comunique eficazmente tanto con la aplicación móvil como con el *frontend*, cabe resaltar que la base de datos empleada es Postgresql. Uno de estos *endpoints* permite realizar un análisis por zona de los ap detectados, permitiendo detectar remotamente en que zona WiFi existen AP falsos en un momento dado de conexión.

Servidor web para el control de los puntos de acceso

El *frontend* fue desarrollado en React, el cual permite que la aplicación sea Single Page Application (SPA) (Facebook, 2018a), haciendo que la aplicación sea más eficiente, debido a que no se recarga todo el sitio sino las zonas o componentes que requieran ser cambiados en dependencia de la ruta escogida, para la autenticación con el *backend* fue empleado JWT¸ lo cual es más eficiente que la autenticación mediante cookies, dicho *frontend* se comunica mediante la API REST del *backend* para listar, insertar, editar y eliminar los AP de la Empresa.

Se emplea además el patrón Redux (Facebook, 2018b),el cual es una alternativa y mejora del patrón MVC evitando el llamado código spaggetti. Para el diseño del *frontend* se empleó el framework CSS Boostrap 4 de Twitter.

Cabe resaltar que este conjunto de herramientas fue desarrollado siguiendo las tendencias de desarrollo web utilizadas en la actualidad, permitiendo que la aplicación sea escalable, con mejor desempeño, debido a que se emplean dos servidores -uno para el *frontend* y otro para el *backend*- que serán alojados en Nginx en la distribución de Linux Red Hat 7.5.

Figura 3. Interfaz visual de la aplicación móvil.

Además al emplear API REST hace que sea menor el *payload* debido a que en cada petición al servidor no se descargue todo los archivos js y css, simplemente los datos en JSON que posteriormente son renderizados por el frontend.

Resultados y discusión

Como resultado de la investigación se implementó una aplicación móvil cuyo nombre es ENP que corre sobre sistema operativo Android la cual se muestra en la Figura 3. Esta aplicación muestra una lista con los AP cuyo SSID es WIFI_ETECSA que se encuentran irradiando en la zona, donde el usuario puede establecer la conexión con el AP seleccionado, desencadenando todo el proceso de verificación de veracidad de dicho AP.

Además, se logró la implementación del *backend*, el cual posee todos los *endpoints* para listar, insertar, editar y eliminar los AP utilizados por la Empresa. También permite realizar análisis por zonas utilizando los listados de AP enviados por cada usuario mediante la aplicación móvil, dicho análisis resulta de gran utilidad ya que arroja como resultados en qué zona WiFi se intenta cometer fraude mediante *spoofing* o usando el mismo SSID empleado por ETECSA.

El *frontend* implementado, llamado CAPS, permite la autenticación en el sistema, la cual es mostrada en la figura 4. También permite listar todos los AP como se observa a continuación en la figura 5, aplicarles filtros de búsqueda, insertar nuevos AP (Figura 6), que en caso de que la mac esté siendo utilizado por otro AP se genera un error, se elimina y se edita en dependencia de los roles de usuario. Además, permite realizar el análisis remoto de las zonas donde se han conectado clientes usando la aplicación móvil, mostrado en la figura 7, lo que permite que se conozca en qué zona existen AP falsos.

Las herramientas utilizadas se le realizaron pruebas de testeo para detectar *bugs* mediante la utilización de Jtest, PyTest así como de *profiling* para lograr un producto más eficiente y rápido.

Conclusiones

Con la investigación realizada se demostró la necesidad, tanto de la Empresa de Telecomunicaciones como de sus usuarios, de lograr que este tipo de hechos no sucedan, por lo que se realizó una aplicación móvil y un conjunto de herramientas web que permiten bloquear las redes WiFi falsas. Cabe resaltar que hasta el momento no se conoce de la existencia de aplicaciones similares, lo cual representa un logro, además de los beneficios en cuanto a seguridad y fiabilidad que brinda a los usuarios que utilizan el servicio de Internet mediante las redes WiFi públicas de ETECSA, protegiendo las cuentas nautas de los clientes, contribuyendo enormemente a la protección al consumidor e influyendo directamente en su satisfacción, así como en la imagen de la empresa.

Figura 4. Autenticación mediante la aplicación web CAPS.

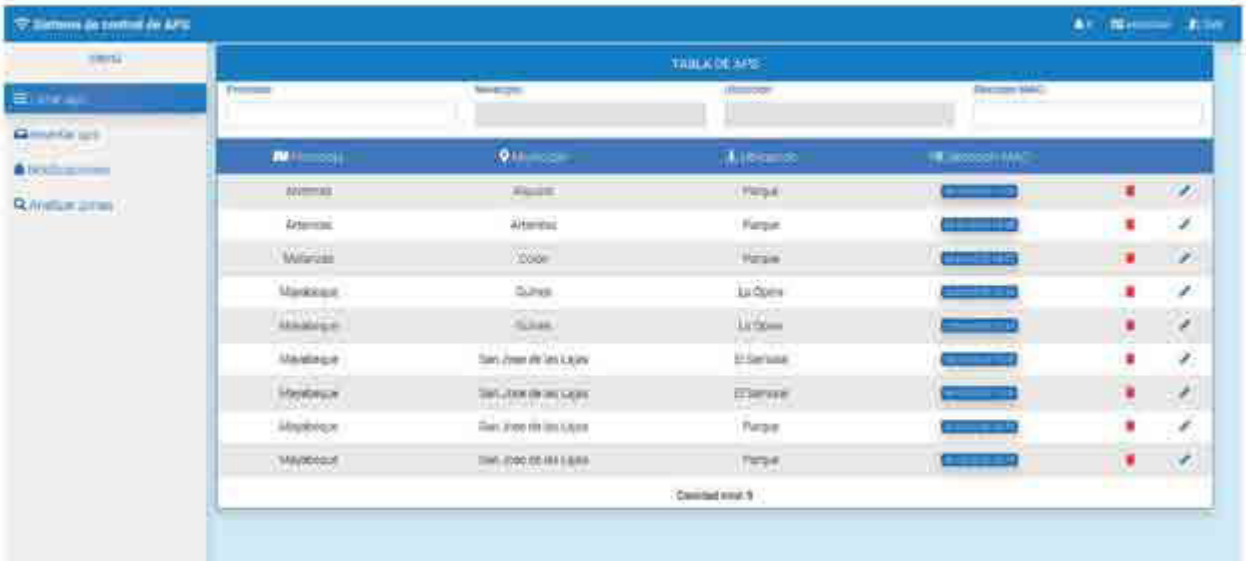

Figura 5. Listado de AP en cada zona.

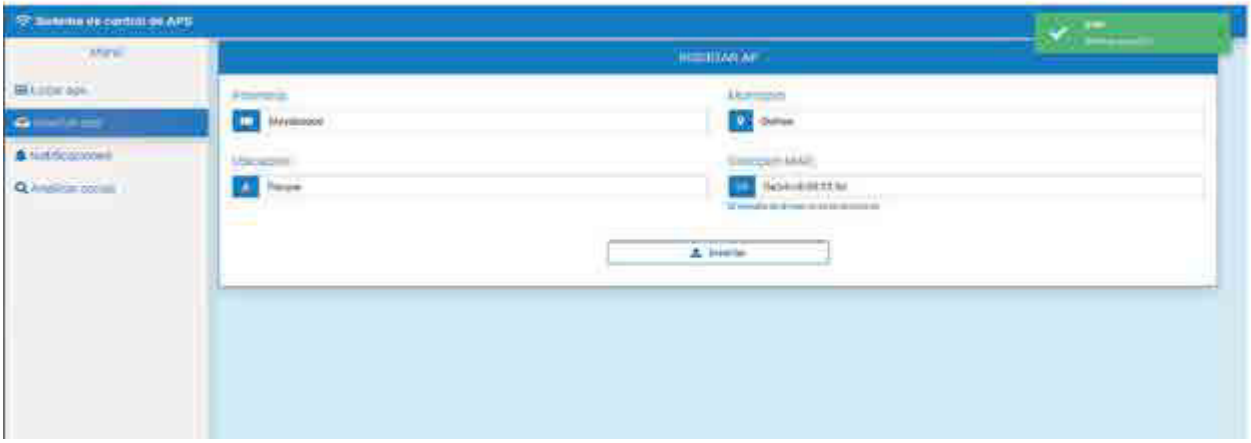

Figura 6. Insertar AP en zona WiFi.

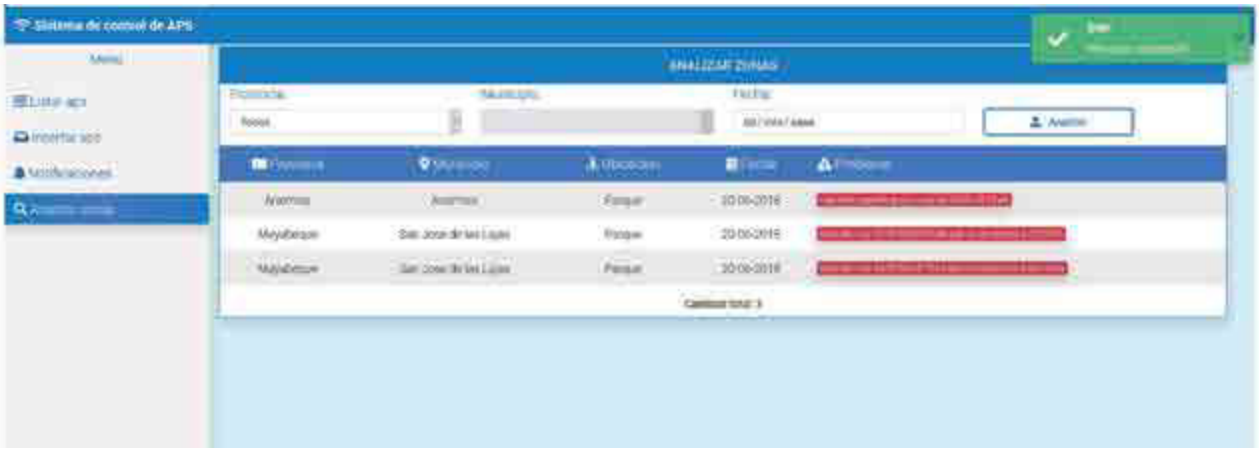

Figura 7. Análisis de zonas donde se han conectado clientes usando la aplicación móvil.

ISSN: 1813-5056 Vol. 14, No. 1 , enero-julio, pp. 41 - 46, 2018 tono

Referencia

Biondi, P. (2018). Scapy. Packet crafting for Python2 and Python3. Consultado 10 de febrero de 2018, disponible en: https://scapy.net/.

Christie, T. (2017). Django Rest Framework. Consultado 10 de marzo 2017, disponible en: http://www.django-rest-framework.org/.

Django. (2017). Django Web Framework. Consultado 3 de marzo de 2017, disponible en: https://www.djangoproject.com/.

ETECSA. (2017). Internet y conectividad. Consultado 31 de mayo de 2017, disponible en: http://www.etecsa.cu/internet_conectividad/internet.

Facebook. (2018a). React Framework. Consultado 15 de enero de 2018, disponible en: https://reactjs.org/docs/thinking-in-react.html.

Facebook. (2018b). Redux. Consultado 20 de enero de 2018, disponible en: https://es.redux.js.org/.

Nginx. (2018). NGINX | High Performance Load Balancer, Web Server, Reverse Proxy. Consultado 4 de mayo de 2018, disponible en: https://www.nginx.com.