2018 - MAYO

informes

apisa

Asociación de Profesionales de Informática Sanitaria de Andalucía

SEGURIDAD - 03

seguridad 03

Seguridad IoT en Sanidad: ¿estamos preparados?

Presentación

INFORMES APISA

SERIE SEGURIDAD

3

Seguridad IoT en Sanidad: ¿Estamos preparados?

Autores:

Miguel Ángel Arroyo Moreno, Josep Bardallo Gay, Juan José Domenech Sánchez, Francisco Jesús Gómez López, Ramón Salado Lucena

Año 2018

La presente edición ha sido posible con la colaboración de las empresas: ACCENTURE, CARESTREAM HEALTH SPAIN, CHECKPOINT SOFTWARE TECHNOLOGIES, DELL EMC, FUJITSU, INDRA, ISOTROL, MICROSOFT, NEW DOORS, PULSIA TECHNOLOGIES, QUANTION, SEMIC EFECTIVE IT SOLUTIONS, SOLUTIA IT, SYMANTEC, T-SYSTEM.

Coordinación de la edición y maquetación:
Miguel Ángel Arroyo
Francisco Jesús Gómez López

Portada diseñada y cedida a APISA por: José Miguel Pla Ruiz

Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 3.0 Unported.

Edita: APISA, Asociación de Profesionales de Informática Sanitaria de Andalucía. 4 de mayo de 2018

www.apisa.org.es

1ª edición

ISBN:

DL:

Impreso en España

Seguridad IoT en Sanidad: ¿Estamos preparados?.

Agradecimientos

Desde la Junta Directiva de la Asociación de Profesionales de Informática de la Salud de Andalucía queremos agradecer el esfuerzo y la dedicación de los autores de este libro. Sin su voluntariedad y su compromiso por la divulgación del conocimiento sobre las particularidades de la informática en el sector sanitario no sería posible que nuestra asociación cumpliera con su objetivo de publicar un libro especializado para el sector cada año.

Queremos agradecer especialmente su colaboración a Vicente Aguilera, Fundador y Presidente del capítulo OWASP España, El proyecto abierto de seguridad en aplicaciones Web, que se ha prestado amablemente a prologar este libro, el tercero de la serie de seguridad de la información que publica APISA. Sin duda, el papel de las asociaciones sin ánimo de lucro es de gran importancia para la mejora continua en la seguridad y la calidad de los productos que desarrollamos y mantenemos. Muchísimas gracias Vicente.

Agradecemos asimismo a las empresas que patrocinan las actividades de APISA, ACCENTURE, CARESTREAM HEALTH SPAIN, CHECKPOINT SOFTWARE TECHNOLOGIES, DELL EMC, FUJITSU, INDRA, ISOTROL, MICROSOFT, NEW DOORS, PULSIA TECHNOLOGIES, QUANTION, SEMIC EFECTIVE IT SOLUTIONS, SOLUTIA IT, SYMANTEC, T-SYSTEM, sin cuya colaboración difícilmente se podrían abordar tareas como la que representa esta publicación.

APISA
Junta Directiva

Presentación

Presentación de la Serie: Seguridad

Desde sus comienzos, las TIC no han dejado de vivir una revolución tras otra. Al principio, estos cambios solían venir de la mano de los cambios tecnológicos, que nunca han dejado de sorprendernos año tras año, y que con toda seguridad seguirá siendo así en el futuro. Sin embargo, recientemente, a estos cambios tecnológicos se están sumando nuevas "revoluciones", más relacionadas con la información y su tratamiento, que con la tecnología. La información cobra mayor protagonismo y, en cierto modo, promueve estos nuevos tipos paralelos de revoluciones en las TIC. Fenómenos con el Big Data, el Cloud Computing o el Internet de las Cosas, no dejan de madurar y crecer en nuevas utilidades y prestaciones a nuestra sociedad. Y acompañando a estos fenómenos, venimos viviendo también una creciente preocupación por la calidad de los productos TIC y la seguridad de la información, conceptos íntimamente relacionados, por supuesto, y que no solo preocupan ya a los profesionales del sector, sino también a los usuarios finales de los productos que estos desarrollan.

Ya en 2013, APISA publicó el nº 1 de esta serie de seguridad de la información, "***Mitos y leyendas de la LOPD***". En 2017 publicamos el 2º número de esta serie, un poco a caballo entre la calidad y la seguridad de la información, y mucho más centrado en el sector sanitario, "***Impacto de las Tecnologías de la Información en la Seguridad del Paciente***".

En apenas veinte días después de la presentación de este libro asistiremos a la entrada en vigor efectiva del Nuevo Reglamento Europeo de Protección de datos, y a la aplicación obligatoria para

toda la Administración Pública del Esquema Nacional de Seguridad. Ambas regulaciones basadas en buenas prácticas reconocidas internacionalmente. Legislación mucho más madura y exigente con la seguridad de la información que la actual Ley Orgánica de Protección de Datos, que también será sustituida en breve, y que el anterior Real Decreto de Medidas de Seguridad ya sustituido por el actual Reglamento Europeo.

Ahora, las regulaciones incluyen conceptos como "Análisis y Gestión de Riesgos", "Amenaza", "Riesgo Potencial", "Riesgo Residual", "Sistemas de Gestión de Seguridad de la Información", "Controles de Seguridad", "Rendición de Cuentas" …. Términos más propios de una jerga, hasta ahora muy específica, de un grupo profesional siempre relacionado con la seguridad de las TIC, y que ahora transciende a las leyes y a otros grupos profesionales relacionados con el derecho.

La creciente "invasión" de las tecnologías en todos los planos de la vida de los ciudadanos está conllevando, además, una nueva conciencia sobre esa seguridad, que sigue cobrando más y más protagonismo. Y en esta creciente preocupación por la seguridad de la información, nuestros profesionales han querido volver sobre el tema, y no será la última vez, escribiendo este tercer número de la serie que aquí presentamos, "**Seguridad IoT en Sanidad: ¿Estamos preparados?"**

Sin ninguna duda, el compromiso de APISA y de nuestros profesionales, por la calidad y la seguridad no podía dejar de tener su reflejo una vez más en la forma de una nueva publicación. Esta serie, a través de los tres ejemplares publicados en los últimos años, también comienza a reflejar de alguna forma, la enorme complejidad y variedad de esta rama de las TIC que es la seguridad de la información. Diversas caras y muchas aristas y ángulos deben abordarse si queremos trabajar con sistemas seguros. Varias especialidades profesionales, solo dedicadas a la seguridad de la información, son ya necesarias para abordarla. En cada una de las ramas del Desarrollo, el gobierno, la gestión, los sistemas, las comunicaciones, la privacidad…. ya están haciendo falta

profesionales especializados, y es seguro que continuarán apareciendo nuevas oportunidades profesionales en este campo.

El sector sanitario, por supuesto, no está al margen de esta creciente necesidad. En este número en concreto, se analizan el "IoT", Internet de las Cosas, que en sanidad está presentando numerosas oportunidades a los profesionales sanitarios para el seguimiento de los pacientes. Cada día más dispositivos aparecen para ayudar a médicos y pacientes en el tratamiento y seguimiento de sus enfermedades. Sólo unas semanas antes de la publicación de este número, la presidenta de la Junta de Andalucía anunciaba la inclusión en la cartera de servicios de salud de dispositivos "wereables", que ayudarán a los niños diabéticos andaluces a controlar su nivel de azúcar en sangre. Por un lado, informando al niño de forma continua de los niveles de azúcar en su teléfono móvil, y por otro, permitiendo a su médico y a su pediatra conocer toda la información relativa al problema de salud del menor.

Pero como decíamos antes, la tecnología puede tener muchas caras y aristas, y en este caso, mantener la integridad de los datos, garantizar que no puedan ser manipulados, de forma intencionada o por error, y garantizar la privacidad de los pacientes, cuyos dispositivos envían su información de salud de forma continuada, resulta una obligación no solo legal, sino también ética para nuestros profesionales. Y desde luego, todo un reto.

En este número, los autores hacen una aproximación técnica de esta problemática, analizando aspectos como la seguridad desde el diseño en el desarrollo, en la WiFi, el Bluetooh o la seguridad en las aplicaciones web. Y una vez más, con esta obra, no pretendemos más, que hacer una humilde aportación a la difusión de la mejora, la calidad y la seguridad, en el diseño, la implantación y el uso de las Tecnologías de la Información Sanitarias.

Manuel Jimber del Río
Presidente de APISA

Índice

AGRADECIMIENTOS **7**

PRESENTACIÓN DE LA SERIE: SEGURIDAD **9**

ÍNDICE 12

PRÓLOGO **15**

INTRODUCCIÓN **17**

SITUACIÓN ACTUAL **23**

UN POCO DE HISTORIA 24
LA ERA DE LA TRANSFORMACIÓN DIGITAL 25
SMARTCITIES 28
ESALUD 30
COMPUTACIÓN EN LA NUBE 37
APROXIMACIÓN A LA SEGURIDAD IOT 39
EUROPA 41
ESPAÑA 44
ANDALUCÍA 46

GOBIERNO Y NORMATIVA LEGAL EN ENTORNOS SANITARIOS **53**

INTRODUCCIÓN 54
PRINCIPALES RIESGOS CON LA ADOPCIÓN DE IOT EN ENTORNOS SANITARIOS 55
MEJORES PRÁCTICAS 57
PRIVACIDAD, GDPR Y OTRAS NORMATIVAS APLICABLES EN ENTORNOS IOT 61

SEGURIDAD EN APLICACIONES WEB **69**

TESTING SOBRE APLICACIONES WEB 74
OWASP TOP 10 84

SEGURIDAD EN APLICACIONES IOS 87

INTRODUCCIÓN A LA SEGURIDAD EN IOS.....88
ARQUITECTURA DE SEGURIDAD EN IOS 90
OWASP COMO REFERENCIA 96
TOP 10 DE RIESGOS EN APLICACIONES MÓVILES SEGÚN OWASP.....98
TOP 10 DE CONTROLES DE SEGURIDAD 100
JAILBREAK 102

SEGURIDAD EN APLICACIONES ANDROID 105

INTRODUCCIÓN A LA SEGURIDAD EN ANDROID.....106
ARQUITECTURA DE ANDROID 107
MODELO DE SEGURIDAD DE ANDROID.....110
OWASP - TOP 10 VULNERABILIDADES EN APLICACIONES MÓVILES..114
OWASP Y ENISA - TOP 10 CONTROLES DE SEGURIDAD.....119

SEGURIDAD EN TRÁFICO DE RED 123

SEGURIDAD EN TRÁFICO DE RED (802.3 Y 802.11) 124
DISPOSITIVOS IOT EN LA RED.....136

SEGURIDAD EN BLE Y ZIGBEE 140

BLUETOOTH LOW ENERGY (BLE) 141
EL PROTOCOLO BLE.....145
CONSEJOS DE SEGURIDAD.....149
HERRAMIENTAS PARA DEPURACIÓN.....150
APPS PARA IOS Y ANDROID 154
ZIGBEE 156
ROLES Y REDES ZIGBEE 157
DEBILIDADES EN ZIGBEE.....160

ANÁLISIS DE FIRMWARE.....163

INTRODUCCIÓN AL ANÁLISIS DE FIRMWARE 164
SISTEMAS DE FICHEROS DE LOS ARCHIVOS FIRMWARE.....167
EXTRACCIÓN DEL SISTEMA DE FICHEROS.....168

INGENIERÍA INVERSA.....179

INTRODUCCIÓN A LA INGENIERÍA INVERSA 180

LENGUAJE ENSAMBLADOR EN ARQUITECTURA ARM 181
REFERENCIAS BIBLIOGRÁFICAS 187

GLOSARIO 189

LOS AUTORES 193

Prólogo

Vicente Aguilera Díaz
Fundador y Presidente del capitulo OWASP España

La evolución de la tecnología y, en especial, en el ámbito de la informática y las comunicaciones, permite que la sociedad desempeñe sus tareas de una forma más eficiente y con mayor facilidad, así como llevar a cabo acciones que resultaban inimaginables hace tan sólo unos años. Vivimos en una etapa de transformación digital constante.

En este sentido, los dispositivos IoT (*Internet of Things*) han sido rápidamente adoptados por el conjunto de la sociedad y la industria, y están cambiando la forma de interactuar con nuestro entorno. En el sector hospitalario, su uso se ha visto incrementado en los últimos tiempos. De hecho, los requerimientos para este sector ha llevado a la industria a crear entidades específicas como IoHT (*Internet of Healthcare Things*) o IoMT (*Internet of Medical Things*), con el objetivo de mejorar determinados procesos y la atención al paciente. Así, por ejemplo, es posible usar sensores para monitorizar el estado de salud en tiempo real, monitorizar la temperatura de la habitación, o realizar el seguimiento y gestión de suministros y medicamentos. Esto implica conectar nuevas aplicaciones y dispositivos a los sistemas de información tradicionales.

Es aquí donde, aún teniendo presente los beneficios que IoT aporta (y aportará) al sector hospitalario, no hay que olvidar los riesgos que implica ampliar la exposición de los sistemas de información y la necesidad de garantizar el elevado nivel de seguridad y privacidad requerido en estos entornos.

Este libro aborda todas estas cuestiones, con un enfoque en la seguridad en el uso y despliegue de dispositivos IoT, donde los autores aportan su extensa experiencia en este campo y utilizan proyectos de referencia internacional, como los desarrollados por OWASP (*Open Web Application Security Project*), en sus interesantes argumentaciones. Entran en juego aspectos tan diversos como la seguridad en el desarrollo de aplicaciones móviles, la seguridad en las comunicaciones, uso de configuraciones seguras en los dispositivos, o aspectos relacionados con la privacidad, entre otros. Sin duda, las buenas prácticas recogidas en este libro ayudarán, no sólo a los profesionales relacionados con el ámbito sanitario sino, en última instancia, a la atención y la mejora de la salud de los pacientes.

Mis más sinceras felicitaciones a los autores.

Introducción

Francisco Jesús Gómez López

¿Anoche qué? Otra vez te acostaste tarde, ¿no?

¿Te vas a quedar todo el día ahí sentado? Anda, levántate y muévete un poquito.

¿Qué vas a coger el coche? Hace un día estupendo, así que andandito y utiliza esas dos piernas que falta te hace…

Parece una conversación, bueno o monólogo, de cualquier padre/madre con su hij@ adolescente, ¿verdad? Pues también, pero en este caso estaba haciendo referencia a esas cosas "inteligentes" que tenemos por todos lados y que nos hacen la vida…

☐ Más fácil ☐ Más difícil (Elige la respuesta que consideres más apropiada)

Este es un ejemplo de andar por casa de cómo Internet de las Cosas está haciendo cambiar nuestro día a día como individuos pero que sin duda también lo está haciendo a nivel empresarial. Esta tecnología disruptiva está revolucionando todos los sectores: Energía, Medio Ambiente, Logística… y por supuesto Salud, que están sabiendo aprovechar las ventajas que supone utilizarla en sus negocios y por lo que están modificando sus modelos hacía ella.

Una Salud en la que los ciudadanos cada vez estamos más involucrados en su cuidado, estamos en la época de grandes tendencias como son:

- Envejecimiento Activo (Active Ageing)
- Bienestar mental y físico (Wellness)
- Amigo de la Salud (Health friendly).
- Empoderamiento del paciente (Empowerment)

Queremos participar activamente de su "gestión", por eso cada vez más demandamos que toda la información que generan los tensiómetros, glucómetros, smartwatches,… repercuta en mejoras diagnósticas y en tratamientos ofrecidos por los profesionales sanitarios.

Profesionales sanitarios que a su vez necesitan de un mayor número de herramientas que faciliten su trabajo diario como pueden ser los Sistemas de Apoyo a la Decisión Clínica o bien les ayuden en tareas de investigación mediante el análisis de datos generados.

Así que nuestros centros sanitarios no pueden quedarse atrás en los avances que están llegando y sobre todo están por llegar con Internet de las Cosas (IoT[1]). Sin embargo, las ventajas de disponer de todos los datos que se generen hacen que sea imprescindible establecer una estrategia apropiada en la organización que saque el mayor rendimiento posible a los mismos sean del ámbito que sean. No hay que olvidar que los centros también se pueden beneficiar transversalmente de aplicaciones vinculadas a otros sectores ya que se pueden poner en marcha soluciones orientadas al control de activos, control de accesos, gestión medioambiental, etc.

Nosotros como profesionales del área de Tecnologías de la Información Sanitarias debemos estar a la altura y ser capaces de gestionar esta demanda. Desde un principio la dirección de los centros debe ser consciente de la necesidad de contar con las TIS a la hora de confeccionar proyectos donde se prevea la implementación de soluciones que incluyan dispositivos ligados a "Internet de las Cosas", tal y como ocurren con el resto proyectos TIC ¿no?

Pero, aunque seguro que en algunos centros ya estáis trabajando con IoT, la realidad es que para otros aún es una tecnología nueva con la que no están habituados a trabajar. Habrá que familiarizarse con nuevos dispositivos móviles muy específicos, protocolos de comunicación propios, nuevos estándares de interoperabilidad, metodologías y guías de buenas prácticas algunas ya existentes y por supuesto habrá que dar un paso más en la gestión de la seguridad de la información y privacidad de los datos.

Lo primero que se nos podría ocurrir es echar el freno ante cualquier propuesta de solución vinculada a estos dispositivos para, dado nuestro desconocimiento, evitar posibles incidentes de seguridad.

1 Internet of Things

Pero sabemos que el riesgo cero no existe así que lo mejor es prepararse, conocer la normativa vigente, aprovechar el conocimiento ya adquirido por asociaciones especializadas en seguridad como OWASP o ISACA y adaptar nuestros sistemas para minimizar los posibles daños producidos por potenciales ataques. De esa manera ante la pregunta "¿Estamos preparados?" podremos dar una respuesta afirmativa.

Esperamos que sea de vuestro gusto y que además de aportar algo de conocimiento nos permita hacer una reflexión sobre la posición en la que nos encontramos a día de hoy y sirva de ayuda para establecer una estrategia común en el marco de las TIS. Unas TIS en las que sus profesionales puedan ser partícipes activos en la toma de decisiones y de las que seguro que nuestros centros, tanto los hospitales más grandes como los consultorios más pequeños y recónditos, sean los mayores beneficiados.

Referencias Bibliográficas

1. Advancing the Internet of Things in Europe http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52016SC0110

Situación actual

Francisco Jesús Gómez López

UN POCO DE HISTORIA

Bueno antes de nada voy a hacer un repaso a algunos hechos que han marcado el devenir de Internet de las Cosas hasta convertirse en lo que es hoy en día.

El primer uso de este término es atribuido a un profesional del MIT[ii] (Massachusetts Institute of Technology) por el año 1999 llamado Kevin Ashton. En él hacía mención a la necesidad de un Internet de las Cosas de cara a utilizar una forma estandarizada en que los ordenadores puedan capturar información del mundo real y comprenderlos. Se trataba por entonces de investigar sobre identificación por radiofrecuencia en red (RFID) y tecnología de sensores.

Con el tiempo el término se fue extendiendo entre la comunidad científico-técnica apareciendo ya tanto en prensa como en títulos de diferentes libros por primera vez. Además RFID es desplegada a gran escala por el departamento de defensa norteamericano (2003-2004).

Poco después, en el año 2005 apareció en el mercado un gracioso conejito llamado "Nabaztag" (el conejo Wi-Fi) capaz de avisarte cuando recibías un email o un amigo se conectaba a Google Talk. Además, era capaz de informarte sobre el clima o índices bursátiles y disponía de lector RFID.

Conejo Wifi - Nabaztag[iii]

ii Instituto Tecnológico de Massachusetts

iii By Rama & Musée Bolo - Own work, CC BY-SA 2.0 fr, https://commons.wikimedia.org/w/index.php?curid=37010428

Así fue evolucionando esta tecnología hasta que en el año 2008 ya se había extendido el número de "cosas" u "objetos" conectados a Internet de tal manera que era superior a la población mundial. De hecho para muchos, por esta razón, el 2008 es el año del nacimiento real de una de las tecnologías disruptivas con mayor potencial de los últimos y de los próximos años. Había llegado para quedarse "Internet de las Cosas".

A partir de aquí, se puede considerar casi presente, o al menos parece que lo tenemos algo más fresco ¿o no?

- Lanzamiento de la IPv6
- Crecimiento exponencial de Aplicaciones Móviles
- Aparición de multitud de dispositivos, wearables, sensores,…
- Smarticities
- ...

La era de la transformación digital es ya más que una realidad no solo para las empresas sino también para el ciudadano de a pie.

LA ERA DE LA TRANSFORMACIÓN DIGITAL

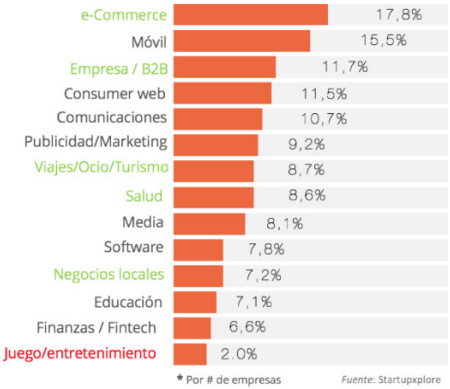
Sin duda nos encontramos en unos años en los que la transformación digital está cambiando tanto los modelos de negocio de las organizaciones como incluso el comportamiento y estilo de vida de los ciudadanos. La aparición de nuevas tecnologías requiere que éstas desarrollen a su vez nuevas estrategias de negocio de manera que puedan ser competitivas en un mercado donde, como se suele decir, en muchos casos hay que renovarse o morir.

Esta transformación se ha llevado a cabo de diferentes formas, una compañías han decidido adaptar su modelo y dar el salto a Internet abriendo por ejemplo una canal de venta online. En otros ha supuesto el cambio de herramientas más tradicionales a otras "2.0"

como Alfresco, Google Drive o Dropbox o potenciando el posicionamiento en las principales redes sociales mediante la creación de perfiles en las mismas.

En el caso de aquéllas vinculadas a tecnología claramente su transformación está derivando en potenciar iniciativas basadas en la movilidad. Por supuesto, esta posibilidad de negocio también está provocando la aparición de numerosas "startups" año tras año.

La inversión en startups ha crecido en 2017 más de un 40% y como se puede ver en la siguiente gráfica aquellas ligadas al sector App/Móvil ocupan un lugar privilegiado en la lista sin olvidar que el sector Salud es uno de los que más creció el año pasado.

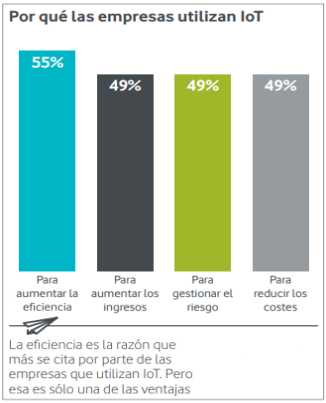


Aunque los principales hubs de startups en España se concentran en Barcelona, Madrid y Valencia, Andalucía no se queda atrás, principalmente en Sevilla y Málaga.

Centrándonos en el caso concreto de IoT, ese crecimiento ocurrido en los últimos años ha provocado que la proporción de compañías que usan esta tecnología se haya más que duplicado. En concreto, se ha incrementado del 12% en 2013 al 29% en 2017. Este dato si nos centramos en Sanidad ha sido del 19% del 2014 al 27% en 2017.

Mientras que en la industria en general un 46% ha integrado IoT con sus sistemas de información central en Sanidad este porcentaje sube hasta el 49%.

Las ventajas para las organizaciones van mucho más allá de la reducción de costes. Se está utilizando IoT para además de reducir éstos, para reducir el riesgo y aumentar los ingresos. Pero el foco principal está en el aumento de la eficiencia (55% de las empresas), que en Sanidad se trata principalmente de la mejora de la calidad asistencial que se les presta a nuestros pacientes.



Fuente: Resumen Ejecutivo Barómetro IoT 2017-18 (Vodafone)

Aunque todavía sigue habiendo cierta resistencia al uso de estas tecnologías en Sanidad estamos empezando a superarlas y a darnos cuenta del impacto positivo de IoT sobre resultado en pacientes, eficiencia operacional y por qué no decirlo, como he mencionado antes a nivel privado, ahorro en costes.

Así que habría que aprovechar las sinergias existentes en el mundo empresarial en general y tecnológico en particular para que las administraciones podamos sacar el mayor beneficio. Y sin duda la Sanidad es uno de los sectores que más rendimiento podrán obtener.

Smartcities

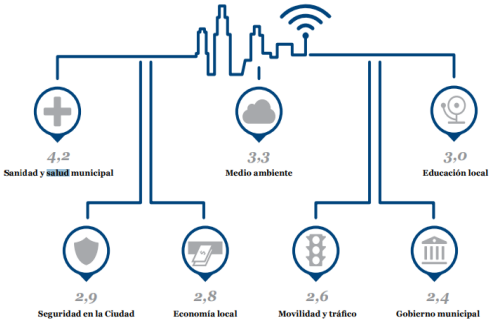
Como comentaba antes en referencia al incremento de startups en España, Andalucía no se queda atrás. Sevilla y Málaga concentran un gran número de estas compañías, sobre todo teniendo en cuenta que ambas están incluidas entre las principales ciudades inteligentes (SmartCities) de Europa.

Esto ha supuesto el desarrollo de múltiples proyectos ligados a tecnología móvil, como Internet de las Cosas y asociados a diferentes sectores entre los que se encuentra "Sanidad y Salud" o como lo define la propia Unión Europea como "Smart living" en el que también se incluye "Seguridad".

UNIÓN EUROPEA	MODELO CENTRO DE INNOVACIÓN DEL SECTOR PÚBLICO DE PWC E IE BUSINESS SCHOOL
Smart environment	Medio ambiente
Smart mobility	Movilidad
Smart living	Seguridad Sanidad y salud
Smart people	Educación
Smart economy	Economía
Smart governance	Gobierno

Fuente: Libro blanco SmartCities (Telefónica)

De hecho "Smart living" no es un ámbito cualquiera sino que es de los más valorados por nuestros ciudadanos. En una encuesta realizada a los ciudadanos ya en el año 2015 al preguntar por el ámbito de la gestión de su ciudad que considera más importante, resultaba que "Sanidad y salud municipal" obtenía la mayor puntuación (4'2/5'0) muy por encima de otras como Medio Ambiente (3'5/5'0) o Educación local (3'0/5'0).



Fuente: Libro blanco SmartCities (Telefónica)

Continuando con esta misma encuesta, al preguntar por los servicios que, dentro de Sanidad, tendrían mayor aceptación en su ciudad, incluso siendo de pago se encuentran:

- Teleasistencia
- Programas de salud: autocuidado y enfermos crónicos
- Gestión de la demanda asistencial

Estas encuestas ponen en relieve la necesidad de dar cobertura a esa demanda. En un escenario marcado por el envejecimiento de la población y la proliferación de enfermedades crónicas, la tecnología es un mecanismo cada vez más necesario para optimizar los recursos de asistencia sanitaria y disminuir sus costes.

Las soluciones y servicios tecnológicos que se están ofreciendo abordan diferentes áreas de actuación:

- La provisión sanitaria, principalmente en dispositivos e instrumentación.
- La salud y el bienestar individual, mediante programas de salud (atención cardiovascular, diabetes, wellness,…) con objeto de fomentar hábitos de vida saludable.

- Colectivos en situación de dependencia y seguimiento remoto de personas dependientes: teleasistencia, localización, alarmas técnicas, televigilancia y localización, seguimiento y presencia monitorizados.
- El bienestar del conjunto de la sociedad, con servicios de alertas y emergencias sanitarias basados en entornos de open data que facilitan la toma de decisiones.
- La gestión asistencial, relacionada en muchos casos con entornos de datos abiertos que optimizan la información (gestión de listas de espera, programación de la oferta asistencial, acceso a historial e informes clínicos,…).

En resumen, habría que aprovechar iniciativas como las de Smartcities ya que podrían servir de catalizador para poner en marcha nuevos proyectos tecnológicos en Sanidad vinculados a eSalud.

eSalud

Se define, según la OMS, como el uso de las tecnologías de la información y la comunicación (TIC) para la salud. En un sentido más amplio, hace referencia a que la eSalud se ocupa de mejorar el flujo de información, a través de medios electrónicos, para apoyar la prestación de servicios de salud y la gestión de los sistemas de salud.

Por tanto, en él se podrían incluir tecnología como mSalud, Internet de las Cosas para Sanidad o Big Data Sanitario.

mSalud

mSalud (o salud por dispositivos móviles) es un término empleado para designar el ejercicio de la medicina y la salud pública con apoyo de los dispositivos móviles tales como teléfonos inteligentes, dispositivos de monitorización de pacientes y otros dispositivos inalámbricos. Otra definición podría ser: aquella parte de la eHealth (salud electrónica/e-salud) que trata la aplicación de sistemas móviles en relación a la salud.

En paralelo a la evolución de IoT que comentaba al principio del capítulo, se han ido produciendo un incremento de ventas de teléfonos inteligentes que junto con la llegada de las redes 3G y 4G más la disponibilidad de sistemas de geolocalización han disparado el uso de aplicaciones móviles ofreciendo servicios para la salud en los últimos años.

El potencial de la mSalud puede ser empleado para un abanico grande de propósitos: promoción de la salud, prevención de la enfermedad, prestación de servicios de salud, capacitación y supervisión, pagos electrónicos asociados a los servicios de salud y como potenciadora de los sistemas de información en salud, entre otros.

Numerosas soluciones han salido al mercado en los últimos años a través de iniciativas privadas y públicas. Entre éstas y con lo que respecta a nuestra región cabe destacar la iniciativa mSSPA.

mSSPA

Es un proyecto orientado a la personalización e integración de servicios móviles de salud, en la que podremos encontrar lo que denominan un ecosistema corporativo de aplicaciones móviles. Los objetivos de la misma son:

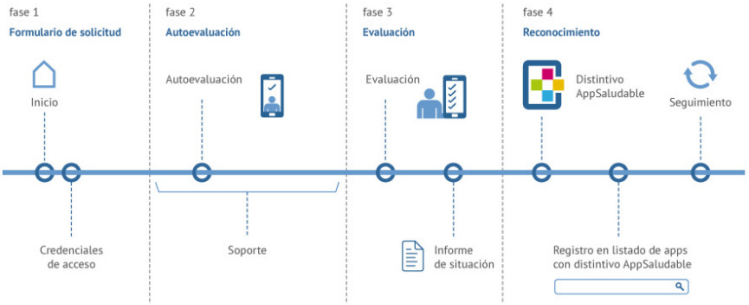
- Generación de un ecosistema de servicios móviles corporativos
- Personalización y Promoción de la Salud
- Generación de Impacto en la ciudadanía y en los profesionales.
- Testeo de modelos reales de uso y de negocio

Esta iniciativa se puso en marcha en la Junta de Andalucía dentro de la estrategia de calidad y seguridad en aplicaciones móviles en salud que incluye recomendaciones a desarrolladores, profesionales

sanitarios y ciudadanos. Desde 2013, a través de la Agencia de Calidad Sanitaria de Andalucía existe un distintivo "AppSaludable", un sello de garantía para reconocer aquellas aplicaciones móviles que sean fiables para los usuarios.

El proceso de obtención de este distintivo consiste principalmente en la autoevaluación de la aplicación de acuerdo a las recomendaciones de la guía existente y la posterior evaluación por parte de un comité de expertos de la Agencias, con el objeto de identificar posibles mejoras.

Una vez que se obtenga este distintivo, la aplicación formará parte de un repositorio de aplicaciones móviles en salud.



Fases del proceso para la obtención del distintivo AppSaludable[iv]

Dada la relación que tiene mSalud con iniciativas IoT, para nuestra organización mSSPA podría ser un buen punto de partida de cara a garantizar la calidad y seguridad.

IoHT[v] (Internet de las Cosas en Sanidad)

Internet de las cosas en Sanidad digamos que sería la convergencia e integración de forma inteligente de los datos recopilados por los sensores de los dispositivos médicos y de las tecnologías móviles que

iv Fuente: www.calidadappsalud.com

v Internet of Healthcare Things

se utilizan en la asistencia sanitaria. Muchos de estos dispositivos están y, por supuesto, estarán en nuestros centros de salud y hospitales facilitando la recogida de datos de nuestros pacientes, monitorización a distancia, almacenamiento en la "nube" (ya veremos cómo) para hacer posteriores análisis.

Sin embargo, cada vez más, el propio paciente quiere tener un mayor control de su salud, éste dispone de un mayor número de dispositivos y éstos de sensores que monitorizan el estado físico, que permiten comprender e interpretar éstos, puede autogestionar su enfermedad, quiere acceder a su historial clínico, se le puede realizar una atención personalizada, etc… es lo que se denomina empoderamiento del paciente.

Las pulseras, relojes inteligentes y dispositivos "wearables" en general nos están condicionando e incluso forzando a modificar nuestro día a día. El auge del bienestar y cuidado de la salud ("wellness" y lo "health friendly") lo ha convertido en un nuevo estilo de vida en el que cada vez hay más adeptos no solo para la práctica deportiva sino para llevar una alimentación saludable. Si a la vez estos datos estuviesen conectados "de alguna manera" con nuestros sistemas de información las posibilidades que nos daría el análisis a todos los niveles (clínico, social, administrativo, asistencial, hospitalario, investigación...) convertirían nuestros centros en más eficientes.

Ese auge comentado antes, ha llevado a que las propias empresas tomen iniciativas al respecto. Según una encuesta de la empresa "TrendMicro" entre grandes compañías europeas el 79% de aseguraba que sus trabajadores llevan cada vez más tecnología wearable al lugar de trabajo y el 77% decía fomentar activamente su uso. De hecho, el 19% confesaba haber puesto en marcha programas de wearables en la oficina, el 28% estar en plena implementación y el 34% interesadas en ella. Más de la mitad de los empresarios consultados consideraba que este impulso podría favorecer la productividad del personal.

Como ejemplos de soluciones podemos encontrar aquellas que ofrecen asistencia sanitaria a distancia. Los componentes que incluyen son diferentes sensores para pacientes como termómetro, pulsómetro, glucómetro o peso, que junto a la App instalada en el dispositivo móvil para el sanitario que le permite monitorizar, administrar y generar los informes.

Otras empresas especializadas utilizan el sistema de localización en tiempo real (RTLS[vi]) dentro del hospital que utilizando dispositivos como pulseras, diferentes tipos de tags, sensores de presencia y gateways permiten múltiples funcionalidades como:

- Identificación segura de pacientes y activos
- Localización y trazabilidad
- Paneles en tiempo real

Multitud de ideas, iniciativas algunas en mente, otras en proyecto y seguro que otras ya una realidad en algunos de nuestros centros. ¿Pero nos estamos dejando llevar por ver culminadas cuanto antes esas ideas o proyectos? ¿Estamos teniendo en cuenta factores como la interoperabilidad, seguridad y la privacidad de datos?

Big Data sanitario

Aunque como ya hemos comentado que el objetivo de este libro es tratar la Seguridad IoT y no al análisis de la información en sí, no podemos dejar pasar la oportunidad de comentar someramente el Big Data en sanidad.

Llamamos "Big Data" a la colección de datos muy grandes, estructurados o no, estáticos o dinámicos, simples o complejos, que pueden ser capturados, almacenados, gestionados y analizados tanto de forma convencional como utilizando técnicas más innovadoras.

[vi] *Real-time locating system*

Herramientas de análisis vinculadas a Inteligencia de Negocio (Business Intelligence) tradicionales permiten la captura de información disponible en fuentes de datos estructurados. Por tanto hay que ir un paso (o dos) más allá, y extraer el "conocimiento" que genera el procesamiento de la información NO estructurada: lenguaje natural, redes sociales, wearables, sensores… Gracias a la Inteligencia Artificial se podrán identificar datos relevantes que apoyen a los profesionales en sus hipótesis, así como ayudarles a aportar nuevas ideas y encontrar las mejores opciones para cada paciente.

Hay que ser conscientes de que la cantidad de información existente relacionada con nuestra salud es abrumadora. Los centros de salud, los hospitales, la administración pública, la sanidad privada e incluso nosotros mismos como pacientes acumulamos grandes cantidades de datos en formatos muy diversos: informes en papel, archivos de Office, imágenes, videos, recetas, tarjeta sanitaria, etc.

Además, hasta que se instauró la Historia Clínica Electrónica cada miembro de la comunidad sanitaria tenía una visión parcial del paciente, lo que dificultaba el diagnóstico y tratamiento. Aunque hoy en día el historial médico de un paciente es compartido dentro del sistema sanitario público autonómico, la situación sigue siendo complicada cuando sales fuera de tu autonomía, o gestionas tu salud en el sector privado.

Por ello, hay que recalcar que es admirable que, a pesar de tener la información poco organizada e integrada, nuestros profesionales de la salud consiguen resultados excelentes en la práctica clínica.

Una práctica clínica que ya se ha visto alterada por los avances tanto en mSalud como con Internet de las Cosas, lo que ha provocado que se esté pasando de una medicina reactiva (actuar si hay enfermedad) a la medicina de las 4Ps:

- Predictiva
- Preventiva

- Personalizada
- Participativa

¿Alguien puede imaginarse cómo serían esos resultados si todos los datos asociados a un paciente se pudieran agrupar, consolidar y analizar en un lugar único y accesible? Los profesionales podrían tomar mejores decisiones. ¿Y cómo se podría mejorar la práctica clínica en general, no sólo para un paciente, sino para toda una patología, si se pudieran agrupar, normalizar, y analizar los resultados de todos los pacientes de forma anonimizada?

El caso es que según la Organización Mundial de la Salud (OMS) únicamente el 17% de los países se sirven de Big Data tienen programas regulatorios para gestionar su uso en el ámbito sanitario. En este contexto y a pesar de contar con una institución como el Ministerio de Agenda Digital, España no se encuentra en la lista de regiones más punteras. Así que todavía queda un largo camino por recorrer.

En mi opinión un claro ejemplo de beneficio del uso correcto de Big Data en la práctica clínica sería la incorporación de sistemas de apoyo a la decisión clínica, por ello me he decidido a dar unas breves pinceladas de su aplicación.

Sistema de apoyo a la decisión clínica (CDS[vii])

En cualquier sistema de registro electrónico de pacientes, podemos acceder a la información cuándo y dónde sea necesaria. Esto ya ha supuesto un gran avance para muchas organizaciones sanitarias, sin embargo, sería importante poder dar un paso más en la evolución de estos sistemas. Habilitar herramientas, no solo que dispongan de indicadores y cuadros de mando sino que incluyan capacidades como alertas ante valores críticos en los resultados de un laboratorio, detección de potenciales errores en la medicación prescrita,

[vii] Clinical Decision Support

protocolos de actuación guiados… para ello se podrían utilizar sistemas CDS (Sistema de apoyo la Decisión Clínica).

Como es lógico estos sistemas lo que pretende es acercar a los equipos clínicos una información oportuna y cualificada que, en ningún momento, sustituirá el juicio clínico, sino que asistirá al personal sanitario a la hora de tomar unas decisiones de mayor calidad.

De hecho esta tendencia a la medicina centrada en el paciente junto a que el sector de IoT en Salud es ya la segunda actividad asociada a dispositivos interconectados hace que se empiece a denominar a IoHT como Internet del Paciente (IoP).

Computación en la nube

El "cloud computing" o computación en la nube es una forma de prestación de los servicios de tratamiento de la información, válida tanto para una empresa como para un particular y, también cómo no, para la Administración Pública.

En un entorno de "cloud computing" la gestión de la información está de forma virtual en manos del cliente que contrata los servicios de la nube, que la trata a través de Internet accediendo a soluciones de bases de datos, correo electrónico, nóminas o gestión de recursos humanos de acuerdo a sus necesidades.

Ya no se trata del clásico "hosting" como ocurría hace unos años o de datacenters donde las empresas alojaban (bueno y alojan aún) sus servidores físicos para despreocuparse de cortes de luz, refrigeración, etc. Esa tecnología inicial en la nube fue evolucionando al comenzar a utilizar otras de basadas en la virtualización. Con ellas se ha conseguido mejorar los automatismos y las herramientas de gestión de las máquinas. Es lo que se denomina, modelo IaaS (Infraestructura as a Service) el cual permite gestionar la infraestructura como un servicio de forma remota pero el cliente se desentiende de la propia virtualización, discos, red… ya que la mantiene el proveedor.

A su vez, el desarrollo de metodologías de desarrollo ágiles como Scrum o Programación extrema (XP) han promovido la aparición de tendencias o movimientos del tipo DevOps en el que se promulga el que el desarrollo de software se haga orientado a operaciones por lo que ambos perfiles deben trabajar en estrecha colaboración facilitando la puesta en producción del software. Esto ha hecho que IaaS evolucione a PaaS (Platform as a Service), plataforma de desarrollo de soluciones directamente en Cloud mediante sus servidores de aplicaciones. Los desarrolladores se centran en programar y el PaaS les facilita las tareas de gestión y configuración: servidores, virtualización, sistemas operativos, discos, red, etc. Esto hace que el desarrollo, el testing y los despliegues sean más rápidos, más simples y más eficientes. Las aplicaciones desarrolladas en el PaaS heredan todas las ventajas de Cloud: escalabilidad, reducción de costes, etc.

Sin duda, el uso de Cloud Computing es uno de los puntos con mayor controversia dentro de organizaciones como la nuestra, donde manejamos principalmente datos clínicos de pacientes. Sin embargo, la proliferación de aplicaciones móviles, nuevos dispositivos, sensores y demás elementos vinculados a Internet de las Cosas, obligan a no mirar para otro lado y afrontar el reto y buscar una estrategia que permita sacar el mayor rendimiento a esta tecnología por supuesto estableciendo las medidas adecuadas en relación a la seguridad y privacidad de los datos.

Está claro que una solución en la nube pública puede que no sea la mejor opción por los riesgos que habría que asumir y que una nube privada, aunque permite un mayor control, no sea aquélla a la que podamos sacar el máximo provecho al ecosistema tan variado de soluciones hardware y software existentes en el mercado. Así que en mi opinión una nube híbrida podría ser la alternativa más plausible, aprovechando las ventajas tanto de la nube pública como de la privada.

¿Hasta qué punto estamos dispuestos a compartir datos? ¿Podemos compartirlos?

APROXIMACIÓN A LA SEGURIDAD IoT

Aunque en los siguientes capítulos se aborda en profundidad la seguridad, voy a realizar una primera aproximación a este tema dando algunos datos y mostrando algunas actuaciones llevadas a cabo por los principales agentes a nivel tanto europeo como español y andaluz.

Probablemente no sorprenda ver que la seguridad es una de las principales preocupaciones. De hecho desde que empezó a arraigar IoT en las empresas tener una brecha de seguridad ha sido la principal preocupación de las mismas, si a ésta le sumamos el aspecto de "privacidad de datos" supone que un 33% de encuestados opinan que es la principal barrera a superar.

Pero y si lo vemos desde otra perspectiva, un 66% de encuestados opinan que NO es la principal preocupación por lo que anteponen otros criterios tales como resistencia al cambio o habilidades insuficientes con esta tecnología. Pero de aquellos con programas/soluciones IOT más grandes –al menos 10.000 dispositivos conectados- **sólo el 7%** dice que la seguridad es su principal preocupación. En comparación con un 19% para aquellos con proyectos más pequeños. Esto sugiere que, a pesar de tener este tipo de soluciones y ganas de aplicar seguridad en las mismas, no todos tienen todavía la experiencia y los recursos para aplicarlo.

Las empresas necesitan conectividad segura, fiable y global. Las principales preocupaciones de cara a desplegar proyectos basados en IoT son la seguridad (75% de las empresas que han implementado IoT) y la cobertura de red (74%).

Según un informe de Gartner habrá que trabajar mucho en factores clave como son las integraciones de sistemas y garantizar la seguridad.

Según este informe las empresas construirán y adaptarán sus implementaciones IoT para incluir una combinación de los cinco componentes de arquitectura claves:

- Las Cosas: Pueden ser simples o inteligentes por sí mismas y almacenar la mayoría de sus datos en ellos. Las Cosas pueden también ser autosuficientes y comunicar con Internet solo para centralización y análisis.

- Gateways: podrían alojar la lógica de aplicación, almacenar datos y comunicar con el Internet de las Cosas que está conectado a él. Las Cosas no tienen porqué ser inteligentes ya que el gateway puede proporcionar estos recursos.

- Dispositivos móviles: los teléfonos inteligentes (o cualquier dispositivo móvil) puede alojar la lógica de la aplicación, almacenar datos y comunicar con el Internet de las Cosas que están conectados a ellos. Las Cosas no tienen porque ser inteligentes ya que el dispositivo móvil puede proporcionar estos recursos.

- La nube: puede actuar como un hub de conexión central, potenciar el análisis de datos y proveer almacenamiento de datos. Las Cosas no tienen porqué ser inteligentes ya que la nube puede proporcionar estos recursos.

- La compañía: Este rol en la arquitectura está enfocado en mantener conectadas las máquinas, la lógica de la aplicación y el almacenamiento de datos dentro de sus instalaciones, es decir, dentro de la seguridad perimetral corporativa.

Cada arquitectura IoT incluirá más de uno de estos 5 componentes funcionales. Las direcciones de las compañías deben tener en cuenta la seguridad, privacidad, costo, accesibilidad, agilidad y ejecución para poder determinar la mejor arquitectura aplicable a su organización.

De hecho ya se habla del perfil “Arquitecto IoT” como pieza angular en la planificación, ejecución y gestión IoT.

- Debe participar y colaborar con la dirección y/o responsables de proyecto para establecer su visión IoT.

- Diseñar la arquitectura IoT para el proyecto.
- Establecer los procesos adecuados para llevar a buen término el proyecto.
- Trabajar en equipo con responsables de Seguridad, Sistemas...

Europa

Haciendo una primera aproximación a la seguridad en IoT, y siguiendo con la exposición de algunos datos estadísticos, indicar que en Europa un 70% de las organizaciones que han incluido IoT considera que tienen los conocimientos adecuados sobre seguridad IoT y que tienen los procedimientos adecuados para gestionarla. Por contra, algo más de un 40% de éstas ha contratado especialistas en seguridad IoT o ha formado a su personal TI en esta materia. Pero a destacar que solo un 34% revisa posibles vulnerabilidades después de entrar en producción. Creo que son datos significativos sobre el nivel de madurez en materia de seguridad de las empresas europeas.

La comisión europea, a través de diferentes informes realizados en los últimos 4 años, asociados a la estrategia sobre Mercado Único Digital (DSM) habla directamente de que IoT representa la mayor innovación socio-económica en la actualidad. Aunque creo que a estas alturas nos ha quedado claro, ¿no?

Ésta, subraya la necesidad de evitar una posible fragmentación de esta tecnología fomentando la interoperabilidad con el objetivo de aprovechar al máximo todo su potencial. Los pilares en los que debe basarse esta estrategia son:

- Los dispositivos IoT y los servicios empleados deberían ser capaces de conectarse de forma única en cualquier punto de la Unión Europea.
- Creación de plataformas abiertas para facilitar la innovación entre la comunidad de desarrolladores.

- IoT debe estar centrado en el ciudadano. Para ello se hace hincapié en potenciar la seguridad y la protección de datos personales, visible a través de la etiqueta "Trusted IoT" a la que las empresas deben acogerse dentro del marco europeo.

Entretanto, en el año 2015 la Comisión Europea junto a los principales actores de la industria IoT lanzaron la Alianza para la innovación en Internet de las Cosas[viii]. Su objetivo no es otro que crear un ecosistema dinámico IoT a nivel europeo. Esta alianza ofrece la oportunidad de discutir los obstáculos legales y regulatorios, forjando el consenso en materia de estandarización a través del Instituto Europeo de Normas de Telecomunicaciones[ix] y oneM2M[x].

Centrándonos en el tema de la Seguridad, la Unión Europea dispone una agencia de seguridad llamada ENISA (Agencia Europea de Seguridad de las Redes y de la Información) la cual publica anualmente un informe completo de las principales amenazas y tendencias en ciberseguridad a la cual nos enfrentamos diariamente. Como podéis ver en la siguiente tabla, todas son familiares en nuestro ámbito y las principales además muy relacionadas con Internet de las Cosas pero lo que sí hay que tener muy en cuenta es el factor multiplicador del riesgo asumido cuando afrontemos un proyecto IoT.

Por cierto, os dejo el enlace[xi] de una aplicación web lanzada por esta agencia que contiene información sobre las 15 principales amenazas cibernéticas encontradas en 2017, indicadas en la tabla anterior.

viii Alliance for Internet of Things Innovation - AIOTI

ix European Telecommunications Standards Institute

x Organización global para definición de estándares IoT

xi https://etl.enisa.europa.eu/#/

Top Threats 2016	Assessed Trends 2016	Top Threats 2017	Assessed Trends 2017	Change in ranking
1. Malware	↑	1. Malware		→
2. Web based attacks	↑	2. Web based attacks		→
3. Web application attacks	↑	3. Web application attacks		→
4. Denial of service	↑	4. Phishing		↑
5. Botnets	↑	5. Spam		↑
6. Phishing	⇒	6. Denial of service		↓
7. Spam	↓	7. Ransomware		↑
8. Ransomware	⇒	8. Botnets		↓
9. Insider threat	⇒	9. Insider threat		→
10. Physical manipulation/damage/ theft/loss	↑	10. Physical manipulation/damage/ theft/loss		→
11. Exploit kits	↑	11. Data breaches		↑
12. Data breaches	↑	12. Identity theft		↑
13. Identity theft	↓	13. Information leakage		↑
14. Information leakage	↑	14. Exploit kits		↓
15. Cyber espionage	↓	15. Cyber espionage		→

Legend: Trends: ↓ Declining, ⇒ Stable, ↑ Increasing
Ranking: ↑ Going up, → Same, ↓ Going down

Fuente: ENISA Threat Landscape Report 2017

Las principales conclusiones a las que llega la agencia en este informe son que debemos:

- Comprender las tendencias emergentes en cuanto a infraestructura maliciosa, tácticas de ataque, malware… y adaptar nuestros métodos de defensa.
- Desarrollar nuevos controles de seguridad y adoptar tecnologías emergentes para garantizarla.
- Desarrollar nuevos modelos de madurez para la CTI.
- Desarrollar programas educativos para cubrir los perfiles y capacidades necesarias en materia de Inteligencia contra amenazas cibernéticas (CTI[xii])

xii Cyber Threat Intelligence

España

En el año 2013 se definió la Estrategia de Ciberseguridad Nacional bajo el amparo del Departamento de Seguridad Nacional. Éste es el documento estratégico que sirve de fundamento para desarrollar las previsiones de la Estrategia de Seguridad Nacional en materia de protección del ciberespacio con el fin de implantar de forma coherente y estructurada acciones de prevención, defensa, detección y respuesta frente a las ciberamenazas. Dentro de él, además de definir los objetivos y estructura orgánica al servicio de la ciberseguridad, se incluyen las diferentes líneas de acción orientadas a la consecución de objetivos.

LÍNEA DE ACCIÓN		CONTENIDO
1	**Capacidad de prevención, detección, respuesta y recuperación ante las ciberamenazas**	Incrementar las capacidades de prevención, defensa, detección, análisis, respuesta, recuperación y coordinación ante las ciberamenazas, haciendo énfasis en las Administraciones Públicas, las Infraestructuras Críticas, las capacidades militares y de Defensa y otros sistemas de interés nacional.
2	**Seguridad de los Sistemas de Información y Telecomunicaciones que soportan las Administraciones Públicas**	Garantizar la implantación del Esquema Nacional de Seguridad, reforzar las capacidades de detección y mejorar la defensa de los sistemas clasificados.
3	**Seguridad de los Sistemas de Información y Telecomunicaciones que soportan las Infraestructuras Críticas**	Impulsar la implantación de la normativa sobre Protección de Infraestructuras Críticas y de las capacidades necesarias para la protección de los servicios esenciales.
4	**Capacidad de investigación y persecución del ciberterrorismo y la ciberdelincuencia**	Potenciar las capacidades para detectar, investigar y perseguir las actividades terroristas y delictivas en el ciberespacio, sobre la base de un marco jurídico y operativo eficaz.
5	**Seguridad y resiliencia de las TIC del sector privado**	Impulsar la seguridad y la resiliencia de las infraestructuras, redes, productos y servicios empleando instrumentos de cooperación público-privada.
6	**Conocimientos, Competencias e I+D+i**	Promover la capacitación de profesionales, impulsar el desarrollo industrial y reforzar el sistema de I+D+i en materia de ciberseguridad.
7	**Cultura de ciberseguridad**	Concienciar a los ciudadanos, profesionales y empresas de la importancia de la ciberseguridad y del uso responsable de las nuevas tecnologías y de los servicios de la Sociedad de la Información.
8	**Compromiso internacional**	Promover un ciberespacio internacional seguro y confiable, en apoyo a los intereses nacionales.

Fuente: Departamento de Seguridad Nacional

Por otro lado, el INCIBE (Instituto Nacional de Ciberseguridad) dependiente del Ministerio de Energía, Turismo y Agenda Digital, se encarga de la protección del ciberespacio utilizado por pymes y ciudadanos. A través de la investigación, la prestación de servicios y la coordinación con los agentes competentes, busca aumentar la capacidad de detección de nuevas amenazas, la respuesta y análisis

de incidentes, la resiliencia de las organizaciones y diseñar medidas preventivas que atiendan las necesidades de la sociedad y de las infraestructuras críticas.

El Equipo de Respuesta ante Emergencias Informáticas de Seguridad e Industria (CERTSI), perteneciente al Ministerio de Energía, Turismo y Agenda Digital y del Ministerio del Interior, ya ha publicado varias guías orientadas a la seguridad para IoT, como:

- Iniciativas y mejores prácticas de seguridad para el IoT. En esta guía se incluyen diferenciadas por categorías.
    - Diseño e Instalación
    - Operación y Mantenimiento
    - Protección de la Privacidad
- Ciberseguridad en las Comunicaciones Inalámbricas en Entornos Industriales
- Riesgos y retos de ciberseguridad y privacidad en IoT

El Centro Criptológico Nacional (CCN) es el organismo responsable de coordinar la acción de los diferentes organismos de la Administración que utilicen medios o procedimientos de cifrado, garantizar la seguridad de las Tecnologías de la Información en ese ámbito, informar sobre la adquisición coordinada del material criptológico y formar al personal de la Administración especialista en este campo.

Además de establecer el Esquema Nacional de Seguridad, hace pocos meses publicó un informe sobre Buenas Prácticas relacionadas con Internet de las Cosas el cual recomiendo su lectura[xiii].

[xiii] https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/2258-ccn-cert-bp-05-16-internet-de-las-cosas/file.html

Al final del mismo se incluye un decálogo de recomendaciones que os muestro a continuación.

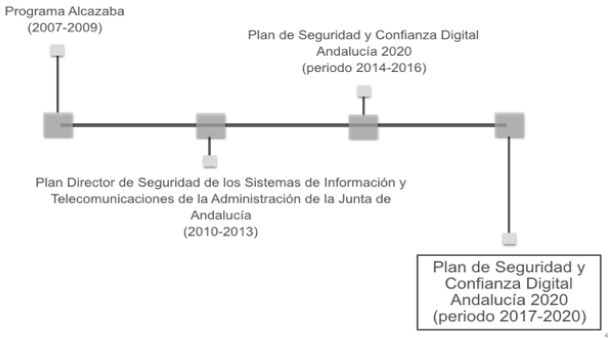
	Decálogo de Seguridad para Instalaciones IoT
1	Evitar utilizar dispositivos IoT siempre que no sean estrictamente necesarios.
2	No utilizar, en la medida de lo posible, aquellos dispositivos IoT que transmiten información a servidores externos (la Nube), incluso si son los del fabricante.
3	Cambiar las contraseñas por defecto de los dispositivos y utilizar contraseñas realmente robustas, que no estén en ningún diccionario, que sean suficientemente largas y por tanto difíciles de adivinar.
4	Mantener actualizados los dispositivos con las últimas versiones disponibles de software y firmware.
5	Desactivar toda conectividad remota (con Internet) de los dispositivos cuando no sea estrictamente necesaria.
6	Mantener abiertos solo aquellos puertos de comunicación que sean realmente necesarios y modificar los puertos de escucha si es posible.
7	Si los dispositivos IoT no permiten la configuración de su seguridad, operar con ellos siempre en una red de área local (LAN) detrás de un dispositivo (enrutador) correctamente configurado que sí provea esa seguridad.
8	En la medida de lo posible, asegurar la autenticidad, confidencialidad e integridad en todas las comunicaciones locales (LAN), especialmente si estas se realizan por enlaces radio (Wi-Fi, Bluetooth, etc.).
9	Comprobar periódicamente y sin previo aviso, la configuración de seguridad de todos los elementos de la arquitectura IoT y de sus dispositivos de comunicación con el exterior.
10	Comprobar la visibilidad de los dispositivos propios en buscadores de dispositivos IoT como Shodan.

Fuente: Centro Criptológico Nacional

Andalucía

En esta época de Transformación Digital, como comentábamos antes, Andalucía no puede quedarse. Por ello, existen numerosas iniciativas para potenciar el desarrollo de nuevas estrategias de negocio que beneficien tanto al tejido empresarial como a las administraciones públicas y sus ciudadanos.

Consciente de la importancia de la seguridad de la información en el modelo de relación con los ciudadanos, la Junta de Andalucía viene desarrollando desde hace varios años distintas políticas de seguridad que se concretan en los siguientes documentos:



Evolución políticas de seguridad Junta de Andalucía[xiv]

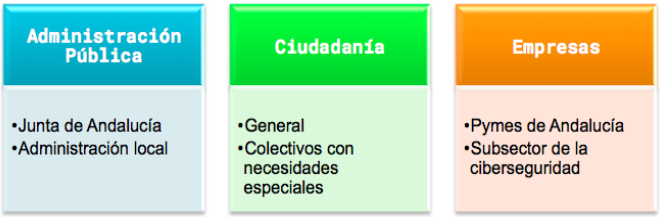
El Plan de Seguridad y Confianza Digital de Andalucía 2020 establece una serie de objetivos centrados en ampliar la cultura de confianza y seguridad digital, impulsar el mercado de la ciberseguridad, potenciar la adopción de buenas prácticas en ciberseguridad corporativa, y reforzar las capacidades de prevención, detección y respuesta a incidentes. Se vertebra en **5 líneas de actuación**:

1. Coordinación de la seguridad TIC en la Administración autonómica, potenciando la adopción de buenas prácticas y ofreciendo servicios de asesoramiento, apoyo y coordinación
2. Formación y concienciación de los trabajadores del sector público, la ciudadanía y las empresas, y extensión de la cultura de confianza y seguridad digital
3. Impulso de la industria de la seguridad digital, mediante el estímulo de la oferta y la demanda de productos, servicios y profesionales de la seguridad digital, y promoción de la adopción de buenas prácticas y de la cultura de seguridad en el tejido empresarial andaluz

xiv Fuente: Premios Asociación @asLAN

4. Coordinación con otras Administraciones y Organismos Públicos en materia de seguridad TIC
5. Protección frente a ciberamenazas, mediante la mejora de las capacidades de prevención, detección y respuesta a incidentes de seguridad en Andalucía (a través del centro AndalucíaCERT)

Dentro de este plan nace la iniciativa Seguridad Digital de Andalucía (SEDIAN). Esta es una iniciativa integral en ciberseguridad que quiere dar respuesta a problemas reales, implicando tanto a las diferentes administraciones, como a empresas y a ciudadanía. Implicar a la sociedad andaluza en general va a permitir dar un paso adelante en la puesta en marcha nuevos proyectos tecnológicos, entre ellos IoT, donde la interacción con el ciudadano va a ser imprescindible.



Colectivos implicados en SEDIAN[xv]

Entre los objetivos de SEDIAN se contemplan los siguientes:

- Generar confianza en el uso de las TIC para impulsar el proceso de transformación digital en todos los ámbitos de la sociedad andaluza con garantías y conocimientos sobre la seguridad digital. Para ello se recurre, entre otras cosas, a programas de sensibilización, asistencia y formación, con especial atención a los menores.

xv Fuente: Premios Asociación @asLAN

- Potenciar la adopción de buenas prácticas en materia de seguridad digital en la administración autonómica y local de Andalucía.
- Impulsar el mercado de la seguridad digital y la creación de empleo, mediante el estímulo de la oferta y la demanda de productos, servicios y profesionales de la seguridad digital.
- Reforzar las capacidades de prevención, detección y respuesta a incidentes de seguridad en Andalucía, a través del centro de seguridad TIC, AndalucíaCERT.

Hablando del AndalucíaCERT ha publicado algunos informes de divulgación para introducir a los usuarios (tanto a personal de la Junta de Andalucía como al público en general) en Internet de las Cosas, así como dar unas nociones básicas sobre seguridad. Entre estas nociones ya se atisban algunos puntos a considerar como estrategia de seguridad en IoT aplicable a cualquier proyecto a desarrollar.

- Incorporar la seguridad en la fase de diseño
- Actualizaciones de seguridad avanzadas y gestión de vulnerabilidades
- Basarse en prácticas de seguridad probadas
- Priorizar medidas de seguridad de acuerdo con el impacto potencial
- Promover la transparencia en todo el IoT
- Conectar de manera cuidadosa y deliberada. En este caso hace mención a una serie de recomendaciones para los propios consumidores para reducir posibles riesgos.

En los siguientes capítulos se aborda con mayor profundidad la seguridad de la información. Inicialmente, tratando Normativa, Buenas Prácticas y Gobernanza para después continuar con una parte

más práctica en la que se abordan diferentes superficies de ataque habituales en proyectos vinculados a Internet de las Cosas.

Referencias Bibliográficas

2. Barómetro Vodafone 2017/2018 http://www.vodafone.com/business/iot
3. Leading the IoT-Gartner https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf
4. Comunicación A Digital Single Market Strategy for Europe COM (2015) 192 Final. Available at: http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1447773803386&uri=CELEX:52015DC0192
5. ENISA Threat Landscape Report 2017 https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2017
6. Advancing the Internet of Things in Europe http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52016SC0110
7. IoT in Healthcare is really the Internet of Patients (IoP) https://www.intersystems.com/pulse-blog/iot-healthcare-really-internet-patients-iop/
8. Smart Cities - La transformación digital de las ciudades https://iot.telefonica.com/libroblanco-smart-cities/media/libro-blanco-smart-cities-esp-2015.pdf
9. Estrategia de calidad y seguridad en aplicaciones móviles de salud http://www.calidadappsalud.com/

10. IEBS-Escuela de Negocios
https://www.iebschool.com

11. Estrategia Nacional de Ciberseguridad Nacional
https://www.ccn-cert.cni.es/publico/dmpublidocuments/EstrategiaNacionalCiberseguridad.pdf

12. SEDIAN. Seguridad Digital de Andalucía
http://www.seguridad.andaluciaesdigital.es/

13. Visión del ecosistema inversor startup de España 2017
https://startupxplore.com/es/blog/informe-vision-del-ecosistema-inversor-startup-de-espana-2017/

14. Plan de Seguridad y Confianza Digital 2017-2020
http://www.juntadeandalucia.es/export/drupaljda/Plan_Seguridad_Confianza_Digital_2017-2020.pdf

Gobierno y Normativa legal en entornos sanitarios

Josep Bardallo

Introducción

Tal como se ha ido comentando en los diferentes capítulos de este libro el ioT en el entorno sanitario español es una realidad en la actualidad, y se encuentra más extendido de lo que en principio pueda parecer (El 60% de las organizaciones sanitarias de todo el mundo han introducido dispositivos IoT en sus instalaciones, convirtiéndose así en el tercer sector más avanzado en la implementación de IoT). Existen incluso iniciativas en Andalucía, como el reciente convenio [(1)] (publicado en el BOE del 28 de Marzo), entre el Servicio Andaluz de Salud y la Entidad Pública Empresarial Red.es donde una de las líneas de trabajo se centra en la "*Aplicación de tecnologías emergentes («Big Data», «Analytics», «Machine Learning», «IoT», etc.) al análisis de fuentes de información (clínicas y no clínicas), para la adopción y potencial extensión de programas de gestión de la salud de la población que permitan mejorar la calidad de vida de los colectivos de pacientes crónicos*".

Desde otros capítulos se han abordado la vertiente más técnica de la implementación de ioT en Sanidad, en este capítulo nos vamos a centrar más en la parte de cumplimiento normativo y gobernanza en este tipo de entornos haciendo un repaso (breve) a las normativas existentes en la actualidad y algunos marcos de trabajo útiles y recomendaciones a tener en cuenta si queremos desplegar dispositivos ioT de manera segura y cumpliendo las normativas existentes, centrándonos en el entorno que nos ocupa: entorno sanitario en Andalucía, España y la Comunidad Europea.

No se pretende realizar un detallado análisis de las normativas o mejores prácticas de seguridad que afectan a tecnologías ioT en entorno sanitario (la mayoría de las normativas en ioT son normativas genéricas, ya que estamos en la fase incipiente de desarrollos específicos para esta área), sino mostrar una guía de estas y unos consejos de los elementos a tener en cuenta para tener una implementación de dispositivos ioT con unas ciertas garantías. En los próximos años se espera un avance significativo de normativas y mejores prácticas específicas en entornos sanitarios, por lo que habrá

que estar atentos a su evolución. (en Estados Unidos por ejemplo el NIST ya ha empezado a desarrollar guías especificas [2] [3] [4], y la FDA (U.S. Food & Drugs Administration) ha publicado diversos estudios donde indica recomendaciones para el tratamiento de los dispositivos por parte de los fabricantes [5] [6] [7]. Incluso el NIST está preparando un interesante documento [2], actualmente en draft, donde se analiza el estado actual de los estándares internaciones en ciberseguridad desarrollados para IoT en los diferentes sectores (incluyendo salud y dispositivos médicos conectados).

Principales riesgos con la adopción de ioT en entornos sanitarios

En un entorno sanitario, al introducir dispositivos ioT estamos introduciendo nuevos elementos que incrementan los riesgos existentes, no solo en el factor de cumplimiento de normativas legales referentes a la privacidad de los datos (que en el entorno ioT se encuentran distribuidos), sino también en la ampliación de la superficie de "ataque" y el nivel de riesgo, ya que los dispositivos ioT no suelen están diseñados inicialmente con "seguridad por defecto", y suelen ser difíciles de actualizar o mejorar su seguridad.

Además, la información gestionada en entornos sanitarios está clasificada en su mayoría como información "personal" (PII o "Personally Identifiable Information") y como información de salud (PHI o "protected Health Information"), requiriendo la mayoría de las normativas de privacidad el máximo nivel de protección para este tipo de información.

En 2017, se demostró que los dispositivos cardiacos implantados por un importante hospital en Nueva York [8], presentaban una vulnerabilidad mediante la cual alguien remotamente podría agotar la batería de los dispositivos, modificar el ritmo de funcionamiento, o incluso administrar descargas, lo que indica que no solo tenemos amenazas a los datos, sino que aparecen nuevas amenazas referidas a la vida de las personas.

Por estas razones, en el entorno sanitario es todavía más importante (si cabe), implementar seguridad basada en gestión del riesgo y con un buen gobierno de la seguridad, teniendo en cuenta los tres pilares básicos de la seguridad además del cumplimiento normativo y la privacidad:

- **Confidencialidad**: Evaluar y gestionar los riesgos a un nivel aceptable para que información sensible (datos de pacientes, etc.) solo sea accesible por quien debe serlo. En el ecosistema ioT es un poco más complejo, ya que los dispositivos que manejan información se encuentran distribuidos y gestionados en entornos que habitualmente no están controlados por los departamentos de IT o Seguridad (wereables, dispositivos de Telemedicina en casa del paciente, dispositivos de diagnóstico por la imagen, etc.)

- **Integridad**: Asegurarse que la información que se gestiona no ha sido modificada y es correcta. En el entorno sanitario es uno de los pilares básicos, ya que en base a esta información se toman decisiones que pueden afectar a la vida de las personas, por ejemplo, o se pueden producir usos inadecuados de dispositivos médicos con el consiguiente riesgo para los pacientes.

- **Disponibilidad**: Es importante que la información esté disponible cuando es necesaria y a las partes autorizadas, así como que los dispositivos médicos estén operativos cuando son necesarios, y funcionen como están diseñados.

- **"Compliance" o Cumplimiento normativo:** Existen multitud de leyes y normativas de obligado cumplimiento en la gestión de datos sanitarios (datos sensibles) en todo el mundo, y concretamente en Europa y España, principalmente el GDPR [(9)] (que veremos más adelante) o en EEUU el HIPAA [(10)].

Mejores prácticas

Actualmente, no se dispone en el mercado español de mecanismos que garanticen a los usuarios que se han tenido en cuenta los riesgos de seguridad en el desarrollo del sistema IoT y que éstos han sido minimizados, aunque existen algunas iniciativas en este sentido:

- **"Buenas prácticas y medidas de confianza en dispositivos ioT"** [(11)]**:** Manual de buenas prácticas desarrollado por el Centro de Estudios de la Movilidad y el IoT (CEM) del ISMS Forum Spain, donde se especifican una serie de recomendaciones y controles divididos en 6 dominios, y del que se recomienda su lectura:

    - Seguridad en el diseño.
    - Gobierno y seguridad en el ciclo de vida comercial.
    - Protección del Hardware/Firmware.
    - Seguridad en las comunicaciones.
    - Seguridad en sistemas.
    - Seguridad jurídica.

    Además, se ha definido el Reglamento de uso de Marca de Garantía que define la normativa que deben cumplir los fabricantes para la obtención del Sello de Confianza IoT en sus dispositivos, una iniciativa que pretende que los fabricantes de dispositivos IoT de manera voluntaria demuestren su grado de "seguridad".

- **Buenas prácticas en Internet de las cosas (ioT)** [(12)]**:** El CCN-CERT es el organismo público español encargado de la gestión de ciber-incidentes que afecten a sistemas del Sector Público, a empresas y organizaciones de interés estratégico para España. Entre los diferentes documentos y guías que edita, en junio del 2017 emitió este de buenas prácticas, donde se recogen el estado y tendencias en IoT y recomendaciones para su uso e implementación.

- El grupo de trabajo (Grupo 29) de las autoridades europeas de protección de datos, elaboró en 2014 un documento [(13)] con una serie de recomendaciones en materia de seguridad, privacidad y protección de datos que sería necesario aplicar a los dispositivos IoT, donde se hacen recomendaciones sobre la realización de evaluaciones de Impacto sobre la privacidad (EIP), privacidad por diseño, recopilación de datos o gestión del consentimiento.

A nivel internacional, como hemos visto se están desarrollando algunos documentos de mejores prácticas y estándares para ioT en sanidad:

- Estándares ISO [(14)] creados por el comité técnico 215 dedicado a informática de la salud. En IoT podemos destacar:

  - ☐ ISO/TR 17522:2015 ("Provisions for health applications on mobile/smart devices")
  - ☐ IEC 80001-1:2010 ("Application of risk management for IT-networks incorporating medical devices - - Part 1: Roles, responsibilities and activities ")
  - ☐ IEC/TR 80001-2-1:2012 ("Application of risk management for IT-networks incorporating medical devices -- Part 2-1: Step by Step Risk Management of Medical IT-Networks; Practical Applications and Examples")
  - ☐ IEC/TR 80001-2-2:2012 ("Application of risk management for IT-networks incorporating medical devices -- Part 2-2: Guidance for the communication of medical device security needs, risks and controls")

- ☐ IEC/TR 80001-2-3:2012 (“Application of risk management for IT-networks incorporating medical devices - Part 2-3: Guidance for wireless networks”)
- ☐ IEC/TR 80001-2-5:2014 (“Application of risk management for IT-networks incorporating medical devices -- Part 2-5: Application guidance -- Guidance for distributed alarm systems”)
- ☐ ISO/IEEE 11073-10419:2016 (“Personal health device communication -- Part 10419: Device specialization --Insulin pump”)
- ☐ ISO/TR 21730:2007 (“Use of mobile wireless communication and computing technology in healthcare facilities -- Recommendations for electromagnetic compatibility (management of unintentional electromagnetic interference) with medical devices”)
- ☐ ISO 27799:2016 (“Information security management in health using ISO/IEC 27002”)
- ☐ ISO/IEC 27030 — Information technology — Security techniques — Guidelines for security and privacy in Internet of Things (IoT)

- Recomendaciones de ENISA ("European Union Agency for Network and Information Security"): "Cyber security and resilience for Smart Hospitals" [15]

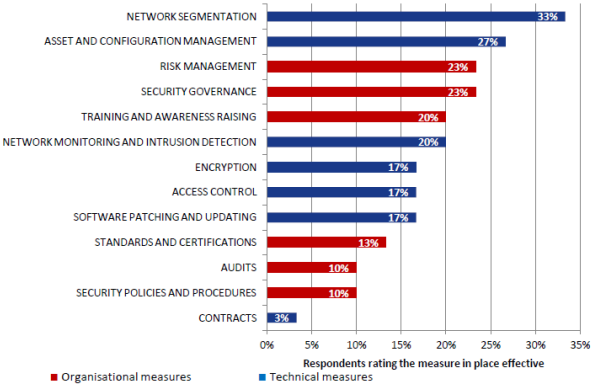


Figure 15 Effective measures in place

- Recomendaciones de NIST como el "Cybersecurity for IoT Program" [4] y la "Cybersecurity Practice Guide SP 1800-8, Wireless Infusion Pumps" [3]

- IoT European Research Cluster: "IoT Governance, Privacy and Security Issues" [16]

- US Department of Homeland Security (DHS): "Securing the Internet of Things" [20]

- IEEE: "Internet of things Related Standards" [21]

- US Health Care Industry Cybersecurity (HCIC) Task Force: "Report of improving cybersecurity in the health care industry" [22]

Privacidad, GDPR y otras normativas aplicables en entornos ioT

En todo el mundo existen multitud de normativas y leyes referentes a la privacidad y la protección de los datos [17], siendo Europa con el GDPR (el nuevo reglamento de Protección de datos) una de las regiones con normativa más proteccionista con los datos personales. La mayoría de las legislaciones son genéricas respecto a la protección de datos personales (es decir, a lo denominado PII) aunque en Estados Unidos por ejemplo existen normativas como el HIPAA [10] que se refiere en exclusiva a datos personales de salud (PHI o "protected Health information").

El Reglamento (UE) 2016/679 o GDPR ("Reglamento General de Protección de Datos") [9] es una normativa europea, que unifica 28 leyes de privacidad de los diferentes países en una única y que aplica a todos los países de la UE, y a todas las empresas que gestionen datos personales de ciudadanos europeos (aunque no se encuentren en ella). El 25 de mayo de 2018 se acaba el plazo para la correcta implementación en toda la unión europea. La LOPD, que es la ley española de privacidad, sigue en vigor hasta que no se apruebe la nueva ley (y su correspondiente reglamento) que adapta el GDPR, que actualmente se encuentra en proceso de consultas y está prevista a mediados del 2018.

Cuando se implementan dispositivos ioT en entornos sanitarios en la mayoría de las situaciones se están gestionando datos considerados sensibles por esta normativa y por lo tanto hay que poner foco en aplicarla correctamente en nuestros entornos ioT (no solo en el

centro sanitario o entornos de pacientes, sino en los proveedores que gestionan la información).

Las principales características de esta normativa aplicadas al entorno sanitario con dispositivos ioT son las siguientes:

- Obligatorio la creación de un DPO ("Data Privacy Officer").
- Nuevo requisito de notificar incidentes de seguridad relativos a datos personales ("brechas") en menos de 72 horas, a las autoridades competentes, y en algunos casos a los afectados.
- Nuevos requisitos sobre consentimiento del titular de los datos. Los artículos 12 a 14 del Reglamento, obligan con carácter previo a la recogida de datos personales, a facilitar información clara y permanente al interesado sobre una serie de aspectos como son los fines y usos previstos, acciones que representan más dificultad en entornos ioT, aunque el propio reglamente permite el uso de enlaces a las políticas de seguridad o iconos normalizados (artículo 12.7). Además, un gran cambio con el nuevo reglamento (y de los que más impacto pueden tener en dispositivos que recogen información sensible), es que el consentimiento deberá prestarse de manera explícita o inequívoco, lo que requiere poder probar por parte del responsable de tratamiento que se otorgó.
- Responsabilidad proactiva ("Accountability"): el responsable del tratamiento tiene que cumplir las estipulaciones recogidas en el reglamento y se capaz de demostrarlo.
- Seguridad por defecto
- Nuevos derechos: "derecho al olvido" y a la "portabilidad" de los datos

- Evaluación de impacto de las operaciones de tratamiento en la protección de datos personales (EIPD)

En la página web de la Agencia Española de Protección de datos [(18)] se pueden consultar las guías publicadas de ayuda a la correcta implementación del reglamento, como por ejemplo la "Guía práctica de análisis de riesgos" o la "Guía práctica de Evaluaciones de impacto".

Otras normas y regulaciones españolas que hay que tener en cuenta en el entorno de ioT a la hora de su implantación y gestión en entornos sanitarios son:

- Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico

- Ley 9/2014, de 9 de mayo, General de Telecomunicaciones

- Ley 41/2002, de 14 de noviembre [(19)], básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica.

- Real Decreto Legislativo 1/2007, de 16 de noviembre, por el que se aprueba el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios (en el caso de implementar dispositivos ioT orientados a usuarios finales).

- Directiva 2016/1148 Del Parlamento Europeo y del Consejo de 6 de julio de 2016 relativa a las medidas destinadas a garantizar un elevado nivel común de seguridad de las redes y sistemas de información en la Unión (conocida como Directiva NIS).

Referencias Bibliográficas

1. Resolución de 9 de marzo de 2018, de la Entidad Pública Empresarial Red.es, M.P., por la que se publica el Convenio con el Servicio Andaluz de Salud, para la aplicación de las Tecnologías de la Información y Comunicación en la gestión de la cronicidad y la continuidad asistencial en el Sistema Sanitario Público de Andalucía. http://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-4368

2. NIST 8200 (Draft): Interagency Report on Status of International Cybersecurity Standardization for the Internet of Things (IoT)

   https://csrc.nist.gov/publications/detail/nistir/8200/draft

3. NIST Cybersecurity Practice Guide SP 1800-8, Wireless Infusion Pumps

   https://www.nccoe.nist.gov/sites/default/files/library/sp1800/hit-infusion-pump-nist-sp1800-8-draft.pdf

4. NIST Cybersecurity for IoT Program

   https://www.nist.gov/programs-projects/nist-cybersecurity-iot-program

5. FDA: “Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices” https://www.fda.gov/downloads/MedicalDevices/.../ucm089593.pdf

6. FDA: “Postmarket Management of Cybersecurity in Medical Devices”

https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf

7. FDA: "Cybersecurity for Networked Medical Devices Containing Off the-Shelf (OTS) Software" https://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM077823.pdf

8. Cybersecurity Vulnerabilities Identified in St. Jude Medical's Implantable Cardiac Devices and Merlin@home Transmitter: FDA Safety Communication https://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm535843.htm

9. GPDR: REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.

   http://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:32016R0679&from=ES

10. HIPAA: Health Information Privacy https://www.hhs.gov/hipaa

11. Estado del arte e implicaciones de seguridad y privacidad en el Internet de las Cosas´ (Centro de Estudios en Movilidad e IoT del ISMS Forum Spain) https://www.ismsforum.es/estudioCEM

12. CCN-CERT: Buenas Prácticas en Internet de las Cosas, IoT https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/2258-ccn-cert-bp-05-16-internet-de-las-cosas/file.html

13. Opinion on the Internet of things (Article 29 EU-GDPR Working Paper 223) .http://ec.europa.eu/justice/article-

29/documentation/opinion-recommendation/files/2014/wp223_en.pdf

14. ISO/TC 215 Health Informatics Standards Catalog https://www.iso.org/committee/54960/x/catalogue/

15. ENISA report: "Cyber security and resilience for Smart Hospitals" https://www.enisa.europa.eu/publications/cyber-security-and-resilience-for-smart-hospitals

16. IoT Governance, Privacy and Security Issues - IERC Position Paper http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Governance_Privacy_Security_Final.pdf

17. United Nations. Data Protection and Privacy Legislation Worldwide: http://unctad.org/en/Pages/DTL/STI_and_ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx

18. AGPD: Reglamento general de protección de datos: https://www.agpd.es/portalwebAGPD/temas/reglamento/index-ides-idphp.php#RGPD

19. Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica https://www.boe.es/buscar/pdf/2002/BOE-A-2002-22188-consolidado.pdf

20. US Department of Homeland Security (DHS): Securing the Internet of Things https://www.dhs.gov/securingtheIoT

21. IEEE: INTERNET OF THINGS RELATED STANDARDS http://standards.ieee.org/innovate/iot/stds.html

22. Report of improving cybersecurity in the health care industry: https://www.phe.gov/Preparedness/planning/CyberTF/Documents/report2017.pdf

Seguridad en Aplicaciones Web

Ramón Salado Lucena

La evolución de la tecnología ha llevado a las organizaciones a prestar cada vez más servicios desde internet, lo que supone que información que antes permanecía dentro de la organización (bases de datos de pacientes, historiales clínicos, resultados de análisis, etc.), pasen a poder ser accesible desde fuera. Esto supone que sea necesario realizar un importante esfuerzo para su protección. Por tanto, deberá tomar cuantas medidas sean necesarias para garantizar su:

- **Disponibilidad:** condición de la información de encontrarse a disposición de quienes deben acceder a ella.
- **Confidencialidad:** propiedad que impide la divulgación de información a individuos, entidades o procesos no autorizados.
- **Integridad:** propiedad de mantener con exactitud la información tal cual fue generada, sin ser manipulada ni alterada.
- **Autenticidad:** propiedad que permite identificar el generador de la información.

Para analizar la Seguridad de las Aplicaciones Web, utilizaremos diferentes recursos de OWASP, fundación sin ánimo de lucro, orientada a promover el desarrollo seguro de aplicaciones web y de software en general. Concretamente haremos referencia a la Guía de Testing (OWASP Testing Guide), al Top 10 de riesgos en aplicaciones web (OWASP Top10) y a la Guía de Desarrollo Seguro (OWASP Development). OWASP es el principal referente en seguridad para todos los desarrolladores web, y sus metodologías son incorporadas en todas las empresas de desarrollo del mundo.

Además de estos tres proyectos más conocidos de OWASP que hemos citado anteriormente, utilizaremos el OWASP Proactive Controls para introducirnos en la temática. Este, recoge 10 puntos de Control Proactivo de Seguridad, que deberían ser incluidos en todos los proyectos de desarrollo de aplicaciones web y de software en general, son:

1. **Verificar la seguridad al principio y periódicamente**

En la mayoría de las organizaciones, las pruebas de seguridad se llevan a cabo fuera del ciclo de pruebas de desarrollo, siguiendo la lógica "escanear y después arreglar". Los equipos de seguridad ejecutan sus escaneo automáticos o pruebas manuales, se clasifican los resultados y las vulnerabilidades pasan al equipo de desarrollo para ser parcheadas. Obviamente, hay mejores formas de hacerlo.

Las pruebas de seguridad deben ser parte integral de del desarrollo del producto. Debe verificarse la seguridad de forma temprana y frecuente, de manera que la aplicación sea robusta desde su desarrollo. Este sistema nos evitaría tener que rehacer una aplicación por un problema de seguridad.

Se deben aprovechar las prácticas ágiles como el desarrollo impulsado por prueba, la integración continua y las "*relentless testing*" (pruebas implacables). Estas prácticas responsabilizan a los desarrolladores de probar su propio trabajo, a través de circuitos de retroalimentación rápidos y automatizados.

2. **Parametrizar consultas**

Las inyecciones SQL son uno de los mayores riesgos de las aplicaciones web. En los últimos OWASP TOP 10, las inyecciones siempre han ocupado el primer puesto del ranking. Suelen ser fáciles de explotar, y existen varias herramientas automatizadas que lo simplifican aún más. Con una simple inyección de código SQL malicioso en una aplicación, se podría robar toda la base de datos, o modificar incluso borrar.

Para mitigar la inyección SQL, se debe evitar que la entrada que no sea de confianza se interprete como parte de un comando SQL. La mejor forma de hacerlo es con la técnica de programación conocida como Parametrización de Consultas. En este caso, las sentencias de SQL se envían y analizan por el servidor de base de datos independientemente de cualquier parámetro.

3. **Codificar datos**

La codificación es un mecanismo potente para ayudar a protegernos contra muchos tipos de ataques, especialmente los ataques de inyección XSS (*Cross-Site Scripting)*. Básicamente, la codificación implica traducir caracteres especiales en alguna forma equivalente que ya no sea peligrosa en el intérprete de destino.

4. Validar todas las entradas

Cualquier información que ingrese directamente o esté influenciada por el usuarios se debe tratar como no confiable. Una aplicación debe verificar que estos datos sean válidos. Además, las aplicaciones más seguras tratan todas las variables como no confiables y proporcionan controles de seguridad independientemente de la fuente de esos datos. La validación de la entrada debe ser totalmente del lado del servidor: los controles del lado del cliente se pueden usar por conveniencia.

5. Implementar controles de identidad y autenticación

Autenticación es el proceso de verificar que un individuo o una entidad es quien dice ser. Normalmente se realiza enviando un nombre de usuario o identificador y uno o más elementos que sólo el usuario conoce.

La gestión de sesiones se encarga de recordar al servidor la forma de reaccionar ante diferentes solicitudes de un cliente. Las sesiones se mantienen en el servidor mediante un identificador de sesión único.

En este extenso apartado podemos incluir los diferentes sistemas de autenticación como el de factor múltiple, basados en tokens, almacenamiento de contraseñas, mecanismos de recuperación de contraseñas, expiración de sesión, etc.

6. **Implementar controles de acceso**

Proceso que concede o deniega solicitudes de acceder a recursos concretos. El control de acceso es una de las áreas principales del diseño de seguridad de aplicaciones que debe estar muy estudiado desde el principio.

7. **Proteger los datos**

Al transmitir datos confidenciales, en cualquier nivel de su aplicación o arquitectura de red, se debe considerar el cifrado en tránsito de algún tipo. TLS es el modelo más extendido, y a pesar de las debilidades publicadas, sigue siendo el método recomendado para implementar cifrado en la capa de transporte.

Debemos clasificar los datos y determinar cuáles debe cifrarse para su almacenamiento, es el caso de los datos de tarjetas de crédito según el estándar PCI-DSS.

Es necesario que nos aseguremos de que los datos confidenciales o no, no sean expuestos por accidente durante el procesamiento. Puede ser más accesible en la memoria, o podría escribirse en ubicaciones de almacenamiento temporal o archivos de registro, donde un atacante podría leerlo.

8. **Implementar el registro y la detección de intrusos**

El registro de aplicaciones no debe ser una idea de último momento o estar limitado a la depuración y resolución de problemas. El registro también se usa en otras actividades importantes, como la monitorización de la aplicación, auditoría de actividad y supervisión del cumplimiento, detección de intrusos en el sistema, forense, etc.

Es importante no registrar demasiado o muy poco. Asegúrese de registrar siempre la fecha y la información de identificación como la IP de origen y el ID de usuario, pero tenga cuidado de no registrar datos privados o confidenciales o datos o secretos.

9. **Aprovechar los frameworks de seguridad**

Las librerías y los frameworks con la seguridad incorporada ayudan a los desarrolladores de software a proteger sus trabajos contra los errores en el diseño y los defectos de implementación. Un desarrollador que cree una aplicación desde cero podría no tener el tiempo y el presupuesto suficientes para implementar las funciones de seguridad adecuadas.

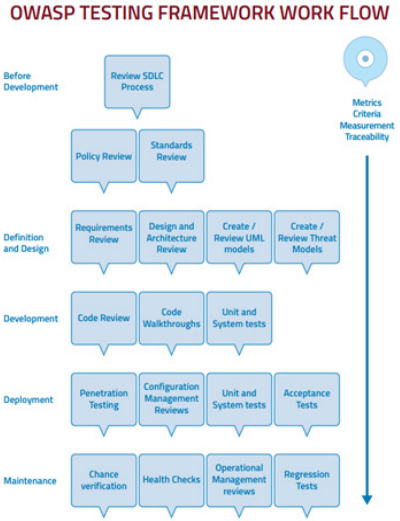
10. **Manejo de errores y excepciones**

Los fallos en el manejo de errores pueden conducir a diferentes tipos de vulnerabilidades como por ejemplo dando a conocer detalles que permiten al atacante conocer mejor la arquitectura interna de nuestra aplicación.

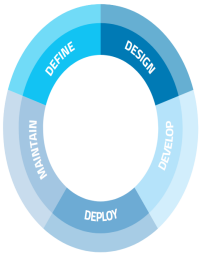
Se recomienda administrar las excepciones de forma centralizada para evitar la duplicación de los bloques “try / catch” en el código y para garantizar que todos los comportamientos inesperados se manejen correctamente dentro de la aplicación.

Testing sobre Aplicaciones Web

La Guía de Testing de OWASP (versión 4, de 3 de agosto de 2015) pretende alentar a los usuarios a evaluar y tomar las correspondientes medidas de seguridad a través de todo el proceso de desarrollo. Nos presenta una metodología que recorre de forma organizada y sistemática todas las posibles áreas que supongan vectores de ataque a una aplicación web. Organiza la auditoría en etapas según el estado de madurez en el desarrollo de la aplicación. Y nos propone un framework de pruebas donde se identifican y detallan los puntos de control sobre los que se aplicarán los tests correspondientes.



Como vimos en los Controles Proactivos, el testing no se puede relegar a una etapa donde el proyecto ya está desarrollado, el testing debe estar integrado en cada fase del Ciclo de Vida del Desarrollo de software (SDLC)



La Guía también elabora una lista de principios a tener en cuenta para el proceso de Testing:

- **No existe una bala de plata**. No hay un método definitivo ni una herramienta perfecta.

- **Pensar estratégicamente, no tácticamente**. El modelo clásico de "parche y penetración" implica que al descubrirse una vulnerabilidad tenemos que ir rápidamente a corregirla, en muchos casos sin una investigación correcta del causante del problema de seguridad.
- **El SDLC es el rey**, lo comentamos en el apartado anterior. Cada fase tiene consideraciones de seguridad que deberían formar parte del proceso existente, para garantizar un programa de seguridad integral y rentable.
- **Detecta pronto y prueba continuamente**. Lo vimos en los Controles Proactivos, es necesario incluir la seguridad desde la fase inicial de desarrollo, para que ya se ejecute con unas bases fuertes. Además es preciso que los controles sean periódicos, ya que continuamente tenemos actualizaciones que nos modifican el entorno.
- **Comprender el alcance de la seguridad**. Una prueba de seguridad de calidad no supervisa sólo los comportamientos normales dentro de la aplicación, es necesario ir más allá y ser creativo en la auditoría. Por esto no debemos depender en exceso de herramientas automatizadas.
- **Entender la situación**. Conocer y comprender el funcionamiento interno de la aplicación, los diagramas de flujo, las especificaciones de cada proceso.
- **Utilice las herramientas adecuadas**. Conocimiento sobre las herramientas que se utilizan para los test, para hacer un uso responsable y correcto de ellas.
- **El diablo está en los detalles**. Un análisis superficial puede crear una falsa sensación de seguridad. Es preciso que analicemos de forma minuciosa y seamos capaces de descartar falsos positivos.
- **Usa Código Fuente si está disponible**. Una auditoría "*black box"* puede proporcionarnos unos resultados muy interesantes, analizando la aplicación desde fuera, pero si

disponemos del código fuente, podremos hacer un estudio más profundo.

- **Desarrolla métricas**. Crearemos métricas que nos permitan hacer un seguimiento a los resultados.
- **Documentar los resultados de la prueba**. Es una parte vital, que permitirá a los desarrolladores ser más certeros en la revisión.

Además de estas directrices, la Guía de Testing de OWASP recopila 91 puntos de control, agrupados en 11 capítulos:

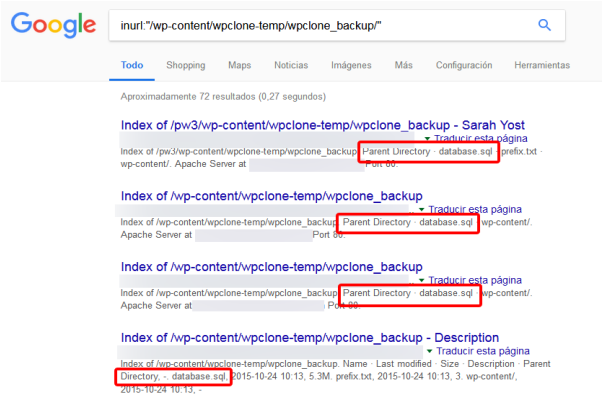
1. Information Gathering
2. Configuration and Deployment Management Testing
3. Identity Management Testing
4. Authentication Testing
5. Authorization Testing
6. Session Management Testing
7. Input Validation Testing
8. Error Handling
9. Cryptography
10. Business Logic Testing
11. Client Side Testing

A continuación realizaremos algunas de las pruebas más importantes de las recogidas en la Guía de Testing.

- DESCUBRIMIENTO EN BUSCADORES DE FUGAS (OTG-INFO-001):

Para esta prueba se utilizan diferentes motores de búsqueda, destacando entre otros: Google, Bing, Shodan, Censys, etc. Haremos uso de algunos operadores que nos permitan visualizar si se está indexando en los resultados de búsqueda, alguna información que no

debería ser pública (nombres de usuario, mensajes de error, logs, archivos, etc.) En el ejemplo siguiente, utilizando Google Hacking DataBase (GHDB) conseguimos descargar la base de datos de una web, debido a una fuga de información indexada en Google.



- FINGERPRINT DEL WEB SERVER (OTG-INFO-002):

Con esta prueba pretendemos conocer si nuestro Servidor Web nos permite conocer qué tipo de servidor es y la versión tenemos instalada. Esta información es vital para cualquier atacante, ya que le facilitamos la búsqueda de vulnerabilidades de nuestro sistema.

```
$ nc 202.41.76.251 80
HEAD / HTTP/1.0
HTTP/1.1 200 OK
Date: Mon, 08 Jan 2018 02:53:29 GMT
Server: Apache/1.3.3 (Unix) (Red Hat)
Last-Modified: Wed, 07 Oct 1998
11:18:14 GMT
ETag: "1813-49b-361b4df6"
Accept-Ranges: bytes
Content-Length: 1179
Connection: close
Content-Type: text/html
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Expires: Yours, 17 Jun 2003
01:41: 33 GMT
Date: Mon, 08 Jan 2018 01:41:
33 GMT
Content-Type: text/HTML
Accept-Ranges: bytes
Last-Modified: Wed, 28 May 2003
15:32: 21 GMT
ETag: b0aac0542e25c31: 89d
Content-Length: 7369
```

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Tue, 16 Jan 2007 14:53:45
GMT
Content-length: 1186
Content-type: text/html
Date: Mon, 08 Jan 2018 14:50:31
GMT
Last-Modified: Wed, 10 Jan 2007
09:58:26 GMT
Accept-Ranges: bytes
Connection: close
```

La técnica de extracción de la firma del servidor se conoce como "Banner Grabbing" y la podemos utilizar con NetCat, Telnet, o herramientas como Httprint.

- REVISIÓN DE META ARCHIVOS DEL SERVIDOR (OTG-INFO-003):

En muchos casos, se utiliza el fichero robots.txt para bloquear el acceso de robots a determinadas parte privadas de una web, dejando visible a cualquier visitante de este fichero esta información.

```
User-agent: *
Disallow: /search
Disallow: /sdch
Disallow: /groups
Disallow: /images
Disallow: /catalogs
```

- TEST MÉTODOS HTTP (OTG-CONFIG-006):

HTTP ofrece una serie de métodos que se pueden usar para realizar acciones en el servidor web. Muchos de estos métodos están diseñados para ayudar a desarrolladores en la implementación y prueba de aplicaciones HTTP, pero una mala configuración del servidor puede tener nefastas consecuencias, si se utilizan estos métodos de forma maliciosa.

Los métodos más frecuentes son HEAD, GET, POST, OPTIONS,… pero existen otros más, algunos muy peligrosos como PUT, DELETE, CONNECT o TRACE, que deberían desactivarse o al menos controlarse.

```
$ nc www.victim.com 80
TRACE / HTTP/1.1
Host: www.victim.com
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sun, 11 Jan 2018 08:01:48 GMT
Connection: close
Content-Type: message/http
Content-Length: 39
TRACE / HTTP/1.1
Host: www.victim.com
```

```
$ curl -s -D- https://domain.com/ | grep Strict
Strict-Transport-Security: max-age=60000;
```

```
$ nc www.example.com 80
JEFF / HTTP/1.1
Host: www.example.com
HTTP/1.1 200 OK
Date: Mon, 8 Jan 2018 22:38:40 GMT
Server: Apache
Set-Cookie: PHPSESSID=K53QW...
```

- TEST DE CREDENCIALES TRANSPORTADOS SOBRE CANAL CIFRADO (OTG-AUTHN-001):

  El análisis se enfoca simplemente a verificar si los datos viajan sin cifrar desde el navegador web al servidor. El hecho de que el tráfico está cifrado no necesariamente significa que sea completamente seguro. La seguridad también depende del algoritmo de cifrado utilizado y la solidez de las claves que la aplicación esté usando. Con Wireshark o cualquier otro sniffer en la red podemos verificarlo.

- TEST SALTAR SISTEMA DE AUTENTICACIÓN (OTG-AUTHN-004):

  A menudo es posible eludir las medidas de autenticación alterando las solicitudes para engañar a la aplicación haciéndole creer que el usuario ya está autenticado. Esto se puede lograr modificando el parámetro de URL dado, o por predicción de ID de sesión.

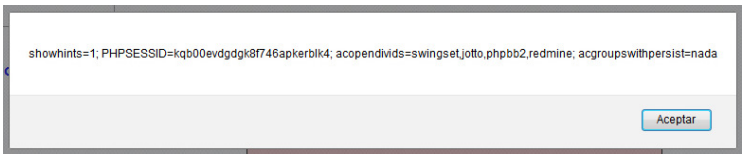
- TEST CSRF (OTG-SESS-005):

  Falsificación de petición en sitios cruzados (Cross Site Request Forgery (CSRF o XSRF)). Este ataque fuerza al navegador de la víctima, validado en algún servicio (correo, banco, compras) a enviar una petición a una aplicación web vulnerable. Esta aplicación se encarga de realizar la acción elegida a través de la víctima, ya que la actividad maliciosa será procesada en nombre del usuario logueado.
  Este tipo de ataques se pueden mitigar con el uso de tokens aleatorios para acciones comprometidas o de un captcha para que el usuario tenga que interactuar con la aplicación.

- TEST XSS REFLEJADO (OTG-INPVAL-001):

  Cross Site Scripting una de las vulnerabilidades clásicas de las aplicaciones web. Son inyecciones de código malicioso, evitando las medidas de control. Existe diferentes tipos de inyecciones XSS, en este caso concreto, no quedan almacenadas las modificaciones, tan sólo se reflejan en el navegador durante la

sesión. El *Cross Site Scripting* puede ser utilizado para robar sesiones de usuario, comprometer la seguridad del cliente a través de *phishing*, etc.
Por ejemplo, insertando el código *<script>alert(document.cookie) </script>* obtendríamos la siguiente información:



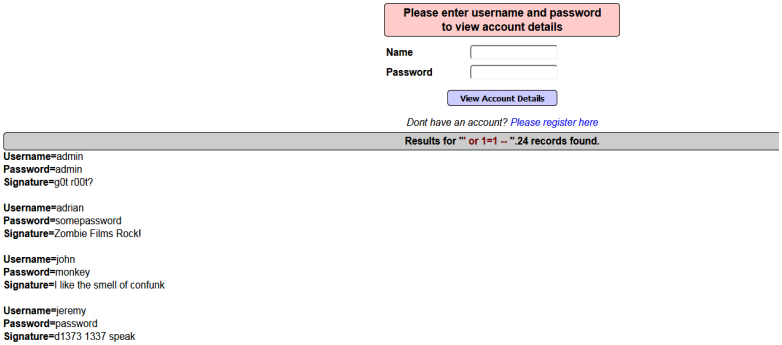
Se nos muestra un una caja de alerta de JavaScript con información como la ID de sesión PHP.

- TEST XSS ALMACENADO (OTG-INPVAL-002):

  A diferencia del caso anterior, la inyección modifica el contenido de la aplicación web y se almacena en el servidor modificada. Los datos maliciosos se mostrarán como parte del sitio web, ejecutándose con los mismos privilegios en el navegador.
  Ambas se pueden probar de forma automatizada con herramientas como WebScrab, ZAP o Burp Proxy.

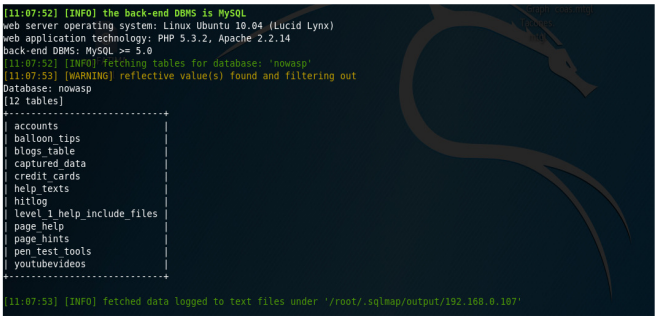
- TEST SQLi (OTG-INPVAL-005):

  Otra de las vulnerabilidades más conocidas en las aplicaciones web. Se trata de una modificación de las consultas SQL de la aplicación web, para extraer, modificar o eliminar algunos datos. Hablamos de inyección ya que se trata de insertar código malicioso dentro de una consulta ya existente. Por ejemplo, introduciendo el código *' or 1=1 –* obtendríamos un SELECT abierto para la consulta SQL, ya que la condición 1=1 se cumple siempre.



En este ejemplo obtenemos la tabla completa de usuarios y contraseñas de la aplicación, pero una vez localizada esta vulnerabilidad, nuestro límite es la imaginación y nuestro conocimiento de SQL. Si capturamos la petición GET que hace la aplicación web y la utilizamos con SQLmap, podemos mostrar todas las tablas y sus contenidos.

sqlmap -u 'http://domain.com/index.php?page=user-info.php' --data 'username=john&password=monkey&user-info-php-submit-button=View+Account+Details' --cookie= "showhints=1; PHPSESSID=kqb00evdgdgk8f746apkerblk4;" -D nowasp --tables



- ANÁLISIS DE CÓDIGOS DE ERROR (OTG-ERR-001):

  Es frecuente que durante un test de penetración en aplicaciones web, nos encontramos con errores generados por aplicaciones o servidores web. Estos códigos son muy útiles para la auditoría, ya que revelan mucha información sobre bases de datos, errores y otros componentes directamente relacionados con aplicaciones web.

  - http://www.example.com/index.jhtml?ID=222222222222222222222222222

  ERROR:
  --- The error occurred in **ibatis**/categorie_abc.xml.
  --- The error occurred while preparing the mapped statement for execution.
  --- Check the getByLogicalCategoryId.
  --- Cause: **java.sql.SQLException:** Invalid parameter object type. Expected 'java.lang.Long' but found 'java.lang.Double'

OWASP Top 10

Es uno de los proyectos más conocidos de OWASP, donde recopila los 10 riesgos más habituales en las aplicaciones web. La última versión se publicó en 2017, y se elaboraba con la información de cientos de empresas y organizaciones que reportan los diferentes incidentes que detectan.

Según la empresa *Veracode* (participante en el proyecto), el 77% de las aplicaciones web, tiene al menos uno de estos 10 fallos de seguridad.

El Top 10 es un estándar para que los desarrolladores puedan determinar y mitigar las causas que hacen las aplicaciones inseguras. El ranking ha ido evolucionando con los años, aunque varias vulnerabilidades están presentes desde el principio. Esto implica por una parte que las tecnologías y protocolos que utilizamos son inseguros, y que además no se presta toda la atención que debería a la seguridad en el desarrollo.

Referencias Bibliográficas

OWASP Testing Guide
https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

OWASP Top10
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP Development
https://www.owasp.org/index.php/Projects/OWASP_Development_Guide

OWASP Proactive Controls
https://www.owasp.org/index.php/OWASP_Proactive_Controls

SANS, Securing Web Application Technologies [SWAT] Checklist
https://software-security.sans.org/resources/swat

ISSA, Network Security Architecture, Mariusz Stawowski
http://www.clico.pl/services/Network_Securite_Architecture.pdf

Open Security Architecture, Public Web Server Pattern
http://www.opensecurityarchitecture.org/cms/library/patternlandscape/222-pattern-public-web-server

Veracode, What's New in the State of Software Security 2017
https://www.veracode.com/blog/research/introducing-state-of-software-security-2017-report

Seguridad en Aplicaciones iOS

Miguel Ángel Arroyo Moreno

Introducción a la seguridad en iOS

Tal y como estamos viendo a lo largo de este libro, la superficie de ataque en un entorno IoT es muy amplio, incluyendo multitud de áreas y vectores de ataque.

Una de sus áreas se refiere a las aplicaciones móviles que permiten el control o gestión de los dispositivos presentes en estos entornos, por ejemplo, para el apagado o encendido de una *Smart Bulb* (bombilla inteligente) desde la app del móvil.

Está claro que supone una ventaja, y sobre todo una comodidad, el hecho de que desde el sofá de nuestro salón podamos apagar, encender o regular la intensidad de esa bombilla.

O lo que es mejor, poder hacerlo cuando nos encontremos fuera de casa, sobre todo cuando son varios los días que nos ausentamos, para simular que hay alguien en el hogar, mediante el apagado y encendido manual y remoto de las luces, o de forma automatizada y programada.

Pero, ¿nos hemos parado a pensar en los riesgos que supone el uso de estas nuevas tecnologías? ¿Son seguras las comunicaciones que se establecen cuando interaccionamos con este entorno IoT? ¿Se

almacenan de forma segura los datos de la aplicación? ¿Atenta contra nuestra privacidad al facilitar datos sin nuestro consentimiento al fabricante o terceros, como puede ser por ejemplo la geolocalización de las bombillas?

No podemos confiar en que el fabricante ha tomado todas las medidas oportunas para hacer que su producto sea totalmente seguro, es más, según los últimos estudios y aunque hay una tendencia a la mejora, aún es insuficiente el esfuerzo de éstos en cuanto a seguridad desde el diseño.

Esto significa que muchas de las vulnerabilidades de estos dispositivos no se detectan a tiempo, en las fases de diseño o desarrollo del producto, y es en producción, es decir, una vez comercializado, cuando se detectan, poniendo en riesgo la información de los usuarios de estos dispositivos.

Llegados a ese punto, es importante qué recursos tenemos disponibles para evaluar la seguridad de las aplicaciones móviles que gestionan y controlan dispositivos IoT.

En este capítulo vamos a tratar la seguridad de las aplicaciones iOS, centrándonos en la arquitectura de seguridad que actualmente Apple nos proporciona en sus sistemas iOS y cuáles son las técnicas y herramientas que podemos utilizar para evaluar la seguridad de las aplicaciones.

En la actualidad tenemos varias organizaciones que podemos tener como referencia, y apoyarnos en sus recursos, como guías y herramientas, para llevar a cabo nuestra labor de evaluación de seguridad de sistemas iOS:

- **Apple**: Como no podía ser de otra forma, una de las mejores referencias es la propia organización o fabricante de estos sistemas. Apple pone a nuestra disposición una guía de seguridad de su sistema operativo, llamada "iOS Security Guide" y que actualiza periódicamente (normalmente después de actualizaciones importantes o una revisión importante de su versión).

- **ENISA**: La Agencia de Seguridad de las Redes y de la Información de la Unión Europea dispone también una serie de recursos que nos pueden ayudar a la hora testear la seguridad de estos sistemas. Dispone además de guías de desarrollo seguro de aplicaciones móviles, algo fundamental de cara a aplicar seguridad desde el diseño (*Security by Design)*, un concepto que cobra cada vez más importancia en el ámbito de la seguridad de la información.

- **OWASP:** *Open Web Application Security Project* es proyecto abierto de seguridad en aplicaciones web. Aunque nació solo con esta perspectiva, a día de hoy cuenta también con proyectos específicos de seguridad de aplicaciones móviles y seguridad en IoT. Pone a nuestra disposición una gran cantidad de recursos muy útiles como son guías de evaluación, guías de desarrollo seguro y herramientas.

Arquitectura de seguridad en iOS

Antes de adentrarnos en las técnicas y herramientas para evaluar la seguridad de las aplicaciones iOS, haremos una breve aproximación a la arquitectura de seguridad de este sistema operativo, lo que nos ayudará a entender mejor cómo se protege iOS y cómo protege a sus aplicaciones.

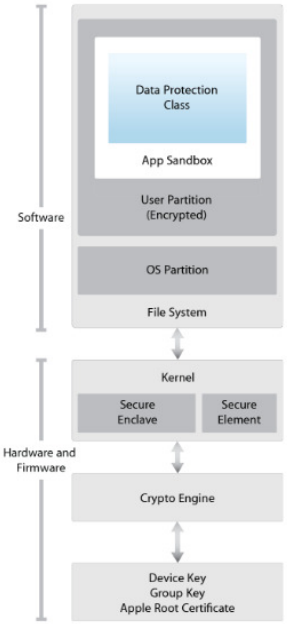


Diagrama de arquitectura de seguridad en iOS

Apple diferencia claramente en su arquitectura de seguridad la parte de software de la de hardware / firmware. En la primera parte encontramos todo lo relacionado con el sistema de ficheros, particiones del sistema operativo, partición del usuario, ***sandbox*** (espacio aislado y reservado) de la aplicación y finalmente los datos.

En la segunda parte, referente al hardware y el firmware, encontraremos todo lo relacionado con el ***kernel*** (núcleo del sistema), el Secure Enclave (un coprocesador de cifrado), el motor de cifrado, las claves del dispositivo y el certificado raíz de Apple.

Todos estos elementos conforman la arquitectura de seguridad de iOS y son los responsables de proteger las aplicaciones y datos de los usuarios, entre otras tareas.

A nivel de arranque del dispositivo, los sistemas iOS cuentan con un sistema seguro que consiste en una cadena segura de arranque, compuesta por: *Boot ROM, LLB, iBoot y el Kernel*.

A continuación, se muestra un diagrama del proceso de arranque:

En esta cada de arranque, se llevan a cabo diferentes procesos en distintas fases, donde se comprueba continuamente la integridad del sistema durante el encendido del dispositivo y carga del sistema operativo.

Estas comprobaciones se realizan para evitar que se pueda cargar un software manipulado en el sistema, por ejemplo, un sistema operativo iOS que ha sido previamente manipulado (contenga un código malicioso o puerta trasera) y no esté firmado por Apple.

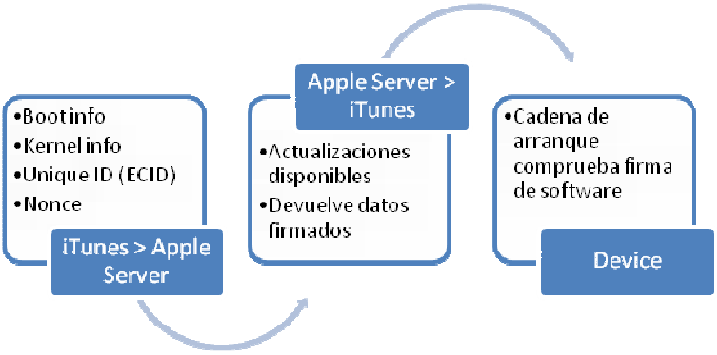
Una vez arrancado el sistema, cabe destacar que la partición del usuario se encuentra por defecto cifrada. Esta medida es una de las más efectivas a la hora de proteger la confidencialidad o integridad de la información.

Para las actualizaciones del sistema, también cuenta con un proceso robusto y seguro. El dispositivo comprueba periódicamente si hay nuevas actualizaciones disponibles, y si las hay, intercambia información con el servidor de Apple, al cual le facilita información de arranque, del inicio y un identificador único.

Con esta información, el servidor de actualizaciones de Apple devuelve los datos de las actualizaciones disponibles y lo hace firmando dichos datos. Esta firma garantizará la integridad de los datos, evitando que alguien o algo manipule los datos durante su transmisión o instalación.

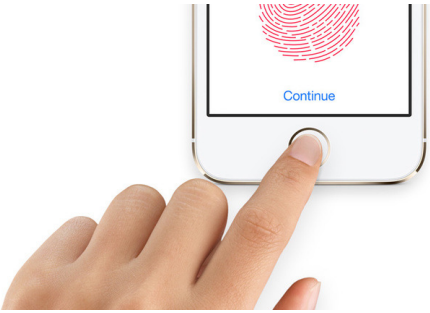
El dispositivo, antes de instalar las nuevas actualizaciones, comprueba la firma del software, y solo si es correcta, procederá a la instalación de las actualizaciones y al reinicio del sistema si fuera necesario.

A continuación, se muestra un diagrama con detalle de la información compartida entre dispositivo y servidores, así como los diferentes procesos que se llevan a cabo durante una actualización de software en dispositivos iOS.



Otra característica importante a tener en cuenta, y que tiene que ver con el cifrado y la autenticación, es el Secure Enclave de Apple.

Este sistema gestiona el cifrado, las credenciales y la autenticación local en las aplicaciones. Uno de sus elementos es el sistema de autenticación biométrica a través de la huella dactilar, y que se conoce como el Touch ID.



Mención especial merece también la firma de aplicaciones, lo que garantiza la autenticidad y la integridad de una aplicación. Mediante un sistema de clave asimétrica (combinación de clave pública y clave privada), las aplicaciones se firman justo antes de ser publicadas. Una vez la aplicación esté firmada y aprobada por Apple, ésta podrá ser publicada en el Apple Store.

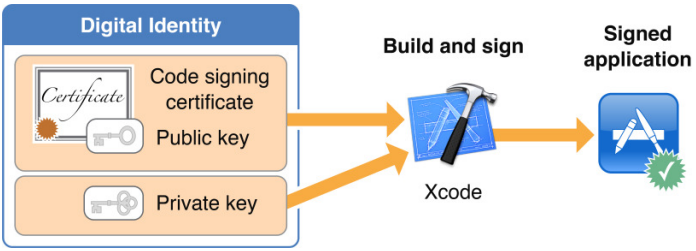


Imagen de developer.apple.com

Para finalizar este apartado acerca de la arquitectura de seguridad en iOS, veremos cómo gestiona este sistema operativo la seguridad en tiempo de ejecución.

Como hemos comentado al comienzo de este apartado, iOS utiliza el concepto de ***Sandboxing*** para aislar el espacio de memoria y proceso de la aplicación para que no pueda ser accedida por objetos que no tienen permiso para ello.

Esta es una medida de seguridad muy interesante ya que no permitiría por ejemplo a una aplicación maliciosa consultar o modificar datos de una aplicación legítima.

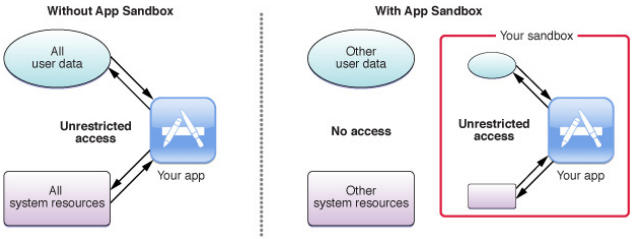


Imagen de developer.apple.com

Como se puede ver en la imagen anterior, de esta forma iOS consigue restringir el acceso a datos de la aplicación y del usuario desde otros recursos del sistema, salvo a aquellos a los que explícitamente se les ha dado acceso.

OWASP como referencia

En este capítulo tomaremos como referencia a OWASP, ya que entendemos que es el marco ideal por el cual guiarnos, al disponer tanto de proyectos de seguridad de aplicaciones móviles como proyectos de seguridad de IoT.

Además, en trabajos tan complejos como puede ser la de auditar una aplicación móvil de iOS se requiere seguir una metodología, mediante la cual podamos marcar unos procedimientos y objetivos claros del proyecto.

Centrándonos en el proyecto de seguridad en aplicaciones móviles de OWASP, *Mobile Security Project* (MSP), podemos destacar dos objetivos principales; ayudar a los desarrolladores de aplicaciones móviles y ayudar a los auditores de seguridad que las evalúan.

Objetivos:

- Ayuda al desarrollo mediante recursos que facilitan llevar a cabo y mantener un ciclo de vida de desarrollo de software seguro (**S-SDLC, Secure Software Development LifeCycle**).

- Ayuda a la evaluación de la seguridad de las aplicaciones mediante recursos que facilitan verificar la eficacia y eficiencia de los controles de seguridad implantados en la aplicación.

El proyecto Mobile Security Project de OWASP cuenta con una gran cantidad de recursos muy útiles para cualquiera de estos dos perfiles, ya sea el de desarrollador como el de auditor.

Entre estos recursos podemos encontrar:

- Top 10 de riesgos en aplicaciones móviles
- Guía de desarrollo seguro
- Guía de testeo de seguridad
- Herramientas
- Top 10 de controles de seguridad
- Esquema resumen de desarrollo seguro

Top 10 de riesgos en aplicaciones móviles según OWASP

A continuación, se relacionan las diez categorías de riesgos más importantes en aplicaciones móviles, según el proyecto OWASP.

- **M1 – Uso inapropiado de la plataforma**: esta categoría engloba todos los riesgos relacionados con fallos de la plataforma (iOS) o un mal uso de ésta. Algunos ejemplos son el uso inapropiado del Touch ID, permisos o de las claves.

- **M2 – Almacenamiento inseguro de datos**: cubre los riesgos relacionados con un almacenamiento no seguro de los datos o una revelación no intencionada de información. Un ejemplo podría ser unas credenciales almacenadas en texto plano.

- **M3 – Comunicaciones inseguras**: este apartado recoge los riesgos asociados al uso de versiones incorrectas o vulnerables de SSL, negociación débil de la conexión o transmisión en texto plano de información sensible.

- **M4 – Autenticación insegura**: esta categoría incluye los riesgos relacionados con una mala gestión de identidades, autenticación débil o una incorrecta gestión de las sesiones de los usuarios autenticados en la aplicación.

- **M5 – Criptografía insuficiente**: aquí se incluyen todos los riesgos derivados de un intento fallido de implementación de criptografía en la aplicación. *Nota: todo lo relacionado con SSL o TLS iría en M3, y si se trata de no implementar toda la criptografía que se debiera, iría en M2*.

- **M6 – Autorización insuficiente**: esta categoría recoge todos los riesgos asociados a una mala implementación de la autorización en la aplicación, permitiendo por ejemplo forzar la navegación a unos recursos los cuales deberían estar restringidos solo para usuarios autorizados. Si la aplicación no autentica a los usuarios en ningún momento y

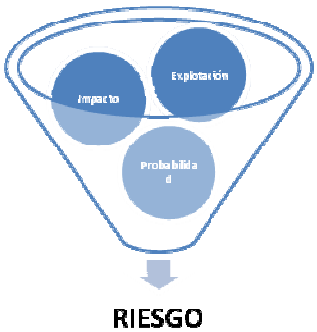
permite por ejemplo el acceso a anónimo, sería un problema de autenticación y no de autorización.

- **M7 – Calidad del código cliente**: aquí encajan todos los riesgos relacionados con problemas de implementación de la parte cliente de la aplicación, que difieren de los del lado servidor. Algunos ejemplos son desbordamiento de búfer o vulnerabilidades de formato de cadena.

- **M8 – Manipulación de código**: esta categoría cubre el parcheado de binarios, modificación de recursos locales, manipulación de métodos y modificación de memoria.

- **M9 – Ingeniería inversa**: en este apartado se incluyen tareas para obtener el código fuente, librerías, algoritmos y otros recursos a partir de análisis del binario. Estas técnicas pueden ser usadas para la explotación de otras vulnerabilidades presentes en la aplicación.

- **M10 – Funcionalidad extraña**: esta categoría incluye todos los riesgos derivados de malas prácticas por parte de los desarrolladores, como puede ser la inclusión de comentarios con información sensible, una contraseña visible o el desactivado del doble factor de autenticación (si existiera).



Top 10 de controles de seguridad

Al igual que ocurre con los riesgos, disponemos de un Top 10 para los controles de seguridad más importante de cara a proteger una aplicación iOS. Este Top 10 ha sido definido conjuntamente por OWASP y ENISA.

La finalidad es que tanto los desarrolladores como los auditores tengan claro cuáles son las medidas más importantes a implantar para proteger la aplicación.

Los controles están definidos en base a los riesgos que hemos comentado anteriormente, y su objetivo es garantizar la confidencialidad, integridad y disponibilidad de la aplicación móvil.

El conjunto de controles que aquí se expone tendrá que ser evaluado periódicamente para comprobar la eficacia y la eficiencia de cada control, para verificar que realmente están haciendo su labor que no es otra que proteger la información de la aplicación.

Los controles definidos por OWASP y ENISA son los siguientes:

- C1: Identificar y proteger la información sensible de la aplicación.
- C2: Proteger las credenciales de autenticación.
- C3: Proteger los datos en tránsito.
- C4: Correcta autenticación, autorización y gestión de sesiones.
- C5: Asegurar servicios y servidores.
- C6: Asegurar integración de terceros.
- C7: Consentimiento para recolectar y usar datos del usuario.
- C8: Proteger los servicios de pago electrónico.
- C9: Mantener sistema y aplicación actualizada.
- C10: Proteger la aplicación en tiempo de ejecución.

Jailbreak

Jailbreak es la técnica que permite, como indica su nombre, romper la jaula de seguridad que Apple implementa en todos sus sistemas operativos y dispositivos.

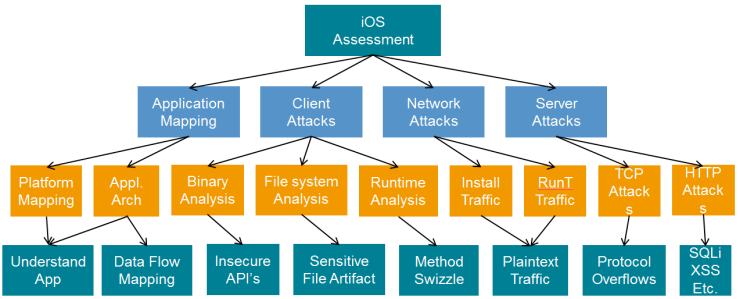
La mayoría de usuarios que hacen Jailbreak en sus dispositivos, lo hacen para poder obtener alguna funcionalidad especial en sus dispositivos móviles, que por defecto no se activa, o para poder instalar aplicaciones de forma "*gratuita*" a través de tiendas alternativas de aplicaciones (*Stores*, tiendas alternativas), como es el caso de **Cydia**.

Esto, que a priori, puede entenderse como una ventaja, se puede tornar en un inconveniente ya que estamos deshabilitando toda la cadena de seguridad que Apple nos implementa con su arquitectura. Por lo tanto, atención, porque no todo son ventajas.

Imagen de macworld.co.uk

A través de la tienda "*alternativa*" (Cydia), podremos instalar todas las herramientas necesarias para realizar la auditoría de aplicaciones móviles en iOS. OWASP nos propone otro recurso que precisamente es un arsenal de herramientas recomendado para realizar las tareas de auditoría.

Cada herramienta tiene una funcionalidad específica, que nos permitirá evaluar cada control de seguridad con pruebas concretas. Estas pruebas están diferenciadas y categorizadas en el esquema resumido de la guía de testeo de OWASP. A continuación, se puede ver un esquema específico para iOS.



Para diferenciar las técnicas y herramientas, este esquema agrupa los ataques en los siguientes apartados:

- Mapeo de la aplicación
- Ataques del lado cliente
- Ataques a las comunicaciones
- Ataques del lado servidor

No entra en el alcance de este libro explicar el detalle técnico de las herramientas y técnicas utilizadas para hacer las auditorías de aplicaciones iOS. El objetivo de este capítulo es adentrar al lector en la seguridad de estas aplicaciones, desde la arquitectura del sistema, los riesgos, los controles y los diferentes recursos disponibles proporcionados por marcos de referencia como OWASP o ENISA.

Referencias Bibliográficas

1. iOS Security Guide de Apple: https://www.apple.com/business/docs/iOS_Security_Guide.pdf
2. OWASP Mobile Security Project: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Home
3. OWASP Mobile Security Project – Top 10 Mobile Risks: https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
4. OWASP Mobile Security Project – Testing Guide: https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide#tab=Main
5. Top Ten Smartphone Security Controls for Developers: https://www.enisa.europa.eu/news/enisa-news/top-ten-smartphone-security-controls-for-developers

Seguridad en Aplicaciones Android

Ramón Salado Lucena y Juan José Domenech Sánchez

Introducción a la seguridad en Android

La irrupción de los smartphones sin duda ha cambiado nuestra forma de comunicarnos para siempre. Llevamos toda nuestra vida en el bolsillo: nuestros contactos, el banco, ubicaciones, la música que nos gusta, datos sobre nuestros hábitos, etc. Como consecuencia, en caso de que vulneren nuestro dispositivo, ya sea de forma física o lógica, nuestra privacidad corre un grave riesgo.

A estos casos, se les suma ahora las apps que forman parte de los ecosistemas IoT, una nueva forma de relacionarnos con el mundo físico, que ya forma parte de nuestras vidas, y que suponen un nuevo foco de ataques.

Con los avances tecnológicos que estamos viviendo, la capacidad de conectividad inalámbrica permite que los dispositivos físicos se hayan convertidos en "inteligentes" permitiendo accesos remotos para su gestión y administración. Este cambio ha supuesto una inversión en seguridad que ha sido adoptado por los proveedores de forma lenta, debido en gran parte a que muchos sistemas, sobre todo a nivel industrial, se fabricaran hace décadas, lo que ha dificultado su actualización.

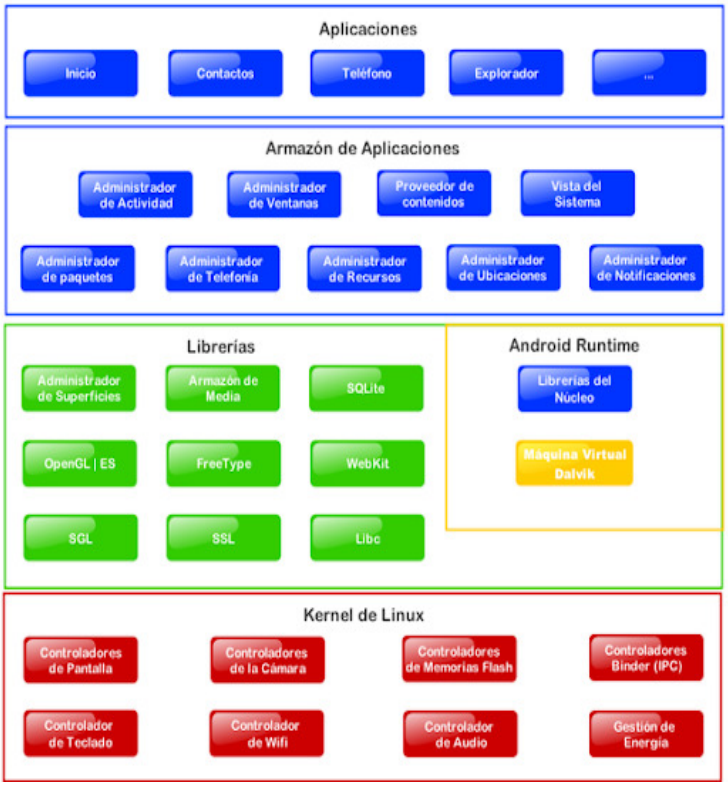
Por otro lado, en los casos de dispositivos modernos comerciales, las prisas por salir al mercado y ser el primero en explotar un sector o la disminución de costes en su proceso de diseño y desarrollo, han relegado a último lugar la seguridad.

En los siguientes apartados veremos el modelo de seguridad de Android y su arquitectura, de forma que tengamos un concepto amplio de su funcionamiento para poder aplicar las guías y herramientas para evaluar la seguridad.

Arquitectura de Android

Android está formado por cuatro capas diferentes, que facilitan al programador el desarrollo de aplicaciones, ya que proporciona diferentes librerías para que no sea necesario programar a bajo nivel para hacer uso de los elementos hardware del dispositivo.

Cada capa utiliza elementos de la inmediatamente inferior para sus funciones, por lo que a esta arquitectura se le denomina Pila



- **KERNEL DE LINUX**: Núcleo de Android, basado el kernel de Linux, similar al que podemos encontrar en cualquier distribución de

GNU/Linux, con la única diferencia de que está adaptado a las características de los dispositivos a los que se dirige.

NOMBRE	VERSIÓN	API LEVEL
(No Codename)	1.0	1
(Internally Known As "Petit Four")	1.1	2
Cupcake	1.5	3
Donut	1.6	4
Eclair	2.0 – 2.1	5 – 7
Froyo	2.2 – 2.2.3	8
Gingerbread	2.3 – 2.3.7	9 – 10
Honeycomb	3.0 – 3.2.6	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	14 – 15
Jelly Bean	4.1 – 4.3.1	16 – 18
Kitkat	4.4 – 4.4.4	19 – 20
Lollipop	5.0 – 5.1.1	21 – 22
Marshmallow	6.0 – 6.0.1	23
Nougat	7.0 – 7.1.2	24 – 25
Oreo	8.0 – 8.1	26 – 27
Android P	9	

Tabla con versiones actuales de Android

El núcleo actúa como capa de abstracción entre el hardware y las demás capas. El desarrollador nunca accede directamente al kernel, sino que utiliza las librerías disponibles en capas superiores. Así, el desarrollador no tendrá que conocer las características específicas de cada hardware, ya que será controlador (driver) el que se encargue.

El kernel también gestiona los recursos del teléfono (batería, memoria, almacenamiento, etc.) y del sistema operativo en sí (procesos, elementos de comunicación, etc.).

- **LIBRERÍAS**: Capa justo por encima del kernel, formada por las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta.

  Entre ellas, destacan SQLite (base de datos), SSL (cifrado de comunicaciones), Webkit (navegador web), OpenGL (gráficos).

- **ENTORNO DE EJECUCIÓN**: no es una capa en sí, ya que está compuesta por librerías, por lo que se asocia a la anterior. El componente principal del entorno de ejecución de Android es la máquina virtual ART o Dalvik. ART comenzó a utilizarse en detrimento de Dalvik a partir de la versión 5 Android Lollipop.

  La ejecución de aplicaciones en máquinas virtuales garantiza que los datos de esa aplicación no pueden ser utilizados por ninguna otra. Esta independencia se materializa tanto en los datos de la aplicación como en la memoria de ejecución de la misma.

  Para que dos aplicaciones puedan comunicarse existe los *content provider*, que no son más que contenedores donde se almacena la información que las aplicaciones tienen que compartir. Es necesario ser minucioso en el desarrollo, ya que si no se hace con precaución, el contenedor puede ser accesible por el resto de aplicaciones del dispositivo, lo que supone una grave brecha de seguridad para los datos.

- **FRAMEWORK DE APLICACIONES**: se compone de todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual ART o Dalvik.

- **APLICACIONES**: En la última capa se incluyen todas las aplicaciones del dispositivo, tanto las instaladas, como las del propio sistema. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores. La estructura básica es la siguiente:

Modelo de Seguridad de Android

Todos los procesos en Android proporcionan un entorno seguro de ejecución, en el que ninguna aplicación tiene autorización para realizar una acción que pueda impactar negativamente en la ejecución de otras aplicaciones o del propio sistema. Sólo podrá

saltarse esta restricción mediante la aprobación explícita de un permiso que lo autorice.

Android establece un Modelo de Seguridad bastante estricto, y con el que nos concede un sistema operativo muy seguro, a no ser que el usuario, por propia voluntad o a través de engaños, pueda desactivarlo.

1. Ejecución en Sandbox

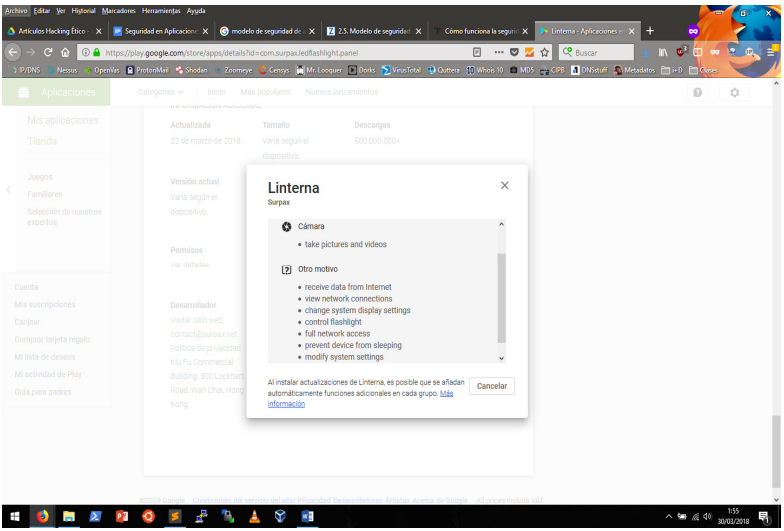
Cada vez que se instala una aplicación se crea un usuario Linux para esta, de forma que una aplicación sólo tendrá acceso a sus propios recursos y no pudiendo interferir directamente en el hardware, es lo conocemos como caja de arena o *sandbox*. Cualquier dato que la aplicación almacene no podrá ser leído por otras aplicaciones, a no ser que la propia aplicación declare que sí se puede hacer.

2. Permisos

En el caso de que una aplicación necesite realizar alguna acción que pueda comprometer la seguridad del dispositivo, debe solicitar autorización expresa a través de los permisos, de forma que el usuario está completamente informado de los riesgos que puede llevar a cabo la acción.

3. Firma

Todas las aplicaciones deben estar firmadas con un certificado digital que identifique al autor. Cada vez que sea modificada la aplicación, ésta deberá ser firmada nuevamente por el autor (propietario de la clave privada).
Esta es la teoría sobre la que se basa la seguridad de Android, pero una vez más rescatamos la genial frase de Kevin Mitnick “el eslabón más débil en la cadena sigue siendo el usuario final”. Vemos constantemente como el usuario instala aplicaciones que solicitan diferentes permisos que no necesita pero, aun así, se les conceden.



Por ejemplo, linternas que solicitan acceso a la lista de contactos, control sobre el sistema, conexiones de red, etc. Con esto prescindimos ya de una de las bases del modelo de seguridad de Android.

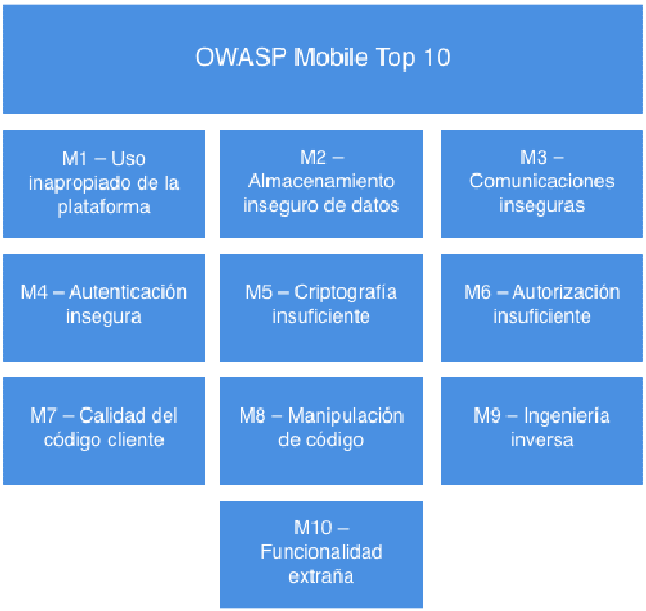
También es frecuente encontrar que se instalan aplicaciones desde fuentes diferentes al repositorio principal, que es Google Play. Instalamos la aplicación aceptando que es de Origen Desconocido, con esto corremos el riesgo de que la aplicación pueda estar modificada con otros fines, y pueda estar poniendo en peligro nuestra privacidad. Aquí estamos saltándonos otro de los 3 principios del modelo de seguridad de Android, en este caso el de la Firma.

Por último, vemos con frecuencia muchos terminales con privilegios de root. El modelo de limitación del acceso a root seguido por los sistemas operativos es un elemento clave en la seguridad, aunque criticado por los defensores de modelos Open Source. Nos permite controlar los accesos a recursos sensibles del sistema, el aislamiento

entre aplicaciones y, en caso de malware, limitar su impacto. El desbloqueo del bootloader (código de arranque del sistema operativo) no es especialmente complejo, y hay miles de tutoriales en internet sobre el procedimiento que hay que seguir para cada dispositivo. Es verdad que obtener el permiso de root nos proporciona poder llevar a cabo determinadas acciones, pero como sabemos de Spiderman, *"un gran poder conlleva una gran responsabilidad"*, por lo que deberemos ser muy precavidos, ya que si alguien accede a nuestro dispositivo, lo hará con el máximo privilegio.

OWASP - Top 10 vulnerabilidades en aplicaciones móviles

La fundación OWASP, de la que os hablamos en varias ocasiones en este libro, elabora diferentes guías para el desarrollo seguro de aplicaciones, tanto web, como móviles o para IoT, además de facilitar diferentes herramientas. En lo referente a Android cabe destacar el Top 10 de vulnerabilidades, que se elabora con información reportada por diferentes empresas de seguridad que colaborar con el proyecto.



A continuación listamos el Top 10 de vulnerabilidades:

- M1 - Uso inapropiado de la plataforma: debemos comprobar la configuración de permisos del archivo AndroidManifiest.xml ya que algunos permisos pueden ser peligrosos (aquellos que

permiten que la app acceda a información confidencial del usuario).

- M2 - Almacenamiento inseguro de datos: la forma correcta de probar este apartado es haciendo uso de la aplicación durante un tiempo, de forma que pueda almacenar datos en disco. Es posible que se necesite un dispositivo Android con acceso root para que se pueda acceder a los archivos en las rutas más usadas como '/sdcard', '/data/data/app_folder', 'sdcard1'.

  Las aplicaciones Android necesitan almacenar datos en local, ya sea en archivos sqlite o en XML, por lo tanto se realizan consultas de entrada/salida. Esto produce dos problemas principales:

  - Inyecciones SQL / XML, y si la lectura es expuesta de forma abierta otras aplicaciones podrían tener acceso a la lectura.

  - La lectura de archivos locales, lo que puede permitir que otras aplicaciones lean archivos de la aplicación y si estos contienen datos sensibles se puede producir una fuga de información.

  Si es una aplicación HTML5 híbrida, también debemos considerar ataques de tipo Cross Site Scripting (XSS). Estos, expondrán toda la aplicación al atacante ya que las aplicaciones HTML5 son basadas de forma nativa en un contenedor WebViews, por lo que llamando a esta funcionalidad nativa se controla toda la aplicación.

  Otra opción para saber que se almacena cuando un usuario interacciona con la aplicación es realizando un backup haciendo uso del comando 'ads backup'.

- M3 - Comunicaciones inseguras: en este punto se pueden realizar diferentes tipos de comprobaciones en el dispositivo Android:

  - Asegurarse que la aplicación navega correctamente
  - Poner un proxy entre la aplicación y el servidor remoto (Man in the Middle). Si falla la carga puede que se esté realizando una validación del certificado.
  - Colocar un Proxy RootCA en la lista de root CA de confianza del dispositivo (podemos hacer uso de Burp o OWASP-ZAP)
  - Probar nuevamente la aplicación. Si todavía no conectase, la aplicación puede estar haciendo un 'certificate pinning' (técnica que usa la clave pública del servidor en el dispositivo para crear un canal seguro entre los dos extremos de la comunicación). Podemos intentar omitir el 'certificate pinning' haciendo uso de Xposed o SMALI.

- M4 - Autenticación insegura: a través del uso de herramientas como ZAP, BURP o Charles como proxy de ataque, o Wireshark para el análisis del tráfico entre el cliente (dispositivo Android) y el servidor, podemos comprobar:
  - El flujo de trabajo y analizar la gestión de sesiones
  - Analizar la API para autenticación
  - WebViews inseguros
  - Credenciales almacenadas en base de datos o a través de servidor
  - Acceso a la cuenta de administrador
  - Verificar la autenticación cuando se llama a los diferentes componentes
  - El timeout de las sesiones
  - Configuración de cookies incorrecta

  - Creación de tokens inseguros
  - Implementación insegura de WebView

- M5 - Criptografía insuficiente: debemos realizar un análisis de forma general para listar donde se ha utilizado cifrado:
  - Tipo de encriptación SSL/TLS usada
  - Obtener archivos de forma segura usando HTTPS URI o un a través de las implementaciones que ofrecen HttpsURLConnection o SSLSocket
  - Autenticación de los tokens de sesión
  - Datos almacenados que contienen información sensible en texto visible
  - Acceso a las claves de encriptación o administración incorrecta de estas
  - Uso de técnicas de criptografía débiles como Rot13, MD4, MD5, RC4, SHA1
  - Implementación de protocolos propios como “hazlo tu mismo” o “yo diseño mi propia encriptación”
  - Clave secreta codificada en el código de la aplicación
  - Uso inseguro de generadores aleatorios

- M6 - Autorización insuficiente: después de comprender cómo funciona la aplicación y cómo trabaja con los datos, se puede verificar el mecanismo de autorización de la forma:
  - Gestión de credenciales: ¿la aplicación hace uso de tokens de autorización en lugar de pedir credenciales todo el tiempo?
  - Comprobar que la aplicación sólo permite acceso a los roles autorizados.
  - Almacenar el nombre de usuario y la contraseña como cualquier otro dato en vez de usar AccountManager.

- M7 - Calidad del código cliente: hay dos enfoques para esto:
    - Tenemos acceso al código fuente, por lo que podemos hacer una revisión del código de la aplicación y la API del servidor
    - No tenemos acceso y tenemos que comprobar el código descompilando el APK
- M8 - Manipulación del código: se necesita un dispositivo con acceso root y técnicas de ingeniería inversa:
    - Descompilar el APK usando herramientas como apktool, dex2jar o enjarify
    - Analizar el código usando un descompilador como JD-GUI o Bytecodeviewer
    - Después de analizar el código, intentar saltarse las funcionalidades bien sea cambiando el código Smali o mediante métodos de hooks usando frameworks como Xposed o Frida
    - Comprobar si la aplicación ha sido ofuscada y verificar el nivel de la ofuscación buscando cadenas de texto específicas
- M9 - Ingeniería inversa: es una parte fundamental de las pruebas que se realizan a aplicaciones móviles y se debe verificar que:
    - La aplicación ha sido ofuscada
    - Buscar cadenas de textos con claves con herramientas como Bytecodeviewer o JEB.
    - Buscar la implementación de SSL pining, accesos como root o conexiones a la API (podemos buscar palabras como 'TrustManager' , 'SHA256', X509 ,SHA, SSL).
- M10 - Funcionalidad extraña: para comprobar estos comportamientos se necesita tener acceso al código, ya sea de forma directa o mediante ingeniería inversa del APK.

OWASP y ENISA - Top 10 controles de seguridad

La fundación OWASP y ENISA, la Agencia de Seguridad de las Redes y de la Información de la Unión Europea (European Network and Information Security Agency), han publicado de forma conjunta una guía para el desarrollo seguro en dispositivos móviles, listas en un Top 10 de los controles de seguridad más importantes que centraremos en aplicaciones Android.

Además del Top 10, la guía ofrece otros consejos entre los que destacan:

- Ejecutar aplicaciones con el mínimo privilegio necesario en el sistema operativo. Hay que tener en cuenta los otorgados por las API y desactivarlos.
- No autorizar la ejecución de código o de la app con privilegios de administrador o root.
- Asegurarse de que el login o registro es realizado de forma adecuada y que no almacena logs en exceso, especialmente información sensible del usuario.

En cuanto a los controles, la lista incluye:

1. Identificar y proteger datos sensibles en el dispositivo
2. Gestionar las credenciales de forma segura
3. Asegurarse que los datos sensibles son protegidos cuando están en tránsito entre las diferentes partes y componentes.
4. Implementar una correcta gestión de la autenticación, autorización y gestión de sesiones del usuario.
5. Mantener los servicios (API) y la plataforma (servidor) seguros.

6. Integración de datos segura con servicios y aplicaciones de terceros.
7. Consentimiento para recopilar, almacenar y usar datos del usuario.
8. Controlar los intentos de acceso sin autorización a los servicios de pago electrónico (wallet, SMS, llamadas, etc.)
9. Mantener sistema y aplicación actualizada.
10. Proteger la aplicación en tiempo de ejecución.

Referencias Bibliográficas

https://developer.android.com/guide/index.html

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project_-_Android

https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide

https://www.owasp.org/index.php/Android_Testing_Cheat_Sheet

https://www.enisa.europa.eu/news/enisa-news/top-ten-smartphone-security-controls-for- developers

Seguridad en Tráfico de Red

Ramón Salado Lucena

Seguridad en tráfico de red (802.3 y 802.11)

El **IEEE** (Institute of Electrical and Electronics Engineers, 1963) es una asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas (aeroespacial, telecomunicaciones, biomedicina, energía, etc.). El IEEE se compone aproximadamente de 425.000 miembros de más de 150 países, entre sus fundadores cuenta con celebridades como Thomas Alva Edison o Alexander Graham Bell entre otros.

Las áreas de conocimiento las divide en Sociedades o Comités, una de las más conocidas es la que tratamos ahora, la 802 que se centra en el desarrollo de estándares de redes de área local (LAN) y redes de área metropolitana (MAN), principalmente en las dos capas inferiores del Modelo OSI (física y de enlace).

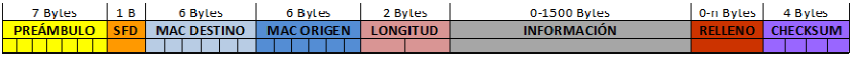
En la siguiente tabla se pueden ver los diferentes grupos de trabajo, dentro del estándar 802:

IEEE 802.1	Protocolos Superiores Área Local (LAN)
IEEE 802.2	Control de enlace lógico LLC
IEEE 802.3	CSMA / CD (ETHERNET)
IEEE 802.4	Token bus LAN
IEEE 802.5	Token ring LAN (topología en anillo)
IEEE 802.6	Redes de Área Metropolitana (MAN) (ciudad) (fibra óptica)
IEEE 802.7	Grupo Asesor en Banda ancha
IEEE 802.8	Grupo Asesor en Fibras Ópticas
IEEE 802.9	Servicios Integrados de red de Área Local
IEEE 802.10	Seguridad de red
IEEE 802.11	Redes inalámbricas WLAN (Wi-Fi)
IEEE 802.12	Acceso de Prioridad por demanda 100 Base VG-Any Lan
IEEE 802.13	Se ha evitado su uso por superstición
IEEE 802.14	Módems de cable
IEEE 802.15	WPAN (Bluetooth)
IEEE 802.16	Redes de acceso metropolitanas sin hilos de banda ancha (WIMAX)
IEEE 802.17	Anillo de paquete elástico script
IEEE 802.18	Grupo de Asesoría Técnica sobre Normativas de Radio
IEEE 802.19	Grupo de Asesoría Técnica sobre Coexistencia
IEEE 802.20	*Mobile Broadband Wireless Access*
IEEE 802.21	Media Independent Handoff
IEEE 802.22	*Wireless Regional Area Network*
IEEE 802.23	Emergency Services Working Group
IEEE 802.24	Vertical Applications TAG (Smart Grid, IoT)

IEEE 802.3 ETHERNET

Creado en 1973, en los laboratorios de la mítica **Xerox PARC** (Palo Alto Research Center), a partir del sistema de comunicación por radio llamado **ALOHA**. Se trabaja sobre la idea de que las estaciones antes de transmitir deberían detectar si el canal ya estaba libre o en uso, en cuyo caso esperarían a que la estación activa terminara. También, cada estación vigilaría el medio físico durante cada transmisión, por si se detectara alguna colisión, en este caso se detendría la comunicación y se retransmitirá más tarde. Este protocolo MAC pasaría más tarde a denominarse Acceso Múltiple con Detección de Portadora y Detección de Colisiones, o **CSMA/CD** (*Carrier Sense Multiple Access / Collision Detection*).

Desde su creación hasta hoy, Ethernet ha sufrido algunas modificaciones. El estándar original definió un tamaño mínimo de trama en 64 bytes y un máximo de 1518 bytes. El estándar IEEE 802.3ac, publicado en 1998, amplió el tamaño de trama máximo permitido a 1522 bytes.



Coexisten diferentes tecnologías Ethernet que se diferencia por algunos conceptos:

- Cableado: Tecnología del nivel físico que usa la tecnología.
- Velocidad: Velocidad de transmisión.
- Alcance: Distancia máxima entre dos nodos.

Nombre	Tipo	Velocidad	Cableado	Distancia
Ethernet	10BaseT	10 Mbit/s	Par Trenzado	100 m
Fast Ethernet	100BaseTX	100 Mbit/s	Par Trenzado (Cat. 5UTP)	100 m
	100BaseFX	200 Mbit/s	Fibra óptica	2000 m
Gigabit Ethernet	1000BaseT	1000 Mbit/s	Cat. 5e UTP	100 m
	1000BaseTX	1000 Mbit/s	Cat. 6UTP	100 m
	1000BaseSX	1000 Mbit/s	Fibra óptica (multimodo)	550 m
	1000BaseLX	1000 Mbit/s	Fibra óptica (monomodo)	3000 m
Ethernet 10 Gigabits	10GBase-T	10Gbits/s	Cat. 6A y Cat. 7 UTP	100m
	10GBase-LX	10Gbits/s	Fibra óptica (multimodo)	100m
	10GBase-LX	10Gbits/s	Fibra óptica (monomodo)	10Km

Es necesario que comprendamos la dificultad de implementar seguridad sobre protocolos desarrollados más de 40 años, que aunque han ido actualizándose, no se desarrollaron inicialmente teniendo en cuenta la seguridad. Actualmente, todos los que trabajamos en seguridad defensiva, nos encontramos en la tesitura de construir una muralla cada día, lo suficientemente fuerte como para proteger nuestro sistema, tenemos que velar constantemente por su invulnerabilidad y consistencia, mientras que un atacante, tan sólo necesita una pequeña rendija por donde acceder.

Siguiendo el enfoque clásico de la seguridad en sistemas informáticos, podemos generalizar agrupando los ataques en tres tipos:

- Ataques Sintácticos: aquellos que van dirigidos con la lógica de los equipos y redes, con la intención de explotar las vulnerabilidades del software o los protocolos implementados.
- Ataques Semánticos: los que intentan aprovechar la mayor vulnerabilidad de los sistemas informáticos, el usuario. La Ingeniería Social ha evolucionado con la tecnología, y los ataques han pasado de ser sobre teléfonos hasta los Smartphone de nuestros días.
- Ataques físicos: se producen contra la propia infraestructura física: equipos, servidores, cableado, etc.

Como comentábamos en la introducción, el estándar IEEE 802 se enfoca fundamentalmente en las capas Física y de Enlace del modelo OSI. A continuación describiremos cada capa haciendo énfasis en sus vulnerabilidades.

- **CAPA FÍSICA**: es la más baja del modelo OSI, se encarga de la transmisión y recepción de una secuencia no estructurada de bits sin procesar a través de un medio físico (cable, UTP, fibra óptica, conexión inalámbrica, etc.). Proporciona los medios mecánicos, eléctricos, funcionales y de procedimiento para activar, mantener y desactivar conexiones físicas. Encontramos protocolos como Ethernet, PPP, Frame Relay, etc.

Las principales vulnerabilidades de esta capa son:

- Pérdida de energía
- Robo físico de datos o hardware
- Alteraciones no autorizadas en la infraestructura

Es necesario tomar medidas de mitigación, tanto de personal externo como interno. Instalaciones con control de acceso a las zonas vitales de la infraestructura, sistema de alarmas y/o cámaras de seguridad, cerraduras sobre los racks, sistemas de alimentación ininterrumpida, protección de sobrecargas eléctricas, balanceado de conexiones, etc.

- **CAPA DE ENLACE**: es responsable de la transferencia fiable de información a través de un circuito de transmisión de datos. Recibe peticiones de la capa de Red y utiliza los servicios de la capa Física. Su objetivo es conseguir que la información circule, sin errores, entre dos máquinas que estén conectadas. Para lograrlo, tiene que montar bloques de información (tramas), dotarlos de una dirección de capa de enlace, MAC (Media Access Control, 48 bits, en 6 bloques de dos caracteres hexadecimales), y gestionar la detección o corrección de errores. Cuando el medio de comunicación está compartido entre más de dos equipos es necesario arbitrar su uso, esto se hace desde la subcapa de Control de Acceso al Medio. En esta capa encontramos un dispositivo muy importante para la seguridad, los Switches.

  Las principales vulnerabilidades de la capa de enlace son:

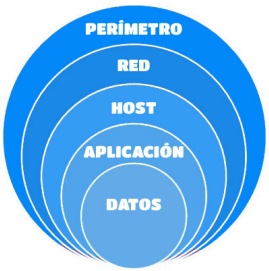
  - Suplantación de direcciones MAC (MAC spoofing)
  - Segmentaciones de la red
  - Configuraciones inalámbricas
  - Ataques a STP

  En esta capa encontramos un elemento muy importante para la seguridad en la red, las redes virtuales o VLAN´s. Son segmentaciones de la red que permiten la separación del tráfico en diferentes capas, por ejemplo, aislando las

conexiones para administrar la red, de las de los propios usuarios.

Tomaremos medidas para evitar la suplantación de direcciones MAC, con esta técnica se pueden causar graves daños en la infraestructura, por lo que es necesario una vigilancia. La suplantación de la dirección MAC posibilita a un atacante la escucha del tráfico de red, por lo que podría obtener información del tráfico del resto de los equipos. Técnicas clásicas como "*Man in the Middle"* se basan en esta vulnerabilidad, que permite envenenar la tabla ARP y hacer creer al resto de equipos de la red, que su dirección es el router, por lo que todo el tráfico pasaría por el atacante. Destacar nuevamente la importancia de la segmentación de la red y del uso de dispositivos switch que permitan detectar la suplantación de MAC, con tecnologías "*Port Security*", característica que permite forzar al switch a permitir sólo una dirección MAC para cada puerto físico en el switch, lo que impide que alguien cambie la dirección MAC de su equipo o que trate de usar más de una dirección a la vez.

Defensa en Profundidad (Defense in Depth)
Es un modelo de arquitectura para garantizar unos niveles de seguridad aceptables en nuestros sistemas e infraestructuras. Consiste en aplicar diferentes controles de seguridad, con el fin de frenar la actividad de un usuario o software malicioso. De esta manera, la explotación de una vulnerabilidad, no comprometería a toda la organización, tan sólo a su capa. La estructura genérica es la siguiente:



De una forma más concreta, y siguiendo el esquema de MSAT (Microsoft Assessment Tool):

1. **Router:** Primera barrera, configurando una ACL podremos permitir o denegar el tráfico.

2. **Firewall**: Representa la línea inicial de defensa. Las reglas aplicadas serán restrictivas y deberán establecerse por host y servicio. Utilizar DMZ si es preciso.

3. **IDS/IPS**: sistemas de detección de intrusiones, ya sea de forma activa o pasiva, en función de las necesidades.

4. **Antivirus/Anti Espías:** Solución que controle el malware tanto a nivel de servidores como de clientes. También soluciones que filtren los buzones de correo electrónico.

5. **Redes Privadas Virtuales (VPN):** Utilizar VPN en el perímetro de la red y en las zonas que necesiten una mayor seguridad. Utilizarlas también para asegurar protocolos que sean transmitidos en texto plano, sin cifrado. Para usuarios remotos utilizar factor múltiple de autenticación y protocolos como IPSEC, SSL y SSH, para acceder a la red corporativa. Auditar periódicamente el registro de las conexiones.

6. **Segmentación:** Separación de la red en diferentes subredes (VLAN´s)

7. **Contraseñas:** Es necesario crear políticas estrictas para las contraseñas, tanto las cuentas de Usuarios estándar, como en las de administrador (más aún si cabe). La política debe recoger un mínimo de caracteres, la duración y el procedimiento para cambiarla.

8. **Conductas de Administración:** Utilizar terminales seguros para los trabajos de administración de la red, y conexiones seguras como SSH o VPN. Verificar que la monitorización se haga con SNMP v3.

9. **Diario de actuaciones:** Mantener un historial de las actuaciones realizadas en lo relativo al mantenimiento, la seguridad y el funcionamiento de los sistemas. Limitar el acceso a estos ficheros, y hacer revisiones periódicas de las tareas.

10. **Actualizaciones y Parches:** Comprobaciones periódicas de actualizaciones de seguridad y de boletines de vulnerabilidades.

La implementación de este tipo de modelos, se ha simplificado con la popularización de herramientas UTM (*Unified Threat Management*) o Gestión Unificada de Amenazas. Se trata de soluciones que integran las funcionalidades de Firewall, IDS/IPS, Antivirus, Antiphishing, Antispyware, VPN, Log Manager, etc.

Otra herramienta que cada vez toma mayor trascendencia son los SIEM (*Security Information and Event Management*), que correlacionan los logs de los diferentes elementos de la red y los eventos de seguridad, para detectar las amenazas en tiempo real. Por último, los sistemas ERD (Detección y Respuesta de *Endpoints*) proporcionan un alto nivel de seguridad a los equipos cliente de la red.

EEE 802.11 RED DE ÁREA LOCAL INALÁMBRICA (WiFi)

La primera versión fue publicada en 1997, sus especificaciones proporcionan la base para los productos con redes inalámbricas, que hacen uso del certificado Wi-Fi. Dentro de la rama 802.11 se definen diferentes tecnologías, las principales son:

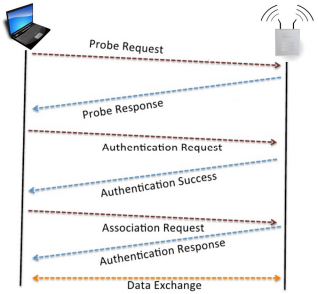
IEEE	Frecuencia	Velocidad Máxima	Distanc.	Modulación	Máx. Potencia Transmis.
802.11	2.4 GHz	2 Mbps	20 m	DSSS, FHSS	100 mW
b	2.4 GHz	11 Mbps	35 m	CCK, DSSS	100 mW
a	5 GHz	54 Mbps	35 m	OFDM	100 mW
g	2.4 GHz	54 Mbps	70 m	DSSS, OFDM	100 mW
n	2.4 GHz, 5 GHz	600 Mbps	70 m	OFDM	100 mW
ac	5 GHz	6.93 Gbps	35 m	OFDM	160 mW
ad	60 GHz	6.76 Gbps	10 m	SC, OFDM	10 mW
af	54-790 MHz	26.7 Mbps	>1km ?	SC, OFDM	100 mW
ah	900 MHz	40 Mbps	1 km	SC, OFDM	100 mW

Las características de las redes WiFi dependerán de las necesidades que tengamos que cubrir. El elemento fundamental de las redes WiFi es la celda, definida como el área en la que varios dispositivos se interconectan entre sí por medio aéreo.

Fundamentalmente existen dos tipos de arquitecturas en estas redes:

- Red Ad-hoc: compuesta por ordenadores (en principio), que conforma una sola celda aislada, sin posibilidad es unirse a otro tipo de celdas.
- Modo Infraestructura: utiliza un dispositivo de interconexión (Punto de Acceso) como nexo de unión entre la red cableada y la inalámbrica, lo que permite crear diferentes celdas que trabajarán conjuntamente. Este tipo es el predominante y será en el que nos centremos.

Fases de la conexión



Escaneo de Redes Inalámbricas

Los Puntos de Acceso (AP) envían periódicamente paquetes denominados "*Beacon Frames*" (Tramas Baliza), para anunciar la red. Estas balizas contienen, entre otros datos, la Dirección MAC y el nombre de la red (SSID, *Service Set Identifier*), el canal donde se encuentra, el cifrado configurado, etc. Este tipo de detección es el Escaneo Pasivo.

En el Escaneo Activo, es el cliente el que intenta localizar un Punto de Acceso (AP) que esté transmitiendo, lo hace a través de un paquete *Probe Request Frames*, y queda a la espera de un *Probe Request Frame* del AP. El *Probe Request Frame* tiene una estructura parecida a los *Beacon Frames*.

Sincronización

Es un proceso necesario para mantener a los clientes acompasados. Los *Beacon Frames* emitidos por el AP contienen un *timestamp*, que usarán los clientes para su sincronización.

Autenticación

La conexión de cada cliente necesita ser autenticada por el AP antes de darle acceso a la red. Encontramos dos tipos de Autenticación: Sistema abierto o Clave compartida.

Al tratarse de un medio compartido (el aire), cualquiera puede intentar conectarse sin necesidad de ningún medio físico, lo que supone una gran exposición. Para ello, existen diferentes medios de

cifrado, que protegen a los dispositivos conectados de cualquier intento de captura del tráfico:

- **Sistema abierto**: sin cifrado, totalmente descartado para utilizar.
- **WEP** (*Wired Equivalent Privacy*): método en desuso debido a su debilidad, a pesar de sus evoluciones, continúa siendo un sistema muy débil y fácil de explotar. Claves de 64 ó 128 bits.
- **WPA-PSK (TKIP)** (*WiFi Protected Access*): las claves utilizadas son de 256 bits. Fue creado para corregir las carencias del sistema WPE. Actualmente tampoco es seguro, existen diferentes métodos para explotarlo.
- **WPA-PSK (AES)**: integra un sistema de cifrado más moderno (AES). Todos los dispositivos que soportan AES, también incluyen WPA2, por lo que es recomendable utilizar este último sistema, que es más actual y más robusto.
- **WPA2-PSK (TKIP)**: utiliza el estándar WPA2, mejora del WPA, con cifrado TKIP. Esta opción no es segura, pero, una buena opción si existen dispositivos antiguos que no soportan una red WPA2-PSK (AES), aunque la mejor sería WPA2-PSK (AES+TKIP).
- **WPA2-PSK (AES)**: es el mayor nivel de cifrado que puedes elegir. Aunque se existen diferentes técnicas de ataque sobre redes WPA2 (KRACK, *Key Reinstallation Attacks*, año 2017) que permite capturar parte del tráfico, es el método más seguro de los que existen y podemos mitigarlo con el uso de SSL/TLS y VPN´s.
- **WPA2-PSK (AES+TKIP)**: es la mejor elección si tienes algunos dispositivos que no se puedan conectar por WPA2-PSK AES.
- **WPA3**: la Alianza Wi-Fi ya está trabajando en el sucesor de WPA2, durante 2018 tendremos novedades al respecto.

Asociación

Habilita la transferencia de datos entre el cliente y el AP. Una vez autenticado el cliente, éste manda un *Association Request* al AP y él le responde con un *Association Response*. Si la conexión es exitosa, el AP entrega un identificador al cliente (*ID*), y lo agrega a su base de datos de clientes conectados.

Transferencia de Datos

Se permite una vez se han pasado los procesos de Autenticación y Asociación. Si se envían datos a un AP sin autenticación o asociación, el AP responderá con un *DeAuthentication Frame*.

Seguridad Redes WiFi

Una red inalámbrica no está limitada a un área concreta, a un cable o fibra, si no que se expande libremente dentro de su radio de cobertura. Su principal riesgo es por tanto, que un cliente no autorizado se conecte a la red y pueda acceder a los datos. Pero no es el único riesgo, también es posible que nuestra red sea bloqueada por alguna interferencia malintencionada que bloquee la señal original, lo que puede dar lugar a una suplantación de la red. Los usuarios podrían llegar a conectarse a la red maliciosa al confundirla con la real, y compartir datos como credenciales de acceso, o cualquier otra información sensible.
La seguridad nunca es absoluta, pero es recomendable implementar unas normas básicas en su configuración:

- Modificar las configuraciones por defecto. Nombres de usuarios, nombre de la red, contraseñas, rangos del DHCP, etc.
- Habilitar el nivel máximo de cifrado. (lo comentamos anteriormente).
- Utilizar contraseñas robustas, combinando caracteres alfanuméricos y símbolos y mayúsculas y minúsculas. Evitar palabras comunes que puedan aparecer en cualquier diccionario o elementos fácilmente identificables como fechas de nacimiento, nombres, etc. Cambio periódico de contraseñas.

- Desactivar el anuncio del nombre de la red (SSID), seguridad a través de la oscuridad. No impide que un atacante pueda conseguir el nombre de la red posteriormente, al menos dificultamos su tarea.
- Uso de direcciones IP estáticas y DHCP en otra trama diferente. Al conectarnos de forma maliciosa, el DHCP nos dará una dirección IP de otra trama diferente a la de la red de trabajo, por lo que dificultamos que el atacante conozca en qué trama trabajamos. También podemos desactivar el servidor DHCP.
- VLAN propia para la red WIFi, para mantener aislada la red inalámbrica de la red de trabajo.
- Utilización de servidor Radius (*Remote Access Dial In User Service)*, es un protocolo que ofrece un mecanismo de autenticación y autorización, y una administración simplificada de las credenciales de acceso a un recurso de red.
- Desactivar WPS (WiFi Protected Setup) este estándar tiene una vulnerabilidad y no es complejo obtener la contraseña para la conexión.
- Dispositivos con versiones actualizadas de firmware y actualizaciones en los diferentes equipos clientes de la red.

Principales ataques WiFi

- **KRACKs** (*Key Reinstallation AttaCKs*): permite al atacante "acceder a la información que hasta ahora se asumía que estaba cifrada de forma segura" pudiendo ser usada la técnica para acceder al router y robar datos personales que naveguen por la red. Se ejecuta sobre WPA2, hasta este descubrimiento, el protocolo más seguro de los existentes.

  No es una vulnerabilidad de un dispositivo concreto, si no del propio protocolo, por lo que para su mitigación será necesario que se actualice el firmware de los diferentes elementos de la red. En el caso de que el fabricante no publique el parche, la recomendación pasa por utilizar medidas de seguridad adicionales como HTTPS o VPN.

- **WiFi Honey**: es un script que crea diferentes interfaces de red en modo monitor, para ser usadas como puntos de acceso simulando una red real, buscando que los usuarios legítimos se conecten a alguno de estos puntos de acceso, dejando la clave de la red.

- **Ataque por diccionario**: La primera fase de la prueba consiste en la captura del Handshake, que no es más que el primer paquete que envía un dispositivo cuando la conexión es correcta. La forma de conseguirlo es con un ataque de desautenticación sobre un cliente legítimo de la red, cuando vuelva a conectarse podremos capturar el Handshake. Una vez capturado, utilizaremos un diccionario de contraseñas para compararlo con el Handshake y obtener la contraseña. La tarea dependerá del tamaño del diccionario, los hay desde varios Mb hasta más de 10 Gb, obviamente, si la contraseña no está en el diccionario, no la obtendremos por esta vía.

- **Ataque por fuerza bruta:** Tras la captura del Handshake como en el caso anterior, utilizamos un sistema que pruebe diferentes combinaciones hasta hallar la contraseña de la red. En función de la dificultad y de las características técnicas de nuestros equipos, se podrá obtener más o menos pronto. Se suelen utilizar frameworks como CUDA o Pyrit para poder utilizar toda la potencia de cómputo del hardware, de forma que podamos calcular el mayor número posible de claves por segundo.

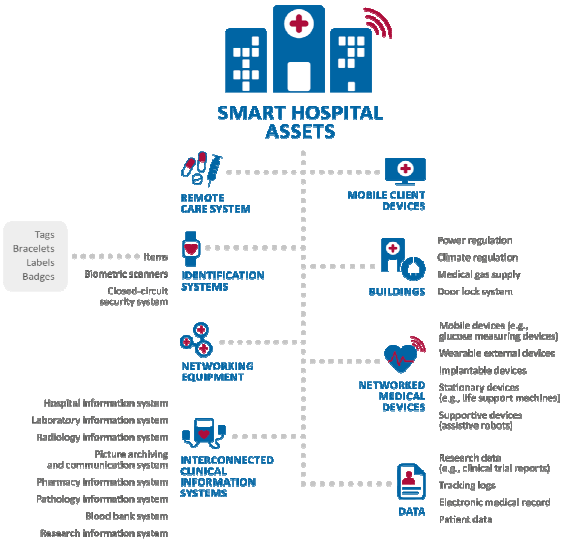
La presencia de un atacante dentro de nuestra red, ya sea por acceso WiFi o infectando cualquier equipo conectado por Ethernet, supone una importante brecha de seguridad en una infraestructura. Mayor aun es la gravedad en el ámbito sanitario que tratamos, por lo espinoso de los datos almacenados y que circulan. El intruso podría tener acceso a datos confidenciales de pacientes, historiales clínicos, incluso a determinado instrumental médico conectado a la red.

Dispositivos IoT en la Red

Los dispositivos *IoT* no son ajenos a los peligros citados durante el capítulo, en definitiva, el Internet de las Cosas hace uso de tecnologías de Redes, Cloud, Móvil y de Aplicaciones, todas ellas con

sus diferentes vulnerabilidades. La mayoría de dispositivos (70% según un estudio de HP) no utiliza ningún sistema de cifrado en sus comunicaciones, ni permiten múltiple factor de autenticación, y además suelen tener diferentes puertos a la escucha y protocolos visibles desde internet (UPnP por ejemplo).

Debemos considerar el ámbito sanitario como una variante de los Sistemas de Control Industrial, debido al gran número de dispositivos conectados, cada uno de ellos con una serie de protocolos distintos. Conceptos como *Smart Hospital* implican un mayor número de dispositivos IoT conectados, por lo que es vital comprender como funciona el ecosistema, el tipo de información que trata y cómo lo hace, para poder blindarlo con una capa de seguridad.



Activos de un Smart Hospital (Fuente: ENISA)

En este tipo de infraestructuras es ineludible utilizar un diseño con zonas segmentadas y mecanismos de control del tráfico entre los diferentes segmentos, como un primer paso para la implantación de una arquitectura de red segura. Para ello, es necesario identificar los segmentos en función de su cometido. La Norma ISA-95 propone el modelo "*Purdue Enterprise Reference Architecture*" en el que se establecen diferentes niveles lógicos en los que se agruparán los

dispositivos conectados. Como mínimo se deberían diferenciar tres segmentos separados: Red de Control, DMZ (desmilitarizada) y Red Corporativa. De esta forma, al explotarse cualquier vulnerabilidad en uno de los dispositivos, el ataque no sobrepasaría el segmento donde se encuentra conectado.

Entre los diferentes segmentos de la red es necesario que exista comunicación, pero ésta debe ser cifrada, para evitar que un atacante pueda ver y manipular el contenido. Por ello, el uso de HTTPS, SSH y SNMP v3 es altamente recomendable.

Además, cada vez es más usual el almacenamiento de resultados de pruebas de diagnosis en la nube, para que el personal sanitario y el paciente, desde diferentes aplicaciones web o móviles, tengan acceso a ellos. Protocolos como **PACS** (*Picture Archiving and Communication System*) o **DICOM** (*Digital Imaging and Communication in Medicine*) están bastante extendidos para estas funciones. El problema viene con las aplicaciones web o móviles que consultan estos datos, ya que se ha demostrado en diferentes pruebas, que no son seguras (puede consultar el capítulo de Seguridad en Aplicaciones Web para ampliar información). Un reciente análisis de seguridad de la empresa Eleven Paths sobre las 35 apps PACS y DICOM más utilizadas, ha revelado un gran número de errores de seguridad en la mayoría de ellas, inyecciones SQL, almacenamiento inseguro, contraseñas almacenadas en el código (*hardcode*), falta de cifrado, etc.

Tampoco podemos olvidar los dispositivos **IMD** (*Implantable Medical Devices*) son dispositivos electrónicos implantados dentro del cuerpo para tratar una condición médica, monitorizar el estado o mejorar el funcionamiento de alguna parte del cuerpo, por ejemplo, marcapasos, neuroestimuladores, bombas de infusión o biosensores. Transmiten información muy sensible, y un ciberataque podría ser letal para el paciente. Por lo que será preciso asegurar el entorno de estos dispositivos y sus conexiones.

Referencias Bibliográficas

1. IEEE 802: https://standards.ieee.org/findstds/standard/802.11-2016.html

2. CCN-CERT, Ciberamenazas y Tendencias 2017: https://www.ccn-cert.cni.es/pdf/informes-de-ciberseguridad-ccn-cert/informes-ccn-cert-publicos/2224-ccn-cert-ia-16-17-ciberamenzas-y-tendencias-edicion-2017/file.html
3. ENISA, Cyber security and resilience for Smart Hospitals: https://www.enisa.europa.eu/publications/cyber-security-and-resilience-for-smart-hospitals/at_download/fullReport
4. INCIBE, Protocolos y Seguridad de red en infraestructuras SCI: https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuia/incibe_protocolos_seguridad_red_sci.pdf
5. SANS Institute InfoSec Reading Room: https://www.symantec.com/content/dam/symantec/docs/reports/proactive-approach-incident-response.pdf

6. Security Guidance for Critical Areas of Focus in Cloud Computing 4.0: https://downloads.cloudsecurityalliance.org/assets/research/security-guidance/security-guidance-v4-FINAL.pdf

7. Eleven Paths, Seguridad web en aplicaciones DICOM Viewers: http://blog.elevenpaths.com/2017/11/seguridad-aplicaciones-dicom-elevenpaths.html

8. Security and Privacy Issues in Implantable Medical Devices: https://www.sciencedirect.com/science/article/pii/S1532046415000074X

9. HP, How safe are home security systems?: http://files.asset.microfocus.com/4aa5-7342/en/4aa5-7342.pdf

Seguridad en BLE y ZigBee

Juan José Domenech Sánchez

BLUETOOTH LOW ENERGY (BLE)

Introducción

Bluetooth Low Energy (BLE) comenzó como parte de la especificación del Core de Bluetooth 4.0. BLE se presenta como más pequeño, más optimizado que la versión del clásico Bluetooth, pero, en realidad, BLE tiene un enfoque y diseño totalmente diferente.

Desde sus comienzos, su diseño se centró en crear un standard de radio con el menor consumo de energía, de bajo coste, bajo ancho de banda y poca complejidad.

Comparado con otros estándares Wireless, BLE ha tenido una rápida adopción debido a que está íntimamente relacionado con smartphones, tablets y otros dispositivos móviles. Además, ha ganado impulso al ser adoptado por grandes empresas de la industria móvil como Apple y Samsung.

Apple en particular ha realizado un esfuerzo para producir un stack BLE y publicar guías de diseños en torno a esta tecnología. Esto generó una cadena de reacciones en el mercado de proveedores de dispositivos que vieron la apuesta de Apple por BLE como tecnología triunfadora a largo plazo.

Así, podemos encontrar soluciones completas por menos de 2€, lo cual es mucho menor que el precio general del resto de tecnólogas inalámbricas como WiFi, GSM, Zigbee, etc.

Uno de los factores que más ha contribuido al éxito de BLE y menos conocido es que fue diseñado para servir como framework para el intercambio de datos, a diferencia de Bluetooth clásico que sólo se centró en un estricto conjunto de datos.

De esta forma, BLE fue concebido para alguien con una idea y un conjunto de datos provenientes de un accesorio pudiera

intercambiarlos sin tener que conocer toda la tecnología subyacente, dándose un gran paso de abstracción a la vez que se crea un posible punto de inseguridad: se desconoce cómo tratar de forma segura la información y las comunicaciones.

La especificación

En 2010 Bluetooth SIG introduce Bluetooth Low Energy con la versión 4.0 del Bluetooth Core Specification. En esta primera versión se producen diferentes polémicas en su desarrollo que no son resueltas hasta la actualización Bluetooth 4.1 de 2013, que se convierte en el referente para todo aquel que quiera desarrollar productos BLE.

Como en todas las especificaciones Bluetooth, la versión 4.1 es compatible con la 4.0, asegurándose de la interoperabilidad entre los dispositivos que implementan diferentes versiones de la especificación.

En 2014 se publicó la especificación 4.2 y en 2016 la versión 5.0, siendo la más adoptada en el mercado la versión 4.2 y sobre la que nos basaremos en los siguientes puntos.

Para obtener más información sobre las últimas versiones de la especificación Bluetooth https://www.bluetooth.com/specifications

Soporte entre especificaciones

La especificación Bluetooth abarca la clásica y la BLE. Estos dos estándares de comunicación no son directamente compatibles: el protocolo on-air, las capas superiores de protocolo y las aplicaciones son diferentes e incompatibles entre ambas tecnologías.

La tabla siguiente muestra los 3 tipos de dispositivos que podemos encontrar en el mercado:

Dispositivo	**Soporte BR/RDR (Bluetooth Clásico)**	**Soporte BLE**
Pre 4.0 Bluetooth	SI	NO
4.x Single-Mode (Bluetooth Smart)	NO	SI
4.x Dual-Mode (Bluetooth Smart Ready)	SI	NO

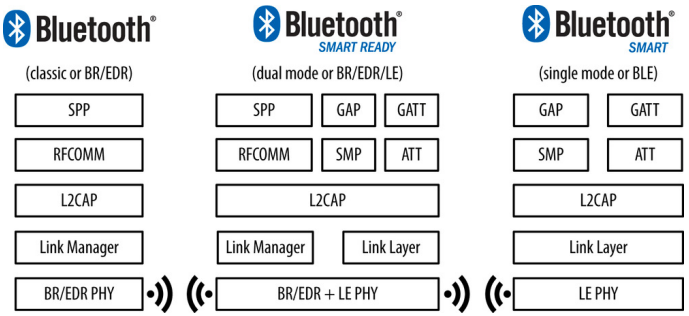
Como vemos, se definen 2 tipos de tecnologías Bluetooth:

- **BR/EDR (Bluetooth clásico):** el estándar sobre el que se ha desarrollado la especificación Bluetooth desde la versión 1.0
- **BLE (Bluetooth Low Energy):** el estándar Wireless de bajo consumo introducido con la versión 4.0.

Y cada tecnología puede usar estas configuraciones:

- **Single-mode (BLE, Bluetooth Smart):** un dispositivo que implementa BLE el cual puede comunicarse con otros dispositivos single-mode y dual-mode, pero no con los que únicamente soportan BR/EDR.
- **Dual-mode (BR/EDR/LE, Bluetooth Smart Ready):** un dispositivo que implementa tanto BR/EDR como BLE, el cual puede comunicarse con cualquier dispositivo Bluetooth.

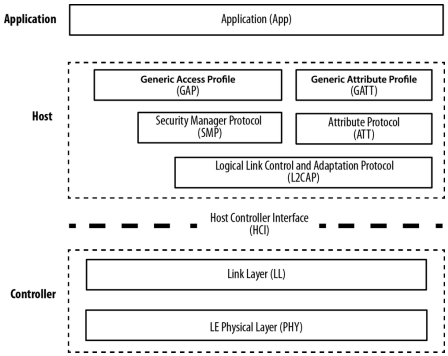
En la siguiente figura podemos ver estas posibilidades de configuración:



A medida que aumenta el número de sensores BLE de tipo single-mode también aumenta el número de dispositivos BR/EDR que soportan BLE. Una característica cada vez más común es que estos dispositivos dual-mode sean capaces de reenviar a internet los datos obtenidos de un dispositivo BLE single-mode utilizando sus conexiones GSM o WiFi.

El protocolo BLE

En la siguiente figura podemos ver las partes de un dispositivo BLE single-mode.



Donde vemos que se divide en 3 partes principales: controlador, host y aplicación. A continuación, veremos cada una de ellas.

CONTROLADOR

Incluye las siguientes capas:

1. Physical Layer (PHY) o capa física

Es la parte que contiene los circuitos analógicos, modulando y desmodulando señalas analógicas y transformándolas en digitales. Se usa la banda 2,4 GHz ISM (Industrial, Scientific and Medical) y se divide esta banda en 40 canales desde 2.4000 GHz hasta 2.4835 GHz.

Para evitar interferencias con las señales en la misma banda, como serían WiFi y Bluetooth clásico, se usa una técnica denominada *salto de frecuencia de amplio espectro,* en la cual la radio salta entre canales en cada evento de la conexión.

2. **Link Layer (LL) o capa de enlace**

Se comunica directamente con la capa PHY y normalmente es implementada con una combinación personalizada de hardware en el que se realizan las funcionalidades más simples (preámbulos, dirección de acceso, CRC generación y verificación, generación de números aleatorios, encriptación AES) y software (estado de los enlaces o como se conecta a otros dispositivos).

Los dispositivos BLE pueden ser master, slave o ambos, dependiendo de los casos de uso y requerimientos. Aquellos que inicializan la conexión serán master (los Smartphone y tablets tienen esta tendencia a ser master) y los que advierten de su disponibilidad y aceptan conexiones serán slave (normalmente son los dispositivos más pequeños, simples y con limitaciones de memoria, como sería el caso de los sensores).

Para establecer una conexión el master debe comenzar un escaneado para encontrar anunciantes (slaves) que están aceptando peticiones de conexión. Acto seguido, se le envía un paquete de solicitud de conexión al slave y, siempre que el slave responda, se establece la conexión. Por motivos de seguridad o en los que el master o el slave pueden estar interesados sólo en un grupo de dispositivo conocidos, la capa de enlace implementa una lista blanca, la cual especifica las direcciones de dispositivos en los que se está interesado en el escaneado, ignorándose aquellos que no pertenezca a esta lista. Una vez realizado el escaneado y establecida la conexión se procede a la transmisión y recepción de datos. La capa de enlace proporciona los medios para que se produzca este tráfico de datos de forma segura a través de un enlace cifrado. Aunque las claves son generadas y administradas por el host, la capa de enlace es quien realiza el cifrado y descifrado de forma transparente para las capas superiores.

3. Host Controller Interface (HCI)

Es un conjunto de comandos y eventos para que el host (el siguiente nivel) y el controlador interactúen entre ellos.

HOST

Incluye las capas:

1. Logical Link Control and Adaptation Protocol (L2CAP)

Está al cargo del enrutado de los dos principales protocolos: el Attribute Protocol (ATT) y el Security Manager Protocol (SMP), encapsulándolos en el formato estándar de paquetes de BLE.

2. Attribute Protocol (ATT)

Es un protocolo simple, sin estado cliente/servidor, basado en los atributos que presenta un dispositivo.

En BLE un dispositivo es cliente, servidor o ambos, independientemente de si es master o slave. Un cliente pide datos al servidor, y un servidor envía datos a los clientes.

Cada servidor contiene datos organizados en forma de atributos, y a cada uno de los cuales se le asigna un identificador único universal (UUID), un conjunto de permisos y un valor. A través del UUID accedemos al valor. Veremos más en profundidad los detalles de esta capa en el próximo punto.

3. Security Manager (SM)

Es de forma simultánea un protocolo y una serie de algoritmos seguros diseñados para dotar a la pila de protocolos la capacidad de generar e intercambiar claves de seguridad, la cuales, posteriormente, permitirán a los nodos comunicarse de forma segura a través de un enlace encriptado, verificar la identidad de dispositivos remotos y ocultar la dirección pública del Bluetooth si se quiere evitar que nodos malintencionados rastreen un dispositivo en particular.

SM proporciona tres procedimientos de seguridad:

1. Emparejamiento (Pairing): es el procedimiento por el cual se establece una clave de encriptación común para habilitar el intercambio seguro de un enlace encriptado. Esta clave no se almacena ni se reutiliza en posteriores conexiones.
2. Vinculación (Bonding): es una secuencia de emparejamiento seguida por la generación e intercambio de claves de seguridad permanentes, que se almacenarán en memoria no volátil y, por lo tanto, creando un enlace permanente entre dos dispositivos, lo cual permitirá agilizar el enlace seguro en siguientes conexiones sin tener que realizar otra vez el procedimiento de emparejamiento.
3. Restablecimiento del cifrado (Encryption Re-establishment): después de que la vinculación se ha completado, las claves deben almacenarse en ambas partes de la conexión. Si las claves encriptadas han sido almacenadas, este procedimiento define como usarlas en las siguientes conexiones para reestablecerla de forma segura, sin tener que realizar el emparejamiento o el enlace nuevamente.

4. Generic Attribute Profile (GATT)

El Generic Attribute Profile (GATT) se construye sobre el Attribute Protocol (ATT) y añade una modelo de abstracción de datos jerárquico. Podría considerarse como el eje central de las transferencias de datos en BLE porque define como los datos son organizados e intercambiados entre aplicaciones.

Define objetos de datos genéricos que pueden ser utilizados y reutilizados por varios perfiles de aplicación (GATT-based profiles).

Mantiene la misma arquitectura cliente/servidor que tiene ATT, pero los datos están ahora encapsulados en servicios, los cuales se componen de una o más características.

Cada característica puede ser pensada como la unión de los datos del usuario junto con los metadatos (información descriptiva sobre ese valor, como pueden ser: propiedades, el nombre visible para el usuario, las unidades, etc.).

5. Generic Access Profile (GAP)

El Generic Access Profile (GAP) dicta como los dispositivos interactúan unos con otros a bajo nivel, fuera de la pila de protocolo real. Se puede considerar que GAP define la capa superior de BLE, dado que especifica como los dispositivos realizan los procedimientos de control: descubrimiento, conexión, establecimiento de la seguridad, y aquellos otros procedimientos que aseguren la interoperabilidad y permitan el intercambio de datos entre dispositivos de diferentes proveedores.

APLICACIÓN

Es la capa superior y la responsable de contener la lógica, la interfaz de usuario y el manejo de los datos de cada caso de uso.

Consejos de seguridad

Como hemos observado, el fabricante del dispositivo BLE debe realizar un desarrollo y diseño con una implementación completa de la capa de enlace y de los procedimientos del Security. Esto no siempre es así y la seguridad de la tecnología Bluetooth no está exenta de problemas: el 80% de los dispositivos BLE no implementan encriptación en la capa de enlace y las aplicaciones móviles delegan el emparejamiento a nivel de sistema operativo.

Por ello, desde el CERTSI (CERT de Seguridad e Industria de España) indican seguir las siguientes recomendaciones para asegurar las comunicaciones:

- Activar el cifrado de las comunicaciones.

- No aceptar conexiones de dispositivos desconocidos. Activar en el maestro la opción de listas blancas y requerir emparejamientos con clave de al menos 5 caracteres, evitando que dispositivos maliciosos puedan conectarse sin permiso.
- Revisar periódicamente la lista de dispositivos de confianza registrados para evitar la aparición de dispositivos maliciosos.
- Asignar un nombre a los dispositivos que no refleje información extra como puede ser la marca, el modelo del dispositivo, la ubicación o el servicio del mismo. Así se dificulta que posibles atacantes aprovechen vulnerabilidades asociadas a dispositivos concretos para realizar ataques dirigidos.
- Mantener la configuración del dispositivo en modo invisible para dificultar la detección por parte de otros dispositivos.

Herramientas para depuración

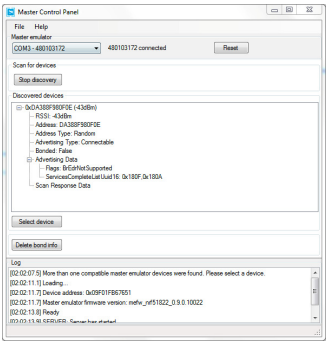
A continuación, mostramos herramientas que son de ayuda para el desarrollo y depuración para trabajar con BLE.

PCA10000 USB Dongle y el Panel de Control Maestro

Este dispositivo USB es una llave forma parte de un kit concebido para embeber desarrollos BLE, pero debido al gran número de herramientas que proporciona puede usarse para aquellos casos en los que se esté desarrollando aplicaciones móviles.

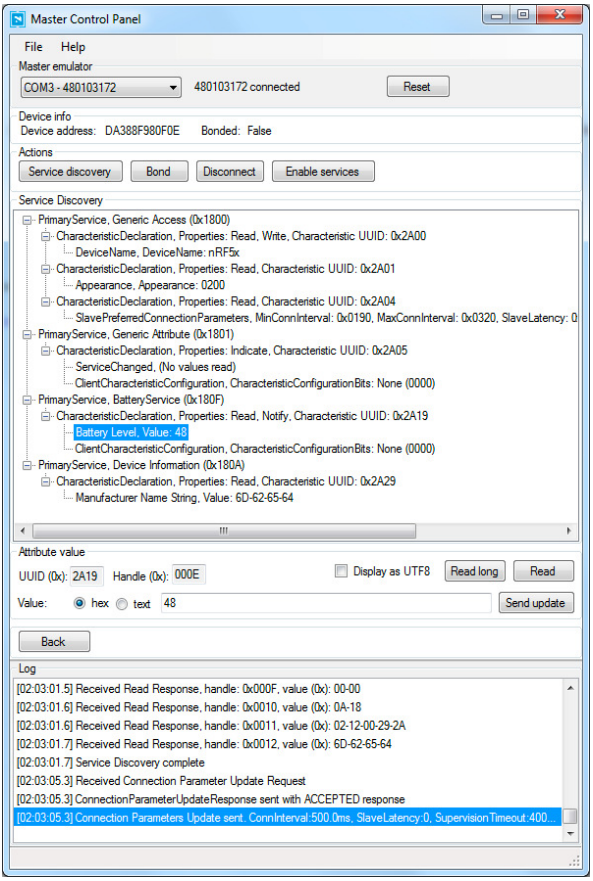
Una de esas herramientas es el Panel de Control Maestro (Master Control Panel, MCP), una aplicación Windows que convierte el PCA10000 USB en algo capaz de simular un dispositivo BLE central, permitiendo visualizar cualquier dato enviado o recibido a los periféricos BLE conectados. En el caso de Windows 7, que no dispone de un soporte nativo para Bluetooth Low Energy, el MCP es una buena solución (El soporte para BLE fue introducido en Windows 8, pero este no incluye ninguna aplicación comparable para pruebas y depuración).

Una vez abrimos la aplicación MCP, podemos interactuar con nuestro periférico a través de la interfaz gráfica, la cual nos permite realizar cualquier funcionalidad de un dispositivo central.



MCP - Resultados de la búsqueda de dispositivos

Después de habernos conectado al periférico y enviado la petición para el descubrimiento de servicios, podemos ver el listado de servicios y características disponibles en el dispositivo. Llegados a este punto, podremos leer o escribir en ellos de igual forma que si lo hiciéramos con un dispositivo BLE.



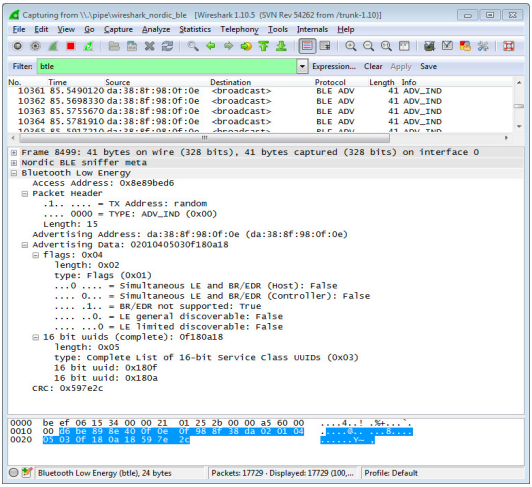
MCP – Listado de servicios y características

MCP también incluye un conjunto de librerías C# que pueden ser usadas para automatizar cualquiera de las funcionalidades vistas, permitiendo a los desarrolladores crear aplicaciones de escritorio o de línea de comandos con acceso a una simple y completa API. Esto puede ser muy útil en casos de pruebas automatizadas o pruebas en producción.

PCA10000 USB Dongle y Wireshark

Como hemos visto en el punto anterior, MCP es la forma más sencilla para interactuar con periféricos BLE, pero en algunos casos podemos necesitar acceder a datos a más bajo nivel. Para estos casos, encontramos que en el kit de desarrollo del PCA10000 una imagen de firmware y herramientas que pueden capturar el tráfico de un dispositivo periférico y mandarlo a Wireshark.

Wireshark es una herramienta para la captura de datos y análisis de código abierto, que permite visualizar de una forma sencilla los datos que hay en los paquetes y a nivel de byte.



Wireshark plugin del kit de desarrollo del PCA10000

Este tipo de utilidades, con un trabajo a tan bajo nivel, es bastante útil para los cuando estamos desarrollando diseños de hardware o firmware, o para depurar problemas de latencia o rendimiento.

Bluez hcitool y gatttool

Si tenemos un sistema operativo Linux, podemos usar dos útiles herramientas de la Bluez Bluetooth Stack (denominación que recibe en Linux la implementación de la especificación Bluetooth), que son

hcitool y gatttool, las cuales permiten interactuar con dispositivos BLE desde la línea de comandos.
Si no tenemos acceso a un equipo con Linux, Bluez también funciona en dispositivos como Raspberry Pi o BeagleBone Black, los que los convierte en herramientas para depurar BLE bastante útiles.

hcitool permite escanear dispositivos BLE por rango, conectarnos a él, o simular un dispositivo BLE usando un USB dongle compatible. Para escanear dispositivos BLE por rango, podemos usar el siguiente comando, dando por hecho que nuestro USB dongle es enumerado como hci0:

sudo hcitool -i hci0 lescan

Como resultado veremos un listado de direcciones de dispositivos. Para conectarnos al periférico que deseamos debemos usar el siguiente comando al que le pasamos como argumento la dirección:

sudo hcitool lecc 6C:60:B3:6E:7C:B1

Por otro lado, gatttool nos permite interactuar con los servicios GATT, tanto para escribir como para leer características en el dispositivo.

Apps para iOS y Android

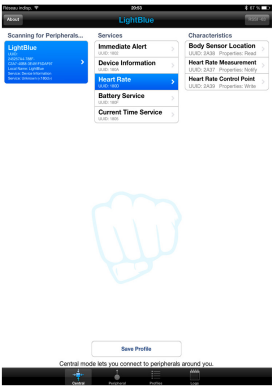
Normalmente a los desarrolladores de aplicaciones móviles no les interesa diseñar su propio hardware BLE, sin embargo, necesitan dialogar con periféricos BLE existentes.

1. Dispositivos iOS

En el caso de iOS, los de Apple realizan una implementación de BLE en la que estos dispositivos no se muestran desde la opción Ajustes / Bluetooth / Otros dispositivos, la cual está reservada sólo para Bluetooth. La única forma de interaccionar con periféricos BLE es

mediante aplicaciones que tendremos que descargar desde App Store.

LightBlue es una de estas aplicaciones. Es gratuita y podemos usarla para ingeniería inversa o emular un periférico BLE desde iPhone o iPad.
De esta forma podremos interactuar con los servicios y características que nos exponga el dispositivo, por ejemplo, escribiendo o leyendo valores. También podemos capturar la firma única de un dispositivo BLE existente y utilizarla para comprobar la seguridad del dispositivo, emulando que no podemos reutilizarla en posteriores ocasiones.



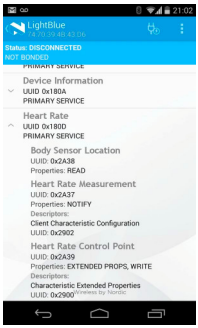
App LightBlue – Mostrando un dispositivo BLE simulado

2. Dispositivos Android

Al contrario que ocurre en iOS, desde Android podemos buscar y conectar a dispositivos BLE desde la opción del sistema Ajustes / Conexiones / Bluetooth
Sin embargo, para interaccionar con los servicios y características del periférico necesitamos instalar una app desde Google Play Store.

De las muchas que podemos encontrar para tal fin, hemos elegido nRF Master Control Panel, desarrollada por el mismo equipo que el UBS dongle PCA10000.

Una vez descargada e instalada, podremos buscar el UUID para cualquier servicio o característica presente o periféricos BLE cercanos, subscribirnos a notificaciones enviadas por las características o escribir valores en el periférico.



nRF Master Control Panel – Mostrando información de un servicio

Es importante destacar que, en el caso de Android ya que es posible realizar conexiones a dispositivos BLE desde sistema a través de la interfaz de usuario, la app móvil que usemos debe estar desarrollada de forma que mantenga la coherencia de renovación de conexión con el dispositivo cada vez que vaya a consumir un servicio, ya existe una alta probabilidad que desde sistema el usuario haya eliminado el periférico y tengamos que repetir los pasos de descubrimiento y enlace desde la app.

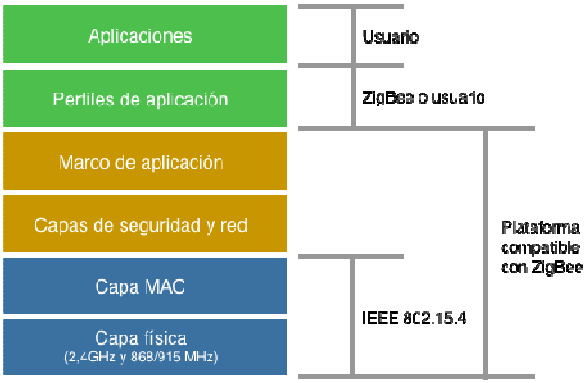
ZIGBEE

Introducción

ZigBee es un protocolo inalámbrico, desarrollado por la ZigBee Alliance que adopta el estándar IEE 802.15.4 para las capas bajas del modelo OSI.
De este forma, se asienta sobre la capa física y de enlace operando directamente por encima de estos niveles.

Este protocolo está teniendo una gran presencia en sectores como la salud, el transporte o las Smart Cities o ciudades inteligentes, ya que entre sus principales características están el bajo consumo de energía y la posibilidad de usar topología de red en forma de malla, proporcionando gran robustez a las comunicaciones y haciéndolo adecuado para entornos industriales.

A nivel de escalabilidad, ZigBee ofrece mayores posibilidades que otros protocolos inalámbricos como sería el caso de Bluetooth, ya que permite usar hasta 65535 nodos distribuidos en subredes de 255 nodos, frente a los 8 nodos máximos posibles en una subred Piconet (Bluetooth). Sin embargo, al tener una velocidad de transmisión menor que la de Bluetooth (250 kbps frente a 1 Mbps en Bluetooth), su uso industrial será en función del gasto energético, coste, necesidades de cobertura y fiabilidad.



Roles y redes ZigBee

Según la disposición de los nodos, podemos tener tres tipos de topologías: en malla, árbol o estrella. Los nodos pueden asumir a su vez, tres roles distintos:

- **Coordinador:** realiza las funciones de inicio, control y enrutamiento, por lo que requiere memoria y gran capacidad de comunicación. El coordinador, es único en cada red y se sitúa en el centro de una red en estrella o en la raíz de una red en árbol. Ejerce un papel clave en las funciones de gestión de la seguridad de las comunicaciones actuando como Centro de Confianza (Trust Center), como veremos más adelante.
- **Router:** gestiona las rutas de comunicación entre los dispositivos. Entre las situaciones que pueden darse existe el caso de congestión en la red o problemas dentro de esta durante el enlace entre nodos. En una red Zigbee puede haber más de un router.
- **Dispositivo final (End Device):** Son aquellos dispositivos que únicamente se comunican con un nodo padre, pudiendo ser este un router o el coordinador, y no tiene capacidad para gestionar otros nodos finales.

Cualquier nodo puede ejercer cualquiera de las tres funciones dependiendo de la configuración y de la infraestructura de red. Aunque existen nodos que vienen configurados con un solo tipo de función, en general todos los dispositivos comerciales pueden ser configurados para asumir cualquiera de los roles.

La seguridad

La seguridad ZigBee, basada en un algoritmo AES de128 bits, incrementa el nivel de seguridad proporcionado por IEEE 802.15.4. Los servicios de seguridad incluyen métodos para el establecimiento y el transporte de claves, la gestión de dispositivos y la protección de marcos.

La especificación ZigBee representa seguridad para las capas MAC, NWK y APS. La seguridad para las aplicaciones se proporciona generalmente a través de Perfiles de aplicación.

Centro de confianza

El centro de confianza (Trust Center) decide si nuevos dispositivos pueden conectarse o no a la red. Además, puede actualizarse de forma periódica y renovar la clave de red.

Primero, transmite la nueva clave encriptada con la clave de red anterior. Después, comunica a todos los dispositivos que cambien a la nueva clave.

La función de coordinación de la red es llevada a cabo, normalmente por el Centro de confianza, pero en otras ocasiones podría ser un dispositivo dedicado. Así, es el encargado de las siguientes funciones de seguridad:

- Administrador de confianza: para autenticar dispositivos que solicitan unirse a la red.
- Administrador de red: para mantener y distribuir claves de red.
- Administrador de configuración: para habilitar la seguridad de extremo a extremo entre dispositivos.

Criptografía

ZigBee basa su arquitectura de seguridad en el uso de clave simétrica realizado con un robusto protocolo para la gestión de claves.

Usa tres tipos de claves según se asocien a una red, a un grupo de dispositivos o a un enlace entre dos elementos:

- Clave maestra (master key): clave a partir de la cual se generan las diferentes claves de enlace. Dada su importancia, la clave maestra inicial ha de obtenerse por medios seguros (preinstalación o transporte de claves, ambos medios se detallan más adelante dentro de este artículo).
- Clave de enlace (link key): cifra las comunicaciones punto a punto a nivel de aplicación, y sólo es conocida por los elementos que participan en dicho enlace. Esta clave sólo se comparte entre

dos elementos de la red y varía para cada pareja de elementos. La clave de enlace es utilizada para minimizar los riesgos de seguridad relacionados con la distribución de la clave maestra.

- Clave de red (network key): clave utilizada a nivel de red y conocida por todos los elementos pertenecientes a ésta. La clave de red también se utiliza en agrupaciones de más de dos elementos dentro de una red.

Debilidades en ZigBee

Muchas entidades privadas y públicas, así como investigadores de seguridad, coinciden en que la principal debilidad que existe en los mecanismos de seguridad realidos en ZigBee, son consecuencia directa de la limitación de recursos en los nodos, debido a que un porcentaje se alimenta mediante baterías, tienen poca capacidad de procesado y escasa memoria.

Los dispositivos ZigBee guardan en memoria las claves que utilizan, por lo que un intruso con acceso físico (mediante un software específico) al dispositivo puede leer de una forma bastante sencilla la clave y no se encuentran implementados mecanismos de seguridad, así como acceso al software de seguridad. Para evitar estos problemas es recomendable el uso de un microcontrolador para la autenticación segura eliminando cualquier riesgo de manipulación física.

Referencias Bibliográficas

1. CERTSI - Seguridad en Bluetooth: fortalezas y debilidades https://www.certsi.es/blog/seguridad-bluetooth-fortalezas-y-debilidades
2. CERTSI – Analizando Bluetooth https://www.certsi.es/blog/analizando-bluetooth
3. OWASP Israel - BLE Hacking Application https://www.owasp.org/images/6/6f/OWASP2017_HackingBLEApplications_TalMelamed.pdf
4. Reversing and exploiting BLE 4.0 communication https://payatu.com/reversing-exploiting-ble-4-0-communication/
5. Getting Started with Bluetooth Low Energy – O'Reilly https://www.safaribooksonline.com/library/view/getting-started-with/9781491900550/ch02.html
6. CERTSI – Seguridad en comunicaciones ZigBee https://www.certsi.es/blog/seguridad-comunicaciones-zigbee
7. General Electric Company – ZigBee Security FAQ http://solutions.currentbyge.com/LightingWeb/la/north/images/DT302-GE-ZigBee-Security-FAQ_ES_tcm402-133465.pdf

Análisis de firmware

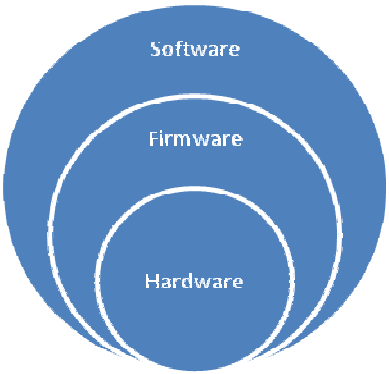
Miguel Ángel Arroyo Moreno

Introducción al análisis de firmware

Tal y como hemos visto en capítulos anteriores, la seguridad en un entorno IoT abarca más de un ámbito y por lo tanto son varios los vectores de ataque y amenazas que pueden poner en peligro la información que se almacene, procese o transmita a través de un dispositivo IoT.

Esto nos obliga a seguir un modelo de seguridad que nos garantice que cubrimos todos los posibles riesgos a los que pueda estar expuesto el sistema. Los elementos que son susceptibles a estar afectados por alguna vulnerabilidad son de diferentes tipos; software, hardware y firmware, por lo tanto, necesitamos un conjunto de controles a nivel de software, hardware y firmware que nos garantice la protección de la información en cualquiera de sus tres dimensiones básicas; confidencialidad, integridad y disponibilidad.

Un TCB (*Trusted Computer Base*) es precisamente eso, un conjunto de elementos de estos tres tipos que nos permiten implementar los requisitos de seguridad establecido, o demandados por una política de seguridad existente.



TCB (Trusted Computer Base)

Este capítulo va dedicado a evaluar la seguridad del firmware de los dispositivos que componen nuestro ecosistema IoT, para ello veremos cómo analizar un firmware desde el punto de vista de la seguridad, identificar vulnerabilidades e intentar explotarlas.

El firmware, que es el software que está más integrado con el hardware del dispositivo, y que tiene la capacidad de manejarlo, contiene diferentes elementos que trataremos para poder analizarlos.

Una de las ventajas de analizar el firmware, es que no necesitamos físicamente el dispositivo para poder analizarlo.

Imagen de la web de enisa.europa.eu

¿Qué contiene un firmware?

Para entender bien cómo evaluar la seguridad de un firmware y los posibles riesgos asociados a las posibles vulnerabilidades presentes, tenemos que identificar las secciones que nos podemos encontrar.

- Programas de arranque del dispositivo (*Boot Loader*).
- Kernel (Núcleo del firmware).
- Sistema de ficheros.
- Conjunto de programas que ejecuta el dispositivo.

¿Qué podríamos hacer con un firmware?

Una vez que hemos conseguido extraer los elementos del firmware, podemos llevar a cabo diferentes tareas y aplicar unas técnicas para:

- Buscar contraseñas integradas en el código o almacenadas de forma insegura.
- Buscar certificados o claves de API (Interfaz de Programación de Aplicaciones) para conectar con aplicación del fabricante.
- Buscar vulnerabilidades en el código de las aplicaciones web.
- Buscar vulnerabilidades en los binarios presentes en el sistema de ficheros.
- Modificar el firmware, integrando una puerta trasera, para crear un firmware malicioso.
- …

¿Cómo podemos obtener el firmware?

Esta es una de las cuestiones más importantes, ya que los fabricantes son cada vez más reacios a publicar su firmware y que esté accesible para todo el mundo. Aun así, todavía hay fabricantes que los publican en un servidor FTP o cualquier otro sitio público de su web.

Las formas de conseguir el firmware de un dispositivo IoT serían:

- A través de la web o ftp del fabricante.
- Volcando el firmware de la memoria flash del dispositivo.
- Interceptando el tráfico durante una actualización, justo en el momento cuando se descargue el firmware que se va a instalar como actualización.
- …

En los siguientes apartados veremos cómo extraer información relevante de un firmware para su posterior análisis.

Sistemas de ficheros de los archivos firmware

Una de las secciones más importantes de un firmware, y donde más profundizaremos en este capítulo, es el sistema de ficheros. Hay varios tipos de sistemas de ficheros para dispositivos embebidos como los de IoT, y cada uno de ellos tiene una firma diferente que los identifica dentro del firmware y permite localizarlos para su posterior extracción.

Esta tarea de extracción del sistema de ficheros requiere unas técnicas y herramientas especiales, ya que hay que destacar que el fichero de firmware es un fichero de tipo binario, por lo que se manipulación no es simple.

Estos son los tipos de sistemas de ficheros más habituales en dispositivos IoT:

- Squashfs
- Cramfs
- JFFS2
- YAFFS2
- ext2

Además, tenemos que tener en cuenta que el espacio de almacenamiento en los dispositivos embebidos es un bien muy preciado, por lo que se intenta aprovechar al máximo el espacio disponible.

Uno de los métodos más utilizados para este ahorro de espacio, es la compresión. Algunos de los tipos de compresión más utilizados en dispositivos embebidos son:

- LZMA
- Zip / Gzip
- Zlib
- ARJ

Extracción del sistema de ficheros

Como hemos visto en el apartado anterior, la primera fase para el análisis del firmware obviamente es la obtención del mismo, y para lo que tenemos varias opciones. Una vez obtenido el firmware, procederemos a extraer el sistema de ficheros del mismo, el cual analizaremos en busca de posibles vulnerabilidades.

Para extraer el sistema de ficheros, tenemos varias opciones. La manera más sencilla es utilizando una herramienta llamada "*binwalk"*.

La podemos descargar desde esta URL:

- https://github.com/ReFirmLabs/binwalk

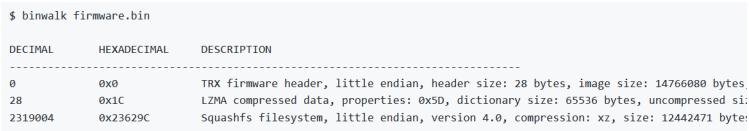


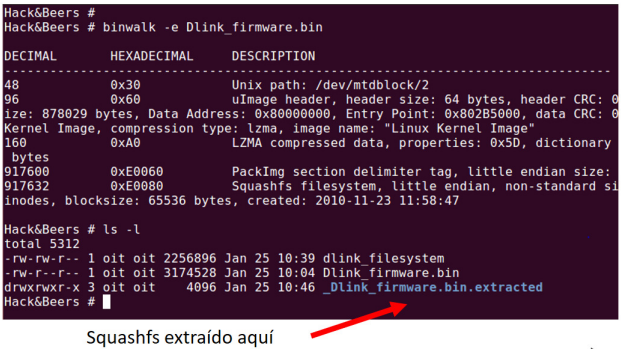
Imagen del sitio Github de Binwalk

En la imagen anterior se puede ver un uso básico de la herramienta, a la que le pasamos como argumento el nombre del fichero del firmware que queremos procesar.

En las dos últimas líneas se pueden observar los conceptos que hemos comentado en apartados anteriores, relativos al tipo de sistema de ficheros y a la compresión utilizada. En este caso, se trata de un sistema de ficheros del tipo **Squashfs**, y la compresión utilizada es **LZMA**.

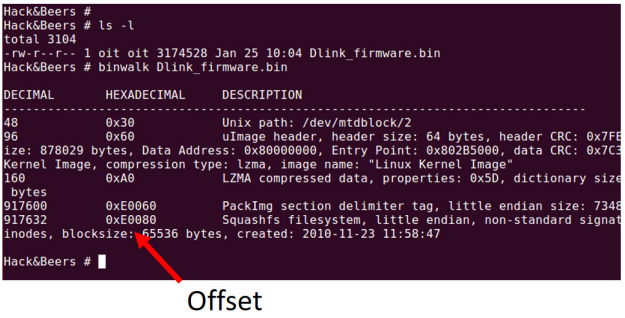
Lo interesante de esta herramienta es que identifica automáticamente cada segmento del firmware, diferenciado las secciones del *kernel*, *boot loader* o sistema de ficheros, lo que facilita mucho la extracción de cualquiera de estos elementos.

En la siguiente imagen podemos ver cómo se extrae el sistema de ficheros de un fichero de firmware de un router **Dlink**.



Otra opción es extraer el sistema de ficheros de forma manual, con otra herramienta como *dd*, un comando de los sistemas operativos Unix capaz de copiar datos a bajo nivel.

Para ello, en primer lugar, necesitamos localizar la dirección o el *offset* (desplazamiento) del sistema de ficheros dentro del fichero del firmware.



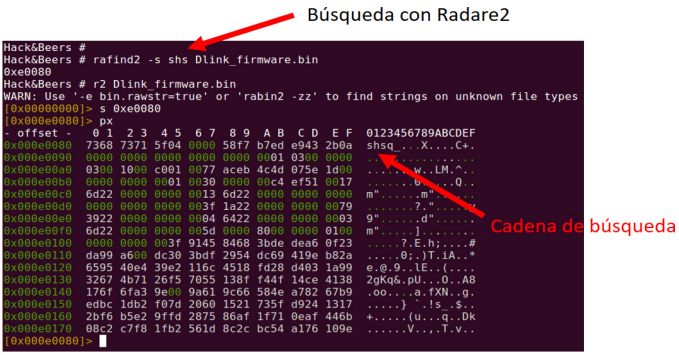
En la imagen anterior podemos ver que el desplazamiento es 0xe0080, correspondiente al lugar donde comienza el sistema de ficheros Squashfs que queremos extraer.

El desplazamiento en este caso (offset) no es ni más ni menos el tamaño que ocupa cada sección del fichero. En la primera columna se muestra en formato decimal (en bytes), y en la segunda columna en hexadecimal, haciendo referencia a la dirección de desplazamiento.

De hecho, podemos comprobar que 0xe0080 en decimal es 917632.

```
/home/oit/lab/pocs [oit@ubuntu] [4:03]
> rax2 0xe0080
917632
```

Analizando el fichero firmware (recordemos que es un binario) con herramientas de análisis de ficheros binarios, también podemos localizar el desplazamiento.



Localizando el offset con Radare2

En la imagen anterior se han utilizado dos herramientas de la suite de **Radare2**, una suite que incluye un conjunto de utilidades para el análisis de binarios. Radare2 es una de las herramientas más potentes para hacer ingeniería inversa a binarios.

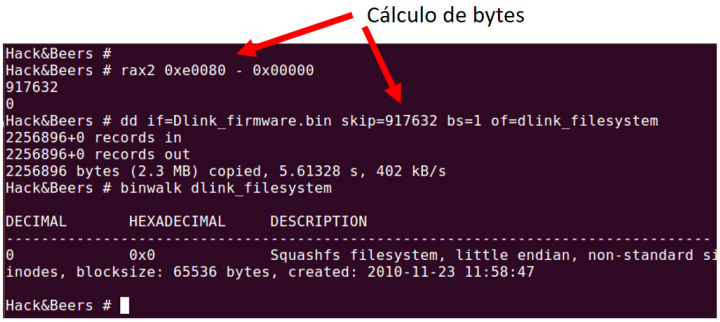
Con la herramienta **rafind2** se ha buscado y localizado la cadena "*shs*", dándonos como resultado la dirección que estábamos buscando.

Con **radare2** (o *r2*), nos hemos ubicado en la dirección y mediante un volcado hemos localizado la cadena de búsqueda que estábamos localizando, confirmando el offset (0xe0080).

Este desplazamiento, que hemos comprobado con diferentes métodos y herramientas, es la cantidad de bytes que hay justo antes de que comience la sección del sistema de ficheros. Esto significa, que tendremos que omitir esta cantidad cuando vayamos a extraer el sistema de ficheros.

Esta extracción, tal y como hemos nombrado antes, la vamos a hacer con la herramienta *dd*, en la cual especificaremos el fichero de origen (fichero firmware), el fichero de destino (que contendrá el sistema de

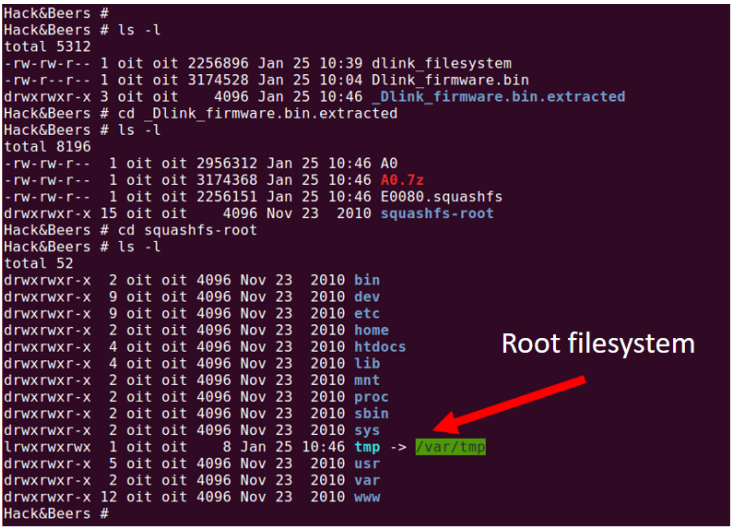
ficheros) y la cantidad de bytes a omitir (indicado con el parámetro *skip*, como podemos ver en la siguiente imagen).



Finalizada la ejecución de la orden, obtendremos un fichero que contendrá el sistema de ficheros del firmware.

Como podemos comprobar, es mucho más sencillo extraer el sistema de ficheros con la herramienta binwalk, ya que no nos tenemos que preocupar de localizar dónde comienza una sección o de omitir un número específico de bytes a la hora de copiar los datos a bajo nivel con la herramienta *dd*.

Si hemos optado por la extracción del sistema de ficheros con binwalk, nos quedará un directorio que contiene el sistema extraído.



Como se puede observar en la imagen, el sistema de ficheros sigue una jerarquía estándar de los sistemas de la familia UNIX, conocida como FHS (*Filesystem Hierarchy Standard*, Jerarquía Estándar de Sistema de Ficheros), con sus típicos directorios /bin, /dev, /etc, /home, /var o /www, etc.

Una vez que tenemos accesible el sistema de ficheros, ya podemos recorrerlo en busca de información interesante. Algunos de los directorios de interés pueden ser:

- /etc
- /home
- /var
- /www
- /bin
- /sbin

En estos directorios podemos encontrar ficheros de configuración, certificados o claves de algún conector o API del fabricante.

```
/home/oit/lab/pocs/_Dlink_firmware.bin.extracted/squashfs-root/etc [oit@ubuntu] [6:20]
> ls -l
total 32
drwxrwxr-x  2 oit oit 4096 Nov 23  2010 config
drwxrwxr-x  2 oit oit 4096 Nov 23  2010 defnodes
lrwxrwxrwx  1 oit oit   10 Jan 25  2017 hosts -> /var/hosts
drwxrwxr-x  2 oit oit 4096 Nov 23  2010 init.d
drwxrwxr-x  2 oit oit 4096 Nov 23  2010 netsniper
lrwxrwxrwx  1 oit oit   12 Jan 25  2017 ppp -> /var/etc/ppp
lrwxrwxrwx  1 oit oit   20 Jan 25  2017 resolv.conf -> /var/etc/resolv.conf
-rw-rw-r--  1 oit oit  512 Nov 23  2010 RT3050_AP_1T1R_V1_0.bin
drwxrwxr-x  3 oit oit 4096 Nov 23  2010 scripts
drwxrwxr-x 12 oit oit 4096 Nov 23  2010 templates
drwxrwxr-x  2 oit oit 4096 Nov 23  2010 tlogs
lrwxrwxrwx  1 oit oit    7 Jan 25  2017 TZ -> /var/TZ
```

En el directorio /www encontraremos archivos correspondientes al código de la página o aplicación web existente. La gran mayoría de dispositivos ofrecen un servicio web que permite su administración. Pues bien, recorriendo este directorio podremos por ejemplo analizar de forma estática algunos ficheros **php** que pueda haber presentes.

```
/home/oit/lab/pocs/_Dlink_firmware.bin.extracted/squashfs-root/www [oit@ubuntu] [6:21]
> ls -l | more
total 960
-rw-rw-r-- 1 oit oit  2745 Nov 23  2010 __adv_app.php
-rw-rw-r-- 1 oit oit 11105 Nov 23  2010 adv_app.php
-rwxrwxr-x 1 oit oit  8503 Nov 23  2010 adv_apx.php
-rw-rw-r-- 1 oit oit   964 Nov 23  2010 __adv_firewall_httpallow.php
-rw-rw-r-- 1 oit oit  3511 Nov 23  2010 __adv_firewall.php
-rw-rw-r-- 1 oit oit 25978 Nov 23  2010 adv_firewall.php
-rw-rw-r-- 1 oit oit   847 Nov 23  2010 __adv_firewall_pingallow.php
-rw-rw-r-- 1 oit oit   960 Nov 23  2010 __adv_firewall_vrtsrv.php
-rw-rw-r-- 1 oit oit   938 Nov 23  2010 __adv_mac_filter.php
-rw-rw-r-- 1 oit oit  9612 Nov 23  2010 adv_mac_filter.php
-rw-rw-r-- 1 oit oit  5460 Nov 23  2010 adv_network.php
-rw-rw-r-- 1 oit oit  3753 Nov 23  2010 __adv_port.php
-rw-rw-r-- 1 oit oit 12682 Nov 23  2010 adv_port.php
-rw-rw-r-- 1 oit oit  8830 Nov 23  2010 adv_qos.php
```

En los directorios /bin y /sbin nos vamos a encontrar los binarios, los cuales podremos analizar y realizar ingeniería inversa para identificar posibles vulnerabilidades.

```
/home/oit/lab/pocs/_Dlink_firmware.bin.extracted/squashfs-root/sbin [oit@ubuntu] [6:24]
> ls -l
total 248
-rwxrwxr-x 1 oit oit  46856 Nov 23  2010 atp
-rwxr-xr-x 1 oit oit 127100 Nov 23  2010 httpd
lrwxrwxrwx 1 oit oit     14 Jan 25  2017 ifconfig -> ../bin/busybox
lrwxrwxrwx 1 oit oit     14 Jan 25  2017 init -> ../bin/busybox
lrwxrwxrwx 1 oit oit     14 Jan 25  2017 insmod -> ../bin/busybox
-rwxr-xr-x 1 oit oit   9948 Nov 23  2010 klogd
lrwxrwxrwx 1 oit oit     14 Jan 25  2017 lsmod -> ../bin/busybox
```

Una cuestión importante a tener en cuenta, es que, aunque estemos visualizando el sistema de ficheros desde nuestro sistema operativo (con arquitectura **x86** de 32 o 64 bits), la arquitectura que habitualmente utilizan los dispositivos embebidos e IoT es diferente, por ejemplo, **ARM** o **MIPS**.

```
/home/oit/lab/pocs/_Dlink_firmware.bin.extracted/squashfs-root/bin [oit@ubuntu] [6:16]
> ls busybox
busybox

/home/oit/lab/pocs/_Dlink_firmware.bin.extracted/squashfs-root/bin [oit@ubuntu] [6:16]
> rabin2 -I busybox
file    /home/oit/lab/pocs/_Dlink_firmware.bin.extracted/squashfs-root/bin/busybox
type    EXEC (Executable file)
pic     false
has_va  true
root    elf
class   ELF32
lang    c
arch    mips
bits    32
machine MIPS R3000 big-endian
os      linux
subsys  linux
endian  little
strip   true
static  true
linenum false
lsyms   false
relocs  false
rpath   NONE
```

En la imagen anterior, nos hemos ubicado en el directorio /bin del sistema de ficheros que hemos extraído anteriormente, y hemos mostrado información de un binario (*busybox*) con la herramienta **rabin2**, de la suite de Radare2.

Como se puede ver en la imagen, se trata de un binario de una máquina MIPS R3000, con arquitectura MIPS, por lo tanto, entre otras cosas, no podremos ejecutar algunos de estos binarios en nuestro sistema, salvo que lo hagamos con un emulador de MIPS

Una opción interesante para esta emulación es el emulador QEMU, que es un emulador de procesadores basado en la traducción dinámica de binarios.

Su página web es: https://www.qemu.org/

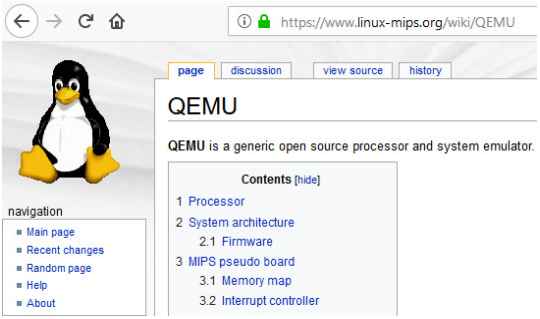


Imagen de Wiki de emulación de MIPS con el emulador QEMU

Otro problema con el que nos vamos a encontrar es que a la hora de ejecutar algunos binarios, éstos no van a encontrar los ficheros necesarios como pueden ser archivos de configuración o librerías.

```
/home/oit/lab/pocs/_rootfs.squashfs.extracted/squashfs-root [oit@ubuntu] [7:10]
> ls -l bin/busybox
-rwxr-xr-x 1 oit oit 1373744 Jun 23  2011 bin/busybox

/home/oit/lab/pocs/_rootfs.squashfs.extracted/squashfs-root [oit@ubuntu] [7:11]
> ./bin/busybox
/lib/ld-uClibc.so.0: No such file or directory
```

En la imagen anterior podemos ver que a la hora de ejecutar **./bin/busybox**, este binario es incapaz de encontrar una librería, debido a que está haciendo referencia a una ruta de nuestro sistema de ficheros, el del propio sistema operativo.

Con el comando **chroot** podemos cambiar la raíz del sistema para un proceso concreto, de esta forma permitiremos al binario a encontrar los ficheros necesarios para ejecutarse. A la misma vez, utilizamos QEMU para emular MIPS y poder ejecutar correctamente el binario.

```
/home/oit/lab/pocs/_rootfs.squashfs.extracted/squashfs-root [oit@ubuntu] [7:11]
> sudo chroot . ./qemu-mips-static ./bin/busybox
BusyBox v1.11.0 (2011-06-23 15:54:48 IST) multi-call binary
Copyright (C) 1998-2008 Erik Andersen, Rob Landley, Denys Vlasenko
and others. Licensed under GPLv2.
See source distribution for full notice.
```

Referencias Bibliográficas

- Página Github de Binwalk: https://github.com/ReFirmLabs/binwalk
- Página Github de Radare2: https://github.com/radare/radare2
- Página web oficial de QEMU: https://www.qemu.org/

Ingeniería inversa

Miguel Ángel Arroyo Moreno

Introducción a la ingeniería inversa

En capítulos anteriores hemos tratado diferentes vectores de ataque de la superficie de ataques en un dispositivo IoT, y hemos visto que uno de los puntos a tratar es el análisis de firmware.

Este análisis de firmware, además de evaluar la seguridad del sistema de ficheros en busca de vulnerabilidades en los ficheros de configuración o contraseñas almacenadas de forma insegura, nos permite analizar los binarios que contiene, es decir, los programas que el dispositivo ejecuta para funcionar.

La forma más efectiva de analizar los binarios es mediante la ingeniería inversa, que tiene como objetivo intentar conocer en detalle cómo funciona el programa, cuál es el flujo del mismo, las funciones que lo componen, librerías externas utilizadas, cadenas incluidas, llamadas al sistema o variables y constantes utilizadas.

Podríamos resumir la ingeniería inversa de un binario como un proceso mediante el cual intentamos obtener un código lo más parecido al código fuente del programa, y con información suficiente para poder identificar vulnerabilidades y explotarlas. De ahí lo de inverso, intentamos pasar de un programa compilado a algo lo más parecido posible al código fuente del mismo, justo todo lo contrario de lo que hacemos cuando desarrollamos, a partir de un código fuente, compilamos y obtenemos el programa final, el ejecutable (binario).

Esta labor se complica cuando en el mercado de las computadoras, actualmente existen multitud de arquitecturas diferentes, cada una de ellas con una sintaxis y modo de operación propia.

En el caso de los dispositivos IoT, que son dispositivos embebidos, la mayoría pertenecen a arquitecturas **ARM** o **MIPS**. En este capítulo vamos a tratar la arquitectura ARM como referencia para nuestros ejemplos. Concretamente, los ejemplos aquí mostrados están ejecutados en una **Raspberry Pi**, que precisamente funciona con arquitectura ARM.

Lenguaje ensamblador en arquitectura ARM

Partimos de la base de que un programa, una vez compilado, contiene datos en un formato entendible por la computadora donde se va a ejecutar, y por lo tanto difícil de entender por un humano, ya que contiene un código más cercano al hardware, llamado código máquina, compuesto por combinaciones de 0 y 1, que se traducen en instrucciones que la computadora interpreta y ejecuta.

El código más cercano que un humano puede entender, es el ensamblador, que utiliza códigos mnemónicos, que son pequeñas palabras o abreviaturas que se corresponden con una combinación de código máquina.

En este capítulo nos centraremos en el lenguaje ensamblador para la arquitectura ARM.

Observemos la siguiente instrucción (en negrita, podemos ver una instrucción en ensamblador para ARM).

e3a0200d> **mov r2, #13**

El primer valor, en hexadecimal, se corresponde con el código máquina, lo que la computadora entiende y ejecutaría, previo paso de convertirlo a binario (en 0 y 1).

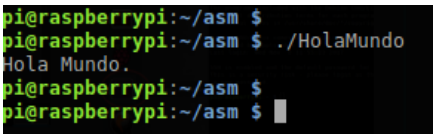
e3a0200d es 11100011101000000010000000001101 en binario.

El texto en negrita, se corresponde a la instrucción en ensamblador para arquitectura ARM, que se obtiene de traducir el código máquina a un código mnemónico, donde;

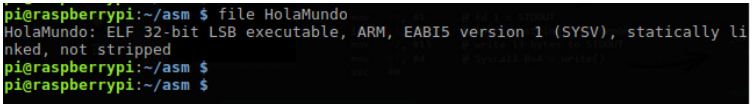
- La instrucción mov se corresponde con el valor e3a
- 02 es el identificador del registro donde queremos almacenar un valor

- 00d es el valor en hexadecimal (00d es 13 en decimal) que queremos asignar al registro r2

En la siguiente imagen podemos ver la ejecución de un simple programa que está escrito en ensamblador para arquitectura ARM, es un sencillo "Hola Mundo".

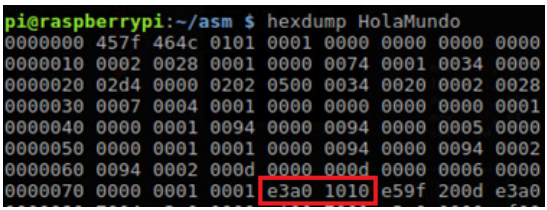


Con el comando *file* de los sistemas Unix, podemos ver información básica sobre el tipo de fichero que estamos tratando.



Podemos observar que se trata de un fichero ejecutable, de tipo **ELF** (*Executable and Linkable Format*, formato habitual de los ficheros ejecutables en la mayoría de sistemas operativos de la familia Unix), para arquitectura ARM.

Si hacemos un volcado en crudo de este fichero, binario, obtenemos el código máquina del mismo. En este caso, lo vamos hacer con la herramienta *hexdump*, que nos da un volcado en hexadecimal del fichero.



Como se puede ver en la imagen, es un código difícil de entender por parte de los humanos. Aunque observando bien, podemos empezar a identificar algunas instrucciones, ¿verdad?

e3a01010 > **mov r1, #010**

Afortunadamente, no tenemos que hacer este ejercicio tan complejo para traducir el código máquina en un código más fácil de leer y entender como es el ensamblador. Para eso existen herramientas que desensamblan el código, y lo traducen a código ensamblador.

Una de estas herramientas, aunque no es específicamente un desensamblador, y que usaremos en varias ocasiones a lo largo de este capítulo, es *objdump*, que forma parte del paquete de utilidades *GNU Binary Utils*.

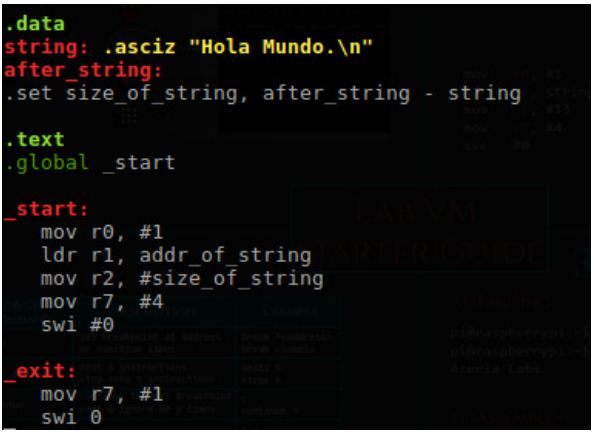
Con el operador –D (**D**isassembly) poder hacer un volcado desensamblando el binario que le pasemos como parámetro, en este caso el programa *HolaMundo*.

```
pi@raspberrypi:~/asm $
pi@raspberrypi:~/asm $ objdump -D HolaMundo

HolaMundo:     file format elf32-littlearm


Disassembly of section .text:

00010074 <_start>:
   10074:       e3a00001        mov     r0, #1
   10078:       e59f1010        ldr     r1, [pc, #16]
   1007c:       e3a0200d        mov     r2, #13
   10080:       e3a07004        mov     r7, #4
   10084:       ef000000        svc     0x00000000
```

Podemos ver el desensamblado del código máquina, con las instrucciones ensamblador para arquitectura ARM. El código ensamblador se organiza en diferentes secciones, y una de ellas, **_start**, es donde se escribe el código del programa.

Existen otras secciones como .text o .data, pero no es objeto de este capítulo entrar en mucho detalle sobre el lenguaje ensamblador.

En la siguiente imagen podemos ver parte del código ensamblador utilizado para la creación del programa *HolaMundo*.



```
.data
string: .asciz "Hola Mundo.\n"
after_string:
.set size_of_string, after_string - string

.text
.global _start

_start:
   mov r0, #1
   ldr r1, addr_of_string
   mov r2, #size_of_string
   mov r7, #4
   swi #0

_exit:
   mov r7, #1
   swi 0
```

Podemos ver las secciones *.data*, *.text* y *_exit*. Esta última utilizada para salir del programa. En la sección .data podemos ver la declaración de la cadena "Hola Mundo." que es la que se muestra cuando ejecutamos nuestro programa.

Para mostrar algo por pantalla, cualquier programa, independientemente del lenguaje de programación usado, sea de más alto nivel o no, utiliza las llamadas al sistema (**syscalls**).

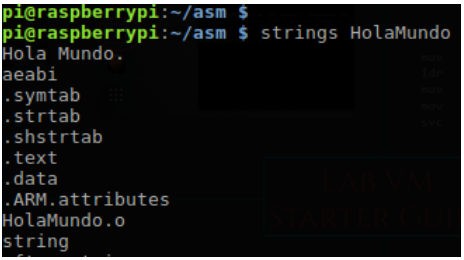
Estas llamadas al sistema son las que nos proporciona el sistema operativo para hacer operaciones básicas como obtener una tecla pulsada, mostrar algo por pantalla, leer un fichero, escribir en un fichero, etc.

Por ejemplo, si en un lenguaje como C, utilizamos la instrucción "*printf*" para mostrar algo por pantalla, internamente esta instrucción utiliza librerías del sistema (stdio.h) para realizar estas operaciones, es decir, acaba haciendo llamadas al sistema para mostrar algo por pantalla, como es el caso de la instrucción *printf*. Lo

mismo ocurriría con *scanf*, instrucción para leer una tecla pulsada, por ejemplo.

Otra opción interesante desde el punto de vista del análisis e ingeniería inversa de un binario, es poder ver qué cadenas son las que están incluidas en el binario a analizar.

Con el comando *strings*, podemos hacer esta consulta, tan solo tenemos que pasarle como parámetro el binario del cual queremos extraer las cadenas.



De la imagen anterior cabe destacar que aparece en la primera línea, precisamente la cadena que queremos mostrar, “Hola Mundo.”. Pero, ¿qué pasaría si un programador incluyera dentro del programa una contraseña sin cifrar? La respuesta es simple, aparecería en la salida de comandos como *strings*.

Supongamos que tenemos un programa que nos solicita un código secreto. Si introducimos el correcto, nos informa de que se ha introducido con éxito, de lo contrario, nos invita a seguir intentándolo.

```
Apisa:
Apisa: ls -l secretCodeStrings
-rwxrwxr-x 1 miguel miguel 7365 Apr  4 09:30 secretCodeStrings
Apisa: ./secretCodeStrings
Introduce el codigo secreto: 12345
Codigo incorrecto, sigue intentandolo.
Apisa:
```

Se ha introducido **12345** como código secreto, sin embargo, no es correcto. Vamos a ver qué nos aporta el comando *strings* sobre este binario, a ver si podemos ver alguna cadena que nos pueda servir como contraseña.

```
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
_IO_stdin_used
__isoc99_scanf
puts
__stack_chk_fail
printf
strcmp
__libc_start_main
GLIBC_2.7
GLIBC_2.4
GLIBC_2.0
PTRh
apisa
UWVS
[^_]
Introduce el codigo secreto:
Codigo correcto!
Codigo incorrecto, sigue intentandolo.
;*2$"  _
```

¡Vaya! El comando *strings* sobre el binario que estamos analizando, nos devuelve varias cadenas, y entre ellas una que nos llama la atención. Probemos con esta contraseña…

```
Apisa:
Apisa: ./secretCodeStrings
Introduce el codigo secreto: apisa
¡Codigo correcto!
Apisa:
Apisa:
```

¡Bingo! Hemos averiguado la contraseña, la cual estaba codificada en el propio código como una cadena.

Con otras herramientas como Radare2, se puede profundizar más en el análisis de este binario, con tareas de ingeniería inversa, analizando funciones del programa, controlando el flujo del mismo e identificando posibles vulnerabilidades a explotar (como desbordamientos en la pila). Este capítulo ha tratado de ser una introducción al análisis de binarios y la ingeniería inversa de binarios con arquitectura ARM.

Referencias Bibliográficas

1. Página web oficial de ARM
   https://www.arm.com/
2. Página web oficial de GNU Binutils:
   https://www.gnu.org/software/binutils/
3. ARM Syscalls:
   https://w3challs.com/syscalls/?arch=arm_strong

Glosario

AIOTI (Alliance for Internet of Things Innovation): Alianza para la Innovación en Internet de las Cosas. Iniciativa enmarcada dentro del contexto Horizonte 2020, pretende convertirse en la base de la investigación e innovación de la Unión Europea en el ámbito de IoT.

ARM: Es una arquitectura RISC (*Reduced Instruction Set Computer*, Ordenador con Conjunto Reducido de Instrucciones) de 32 bits y, con la llegada de su versión V8-A, también de 64 Bits. Es una de las arquitecturas más utilizadas en los dispositivos embebidos e IoT.

Arquitecto IoT: perfil profesional que surge en la industria TIC como pieza angular en la planificación, ejecución y gestión de proyectos IoT.

Black Box: Auditoría de "caja negra", se ejecutan pruebas de intrusión sobre una organización, sin ninguna información de la infraestructura interna. También existe Grey Box y White Box, en función de la información conocida.

BSSID (Basic Service Set Identifier): y se trata de la dirección MAC del Punto de Acceso (AP) al que nos conectamos.

ENISA (European Union Agency for Network and Information Security): Agencia Europea de Seguridad de las Redes y de la Información es un centro de conocimientos especializados para la seguridad cibernética en Europa. La ENISA ayuda a la UE y los países que la integran a estar mejor equipados y preparados para prevenir, detectar y dar respuesta a los problemas de seguridad de la información.

DoS (**Denegation of Service):** denegación de servicio. Es un ataque que afecta a la disponibilidad de un recurso, haciendo que éste deje de estar disponible para usuarios o procesos legítimos.

ESSID (Extended Service Set ID): es el nombre que identifica una red.

Handshake: Apretón de manos, proceso de negociación entre un cliente y su punto de acceso, en el que intercambia información entre ambos.

IaaS (Infraestructure as a Service): Infraestructura como Servicio. Abarca todo el hardware virtualizado, es decir, el espacio en servidores virtuales, las redes, almacenamiento, etc. En definitiva, todos los recursos físicos a los que acceder a través del proveedor del servicio cloud, y en donde poder construir tu propia infraestructura sin realizar grandes inversiones en hardware, ni en mantenimiento.

IoHT (Internet of HealthCare Things): es el término utilizado para definir Internet de las Cosas aplicadas a la Sanidad.

IPSEC (Internet Protocol Security): estándar de seguridad obligatorio para IPv6 y opcional para IPv4, que ofrece encriptación y autenticación de extremo a extremo, haciendo seguras las comunicaciones TCP/IP en redes privadas y públicas.

Jailbreak: proceso mediante el cual se rompe la cadena de seguridad de un dispositivo iOS, permitiendo la instalación de aplicaciones de fuentes inseguras, acceso SSH al terminal, etc.

Man in the Middle (MITM): en esta técnica de ataque, se introduce un intermediario (cibercriminal) entre la víctima y la fuente, de forma que puede observar el tráfico generado entre ambos.

M2M (Machine to Machine): se refiere al intercambio de información o comunicación en forma de datos entre dos máquinas punto a punto.

Objdump: herramienta para el volcado de objetos de un fichero, concretamente podemos volcar el código ensamblador de un binario.

OWASP: Open Web Application Security Project, proyecto abierto de seguridad de aplicaciones web. Proyecto sin ánimo de lucro.

PaaS (Platform as a Service): Plataforma como Servicio. Se añade al servicio IaaS la posibilidad de crear y distribuir aplicaciones en el propio entorno basado en la nube.

Punto de Acceso (AP): es un dispositivo de red que interconecta equipos de comunicación inalámbricos, para formar una red inalámbrica local (WLAN).

Radare2: comenzó como una herramienta de análisis forense, editor hexadecimal, y con la capacidad de abrir ficheros de discos. Actualmente es una herramienta muy potente para el análisis de binarios, ingeniería inversa, desensamblado de código y depuración de programas.

SEDIAN (Seguridad Digital de Andalucía): iniciativa integral de ciberseguridad de la Junta de Andalucía, nacida dentro del Plan de Seguridad y Confianza Digital Andalucía 2020.

SLDC: Ciclo de vida de desarrollo de software, metodologías que constituyen un marco para la planificación, control y desarrollo del software.

SNMP (Simple Network Management Protocol): es un protocolo de la capa de aplicación, que facilita el intercambio de información entre dispositivos de red, permitiendo así a los administradores monitorizar y supervisar el funcionamiento de la misma.

Strings: herramienta en sistemas GNU / Linux para mostrar las cadenas presentes en un fichero binario.

TBC (**Trusted Base Computer)**: conjunto de elementos para implementar las medidas de seguridad requeridas y garantizar así la confidencialidad, integridad y disponibilidad de la información en un sistema.

Trusted IoT (IoT de confianza): Etiqueta que propone la Unión Europea que deben incluir los dispositivos IoT que han sido certificados como seguros. La agencia ENISA es la encargada de liderar dicha certificación.

Los Autores

Miguel Ángel Arroyo Moreno

Consultor de seguridad de la información en Blue Red Security (Grupo SEMIC) desde 2013, donde ha desempeñado labores de hacking ético, pruebas de intrusión y auditorías técnicas. Con más de 10 años de experiencia en el campo de la seguridad de la información, actualmente trabaja como consultor realizando auditorías de estado de seguridad, análisis de riesgos y planes directores de seguridad. Certificación CISA (Certified Information Systems Auditor) y miembro de APISA desde 2013. Autor del blog hacking-etico.com y fundador de la comunidad Hack&Beers. Es Presidente de la Asociación Nacional de Profesionales del Hacking Ético, y profesor en varios cursos de seguridad en diferentes universidades (UCLM, UEX y UNIA).

Josep Bardallo

Ingeniero Informático, responsable del área de Consultoría de seguridad de la información y de DevOps en Blue Red Security (Grupo SEMIC) y CISO del grupo Hospitalario Recoletas. Con más de 20 años de experiencia en el campo de la seguridad de la información, IT y Cloud. Dispone de diferentes certificaciones profesionales como CCSP, CISA, CISSP, CRISC, CISM, o CGEIT, siendo también auditor en ISO 27001 y miembro colaborador de asociaciones como ISACA, ISC2, ASIS, ITSMf, ENISA, CSA (miembro del comité técnico operativo, donde participa en diferentes iniciativas en seguridad,ioT, privacidad y Cloud). Forma parte de la junta de la Asociación Nacional de Profesionales del Hacking Ético, y profesor en varios cursos de seguridad en diferentes universidades.

Juan José Domenech Sánchez

Arquitecto de soluciones e ingeniero de software, dedicado desde 2012 a proponer, asesorar y desarrollar arquitecturas (de aplicación, de infraestructura, de integración y de operaciones), tecnologías y productos enfocadas a Smart Cities, IoT, seguridad y soluciones empresariales. CoLeader de OWASP Sevilla, y CiberCooperante de INCIBE.

Francisco Jesús Gómez López

Diplomado en Informática y Graduado en Ingeniería Informática por la Universidad de Córdoba. Experto en Gestión de Servicios TI basados en ITIL por la UNED y miembro de APISA desde 2009. Comenzó su carrera profesional ya en Sanidad como técnico de campo con INDRA (1999-2005) para posteriormente ser Responsable de Informática en el Distrito Sanitario Córdoba (2005-2011). Desde 2012 trabaja como técnico en el Hospital Reina Sofía y en la actualidad desarrolla tareas dentro de la línea provincial de Implantaciones, Proyectos e I+D en Córdoba. Colabora con IMIBIC como experto TI en proyectos europeos del programa H2020

Ramón Salado Lucena

Estudió Ingeniería Técnica Informática en la UNED. Certificado por el MIT en CyberSecurity: Technology, Application and Policy. Profesor de Seguridad Online en diferentes Másteres de la Universidad de Sevilla, la UNIA y la Cámara de Comercio de Sevilla, CoLeader de OWASP Sevilla, y CiberCooperante de INCIBE. He trabajado durante más 10 años en el sector de la CiberSeguridad, 5 de ellos en la Administración Pública. Actualmente CTO-Founder de BeeHackers y Analista de Seguridad en SmartSec.

indra
solutia
innovación y tecnología
Quantion
tecnologías de telecomunicación
Microsoft
Carestream
cisco
FUJITSU
PULSIA
TECHNOLOGIES
accentureconsulting
semic
Effective IT Solutions
Check Point
SOFTWARE TECHNOLOGIES LTD.
T · · Systems ·
Symantec
DELL EMC
isotrol
hp
new doors