Universidad de Buenos Aires
Facultades de Ciencias Económicas, Ciencias Exactas
y Naturales e Ingeniería

Maestría en Seguridad Informática

Tesis de Maestría

Diseño lógico de una aplicación para determinar el nivel de seguridad de un dispositivo IoT

Autora:

Diana Fernández Sánchez

Director de la Tesis:

Hugo Scolnik

14 de agosto, 2017

Cohorte 2014

Declaración jurada de origen de los contenidos

"Por este medio, la autora manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se responsabiliza de que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual".

_____

Diana Georgina Fernández Sánchez

DNI: 95303923

Resumen

El presente estudio pretende brindar un diseño lógico de una aplicación móvil para definir el nivel de seguridad de un dispositivo clasificado como Internet de las Cosas (IoT). Por tanto, esta propuesta pretende dar una solución sencilla a las inquietudes de seguridad sobre la amplia gama y nuevas tendencias de dispositivos conectados a internet (y entre sí), al proporcionar una interfaz clara al usuario inexperto que le prevenga sobre posibles vulnerabilidades presentes en el dispositivo que la aplicación móvil propuesta analizará.

A fin de enfocar esta investigación dentro de un área específica de estudio, primero se indagan los casos de ataques, debilidades y problemas de seguridad de IoT conocidos a la fecha. Después, se determinan cuáles dispositivos IoT dentro de esa área seleccionada son más populares, críticos y de mayor impacto, con la meta de definir un perfil funcional y de componentes para establecer qué factores de seguridad deben ser analizados por la aplicación.

Finalmente, con la información recolectada, se propone un documento de especificación de *software* con el diseño lógico de una aplicación móvil, cuyo fin es analizar si el comportamiento y funcionamiento de un IoT sigue los resguardos básicos de seguridad, para luego alertar o informar al usuario sobre los resultados del análisis.

Palabras claves: internet de las cosas, IoT, aplicaciones móviles, seguridad.

Índice General

Declaración jurada de origen de los contenidos i
Resumen ii
Índice General iii
Índice de tablas y figuras iv
Agradecimientos v
Nómina de abreviaturas vi
Introducción 1
Capítulo 1: Definición del área de estudio 3
1.1. Estado del arte del Internet de las Cosas 3
1.2. Situación actual de la seguridad en IoT 7
1.3 Enfoque de la investigación: análisis de criticidad e impacto 13
1.4 Soluciones u opciones similares a la propuesta 15
Capítulo 2: Precedentes, análisis funcional y estructural de los dispositivos IoT 17
2.1. Buenas prácticas, estándares y recomendaciones para el desarrollo de IoT. 17
2.2. Estadísticas sobre IoT en el sector industrial 19
2.3 Componentes, servicios, estructura y funcionamiento. 23
2.4. Análisis de similitudes y diferencias 30
2.5 Definición de factores de seguridad y métodos de verificación 31
Capítulo 3: **Requerimientos y diseño lógico** 33
3.1. Análisis de factores de seguridad y definición de medios de solución. 33
3.2 Propuesta del diseño lógico: documento de especificación del *software* 35
Capítulo 4: Análisis, recomendaciones y conclusiones. 56
4.1. Viabilidad del proyecto, recomendaciones y cumplimiento de objetivos 56
4.2. Análisis de los resultados, retos, iniciativas y conclusiones. 57
Glosario 59
Apéndice 63
Anexos 71
Bibliografía Específica 71
Bibliografía General 77

Índice de tablas y figuras

Lista de Tablas

Tabla 1 Resumen de recomendaciones de seguridad 18
Tabla 2. Principales elementos de ICS 22
Tabla 3. Clasificación y ejemplo de componentes según la NIST 800-82 27
Tabla 4. Protocolos más utilizados en ICS 29
Tabla 5. Factores de seguridad y métodos de verificación 31
Tabla 6. Descripción del modelo de dominio 40
Tabla 7. Descripción de los recursos de terceros y los factores de seguridad 46
Tabla 8. Texto del caso de uso 1 48
Tabla 9. Casos de Prueba para caso de uso 1 51
Tabla 10. Texto de caso de uso 2 51

Lista de Figuras

Figura 1. Capas de la composición de la tecnología IoT 4
Figura 2. Infograma sobre ciber riesgos asociados a IoT 9
Figura 3. Vulnerabilidades encontradas por SCADAhackers 22
Figura 4. Típico sistema ICS 25
Figura 5. Proceso típico de un ICS 26
Figura 6. Diagrama de un ataque típico a un ICS y las herramientas utilizadas 33
Figura 7. Diagrama de bloques del proceso completo de la propuesta 38
Figura 8. Modelo de dominio del sistema para la propuesta 40
Figura 9. Diagrama de Casos de Uso 41
Figura 10. Intercambio de información entre la App y otros elementos de la propuesta 43
Figura 11. Propuesta de las pantallas de la aplicación móvil (1 de 3) 44
Figura 12. Propuesta de las pantallas de la aplicación móvil (2 de 3) 44
Figura 13. Pantallas de error para diferentes casos y opciones del menú. 45
Figura 14. Uso de recursos externos contemplados en la propuesta. 45
Figura 15. Diagrama de actividades del caso de uso 1 49
Figura 16. Diagrama de estado del caso de uso 1. 50
Figura 17. Diagrama de secuencia del caso de uso 1 50
Figura 18. Diagrama de actividades del caso de uso 2 52
Figura 19. Diagrama de estado del caso de uso 2 53
Figura 20. Diagrama de secuencia del caso de uso 2 54

Agradecimientos

A mi familia y amistades, por la paciencia y el apoyo brindado al realizar esta Maestría.

A Gabriel Carro, por la información proporcionada durante el curso de Internet de las Cosas en la ECI 2016.

A Laura Delgado por la revisión filológica del documento y Arturo Corrales por las correcciones técnicas y de redacción.

A Hugo Scolnik, por sus aportes a la corrección, mejora del documento y tutoría durante la realización de esta tesis.

Nómina de abreviaturas

BPCS: *Basic Process Controllers,* Controladores Básicos de Procesos.

BMS: *Building Management System,* Sistema de gestión de edificios.

CERT: *Cyber Emergency Response Team,* equipo de respuesta a emergencias cibernéticas.

CVE: *Common Vulnerabilities and Exposures,* Vulnerabilidades y Exposiciones Comunes.

CoAP: *Constrained Application Protocol,* Protocolo de aplicación restringido.

DCS: *Distributed Control System,* Sistema de control distribuido.

FPGA: *Field Programmable Gate Array* o arreglos de compuertas programables en campo.

HMI: *Human Machine Interface,* Interfaz Hombre-Máquina.

ICS: *Industrial Control System*, sistemas de control industrial.

IDoT: *Identity of Things*, identidad de las cosas.

IED: *intelligent electrical device,* Dispositivo eléctrico inteligente.

ICS-CERT: *Industrial Control Systems Cyber Emergency Response Team,* área adjunta al Departamento de Estado de los Estados Unidos.

IO: *Input/Output*, entrada y salida de datos

IoE: *Internet of Everything*, el internet de todo.

IoT: *Internet of Things*, internet de las cosas.

ITU: *International Telecommunication Union,* Unión Internacional de Telecomunicaciones.

MAC: *Message Authentication Code*, Código de autenticación de mensajes.

M2M: *machine to machine*, se refiere a las comunicaciones entre dispositivos.

MQTT: Message Queuing Telemetry Transport

OSINT: Open Source Intelligence o inteligencia de fuentes abiertas.

PII: P*ersonal identifiable information* o información personal identificable.

PLC: *Programmable Logic Controller,* controladores lógicos programables

RTU: *Remote Terminal Unit,* Unidades Terminales Remotas

SCADA: *Supervisory Control and Data Acquisition (a subset of ICS),* Control de Supervisión y Adquisición de Datos (un subconjunto de ICS).

SIS: *Safety Instrumented Systems,* Sistemas instrumentados de seguridad.

VNC: *Virtual Network Computing,* computación en red virtual.

VPN: *Virtual Protocol Network,* protocolo de red virtual.

WSN: *Wireless Sensor Network*, Redes de sensores inalámbricos.

Introducción

El Internet de las Cosas (IoT por sus siglas en inglés) es un tema en auge, tanto en investigaciones de académicos y expertos en seguridad, como en el diseño de productos o áreas de innovación y desarrollo de empresas, esto debido a factores como las nuevas tendencias de interconectividad de los dispositivos, la digitalización de datos, la disponibilidad y procesamiento en la nube de la información, la paulatina adopción del protocolo IPv6, la mejora de las capacidades de bajo consumo, la vida útil de la batería y la disminución de tamaño de los componentes electrónicos.

De modo tal, el uso de dispositivos conectados entre sí y a Internet representa una tendencia cada vez más usual. Debido a esta creciente popularidad, se considera que es un tema que vale la pena estudiar, sobre todo desde el punto de vista de la seguridad informática, pues la información que estos dispositivos transmiten o manejan puede ser considerada sensible o útil para ataques. Por ejemplo, ya se han divulgado pruebas de concepto y existen casos de ataques relacionados con estos dispositivos, razón por la cual se considera importante y justificado proponer como tema de investigación la búsqueda de medios de prevención o contramedidas accesibles y claras al usuario común.

Aunado a esto, los dispositivos IoT son cada vez más intrusivos y manejan más información sensible, datos que están cobrando mayor valor en el mercado. A pesar de ello, los fabricantes no suelen incluir medidas básicas de seguridad en la transmisión, manejo y almacenamiento de estos datos. El usuario inexperto en los temas de seguridad informática desconoce estos peligros, confía en que el producto que compra cuenta con las medidas mínimas de seguridad y, sin el conocimiento técnico requerido, ignora cómo averiguar si el equipo es seguro.

Con base en el panorama descrito, la propuesta de trabajo facilita esta tarea al proporcionar una herramienta para alertar al usuario sobre el comportamiento de sus dispositivos de IoT, en cuanto a problemas de seguridad informática. El

aporte del proyecto consiste en materializar las buenas prácticas de seguridad informática en una herramienta que monitoree los dispositivos IoT de un usuario y notifique sobre filtraciones u otros problemas relativos a la seguridad. Se aclara que el alcance de la investigación es la creación del diseño lógico de una aplicación dedicada a establecer el nivel de seguridad de los dispositivos IoT del área de mayor impacto en la actualidad. Para ello, se pretende crear un documento de especificación de *software*, basado parcialmente en el estándar de la IEEE 830-1998 *Recommended Practice for Software Requirements Specifications*.

Aunque, el proyecto pretende establecer las bases de una aplicación, no se presentarán prototipos funcionales de esta, solo la documentación necesaria para que sea posteriormente implementada. También se recalca que la aplicación no tendrá funciones de anti-*malware*; solo se limitará a validar características inseguras en IoT a nivel de transmisión, manejo y almacenamiento de datos sensibles o detección de posibles vectores que permitan la infección del dispositivo, pero no se podrá categorizar la aplicación como anti-malware de ningún tipo.

Por otro lado, la hipótesis de la investigación es definir un marco de referencia de requerimientos mínimos de seguridad y comportamientos apropiados del dispositivo IoT, mediante la recolección de datos y buenas prácticas y, a partir de esta base, comparar los resultados del dispositivo del usuario que está bajo estudio o monitoreo y alertarle en caso de ser necesario.

El estudio se estructura en capítulos. Las dos primeras partes están dedicadas a la investigación y recolección de datos, tanto para definir el enfoque como para determinar los requerimientos necesarios de la aplicación. La tercera parte es la definición del diseño lógico, a partir del análisis de la información recolectada en las partes previas. La cuarta parte cierra el estudio, a través de la determinación de resultados, su análisis, recomendaciones y conclusiones.

Capítulo 1: Definición del área de estudio

1.1. Estado del arte del Internet de las Cosas

Definición de Internet de las Cosas

La "Internet de las cosas", o IoT por sus siglas en inglés, es descrita por la recomendación ITU-T Y.2060 de la ITU como una "infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras. Por la cual, gracias a la identificación, adquisición, procesamiento de datos y a las capacidades de comunicación, IoT hace pleno uso de los objetos para ofrecer servicios a todo tipo de aplicaciones; además, garantiza el cumplimiento íntegro de los requisitos de seguridad y privacidad" [1]. Sin embargo, dada la novedad de este tema, no existe una definición exacta y definitiva de IoT; por tanto, algunas entidades agregan factores como la diversidad de cosas[1], el énfasis en la poca intervención humana y su interacción con los dispositivos (comunicación autónoma) o la diferencia entre M2M e IoT, y otros.

La idea principal de IoT consiste en utilizar sensores y controles de distintos objetos y de diferentes plataformas, interconectados para reunir y analizar datos sobre el ambiente, los elementos de ese medio y quienes interactúan en él, con el fin de mejorar su comprensión y procesamiento; así como automatizar u optimizar procesos relacionados al objeto/ambiente de análisis.

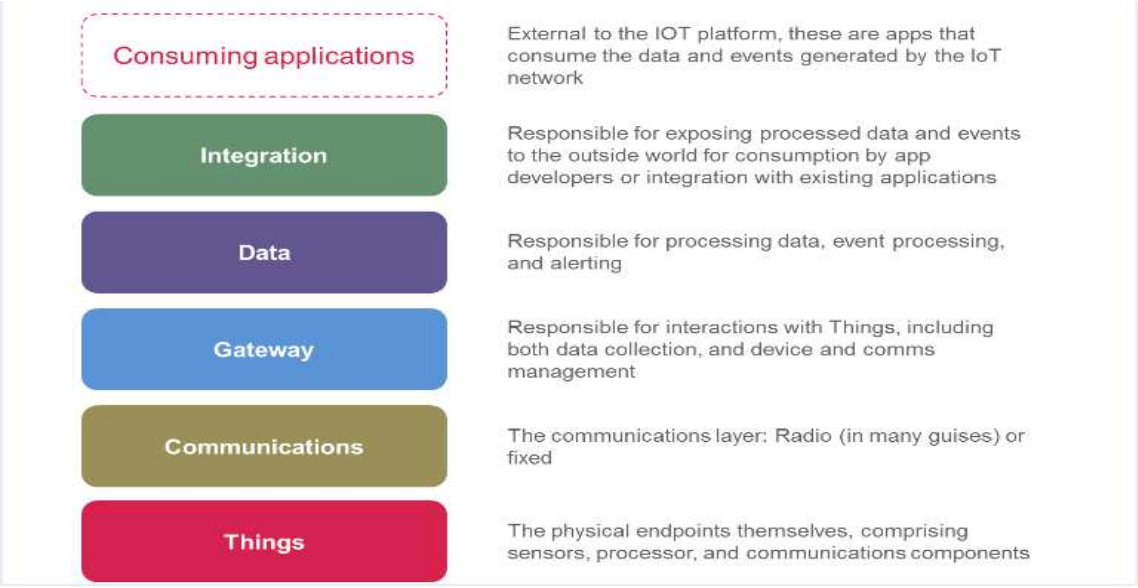
Adicionalmente, algunas características distintivas de los dispositivos IoT son: poseer un identificador único, autoconfiguración (*plug and play*), vincular e interactuar con otros equipos o conectarse a Internet. Además, por lo general,

[1] TechTarget lo ejemplifica al decir que una cosa en el mundo IoT puede ser desde una persona con un monitor cardíaco, un animal de granja con un biochip o un automóvil con sensores para medir el nivel de presión [2].

utilizan baterías primarias, y el usuario final suele recibir una retroalimentación o datos procesados por otra aplicación conectada al equipo IoT, con frecuencia almacenado en la nube.

Estructura básica

Hasta hoy no existe un estándar universal de fabricación ni regulación definido para IoT, sobre todo porque una de las principales limitantes es la poca interoperabilidad entre soluciones de distintos fabricante (existen recelos en el tema de acceso a las API y a las nubes y servicios), lo cual dificulta especificar la estructura de la tecnología IoT. A pesar de ello, a nivel general, Gary Barnett en un informe de Ovum, categoriza en seis capas la composición de la tecnología IoT, como se presenta en la Figura 1.



Fuente: [3].

Figura 1. Capas de la composición de la tecnología IoT.

El nivel más bajo corresponde a las “cosas”: sensores, cámaras, acelerómetros o controladores que recolectan los datos en los espacios donde se encuentran. El siguiente nivel está relacionado con a los dispositivos *gateway* o los de borde, quienes filtran y analizan los datos mientras se encargan de la comunicación, de los protocolos de las interfaces y de la seguridad. Luego, se

encuentra el nivel de Conectividad de Red, responsable del acceso a Internet, comunicaciones máquina a máquina o inalámbricas.

Por otro lado, la etapa de datos se refiere a la abstracción de estos, mediante métodos de agregación y filtración para el procesamiento de la información. La etapa de Integración busca exponer los datos procesados, ya sea por medio de servidores de datos en la nube (procesamiento, almacenamiento, redundancia) o aplicaciones de negocio (análisis, reporte, controles). La última capa corresponde a las aplicaciones externas a las plataformas IoT que utilizan los datos procesados para otorgar un proceso optimizado, también llamadas interfaces *back-end*.[2]

A partir de la selección de recursos de los desarrolladores de IoT, se puede inferir la estructura típica o esperada de un dispositivo IoT a nivel general. Si se toma como base la última encuesta realizada por Eclipse *IoT Working Group*, IEEE IoT y AGILE IoT, se determina que los siguientes son los recursos más utilizados al crear un dispositivo IoT:

- Sistema Operativo: Linux.
- Lenguaje de programación: Python, Java, JavaScript, C.
- Protocolo de mensajería: MQTT, HTTP, CoAP.
- Protocolos de conectividad: TCP/IP, Wi Fi, Ethernet, Bluetooth, Zigbee.
- Servicios de nube: Amazon, nube privada o *"on premise*", Microsoft Azure.

Otra tendencia que la encuesta detectó fue el uso de *hardware* y *software de* código abierto[3] así como plataformas o protocolos de mensajería personalizados o propios. Sobresale también en la encuesta, el uso de plataformas privadas (sobre todo para el área de automatización del hogar) como Google Nest, Apple HomeKit, SmartThings Hub (de Samsung) o Eclipse

[2] Se le suele denominar *front-end* a las aplicaciones móviles o interfaces web a las que el usuario tiene acceso para definir los parámetros.

[3] De los desarrolladores encuestados, 58% participan en proyectos *open source* y 52% utilizan *open hardware* para realizar el prototipo o para la implementación.

Smart Home. Estas opciones, de marcas reconocidas, demuestran el interés que dicha tendencia genera en el mercado actual [4].

Cabe resaltar que, indiferentemente de las tecnologías utilizadas, el Internet de las Cosas viene a aprovechar dos tendencias de interconexión: la comunicación entre máquinas (M2M), las redes de sensores inalámbricos (WSN) y el concepto de espacios sensibles.

Verticales y casos de uso

IoT constituye una tendencia que ha cobrado fuerza y se ha aplicado a múltiples industrias y servicios. Se resaltan las innovaciones en los sistemas de transporte inteligente; las redes de monitoreo; gestión de suministro de recursos energéticos y agricultura; los 'hogares, edificios, ciudades' inteligentes; la automatización industrial; los *wearables*; las iniciativas en sector médico (implantes biomédicos e instrumentos médicos interconectados); los servicios de emergencia y *fitness*, etc. Los sectores que reportan mayor auge en el tema son la industria manufacturera (15%), sanidad (15%) y los seguros 11% [5].

Tal ha sido el impacto de la mencionada aplicación de la tecnología, que se ha llegado a considerar que la conectividad será una característica estándar de los productos del futuro, y ya se plantea la expansión del concepto al del Internet del Todo (más conocido como IoE, por sus siglas en inglés), donde las cosas toman conciencia de su contexto, adquieren una mayor potencia de procesamiento y una mayor capacidad de detección [6].

En el presente, se estima que un 99% de las cosas en el mundo físico aún no están conectadas a Internet [7], pero se espera que para el 2020 la base instalada de dispositivos IoT alcance los 26 billones de unidades[4], gracias al bajo costo de los componentes, la demanda de estos productos por parte de los usuarios, y el interés de inversión de las empresas.

[4] De acuerdo al reporte de Gartner '*Forecast: The Internet of Things, Worldwide*, 2013' [7], pero según el Federal Trade Commission, este conteo puede elevarse a 50 billones [8].

Varias entidades[5] han categorizado los casos de uso del Internet de las Cosas (actuales y futuras), clasificación que se sintetiza en los siguientes escenarios de uso:

- Manejo del estado de los activos mediante la gestión, monitoreo, localización, rastreo y mantenimiento. Se incluyen aspectos de seguridad y riesgo.
- Medio para lograr la optimización de recursos.
- Fundamentar el modelo de negocio del cobro por el uso del activo sobre una base incremental, es decir, la monetización de un activo físico, midiendo con precisión su uso.
- Mejorar la experiencia del usuario de manera efectiva y eficiente con la información recolectada.
- Mejorar o actualizar un producto o servicio con menos intervención humana.
- Automatización de tareas.

Inclusive, empresas consolidadas están añadiendo opciones de soporte a IoT en la oferta regular de sus productos. Por ejemplo, SAP integró esta tendencia con su nueva área comercial SAP IoT, la cual se dedica a proveer apoyo a los clientes que utilizan sistemas basado en IoT, pero pretende en un futuro brindar servicios de implementaciones de IoT de SAP [10].

Por otro lado, el interés económico alrededor de esta nueva tendencia también está ayudando a adoptar la tecnología IoT. Según un reporte de McKinsey Global Institute sobre el impacto global económico de las aplicaciones de IoT, se espera que para el 2025 el impacto sea de entre 4 a 11 trillones de dólares, encabezando la lista las iniciativas en el área industrial, *smart cities* y bienestar humano (salud); este último con un 40% de impacto global [11].

1.2. Situación actual de la seguridad en IoT

Como lo indica la ley de Metcalfe[6], las redes se vuelven exponencialmente más valiosas a medida que el número de usuarios aumenta.

[5] Gartner las clasifica en cuatro categorías [5] pero SAP en tres categorías [9].

Razón por la cual, la expansión de dispositivos IoT conlleva a que este sector se convierta en un punto de ataque valioso a corto o mediano plazo. Aunado a este hecho, la información que tales aparatos pueden manejar (datos médicos, datos financieros, hábitos de los usuarios, entre otros) cobran cada día más valor.

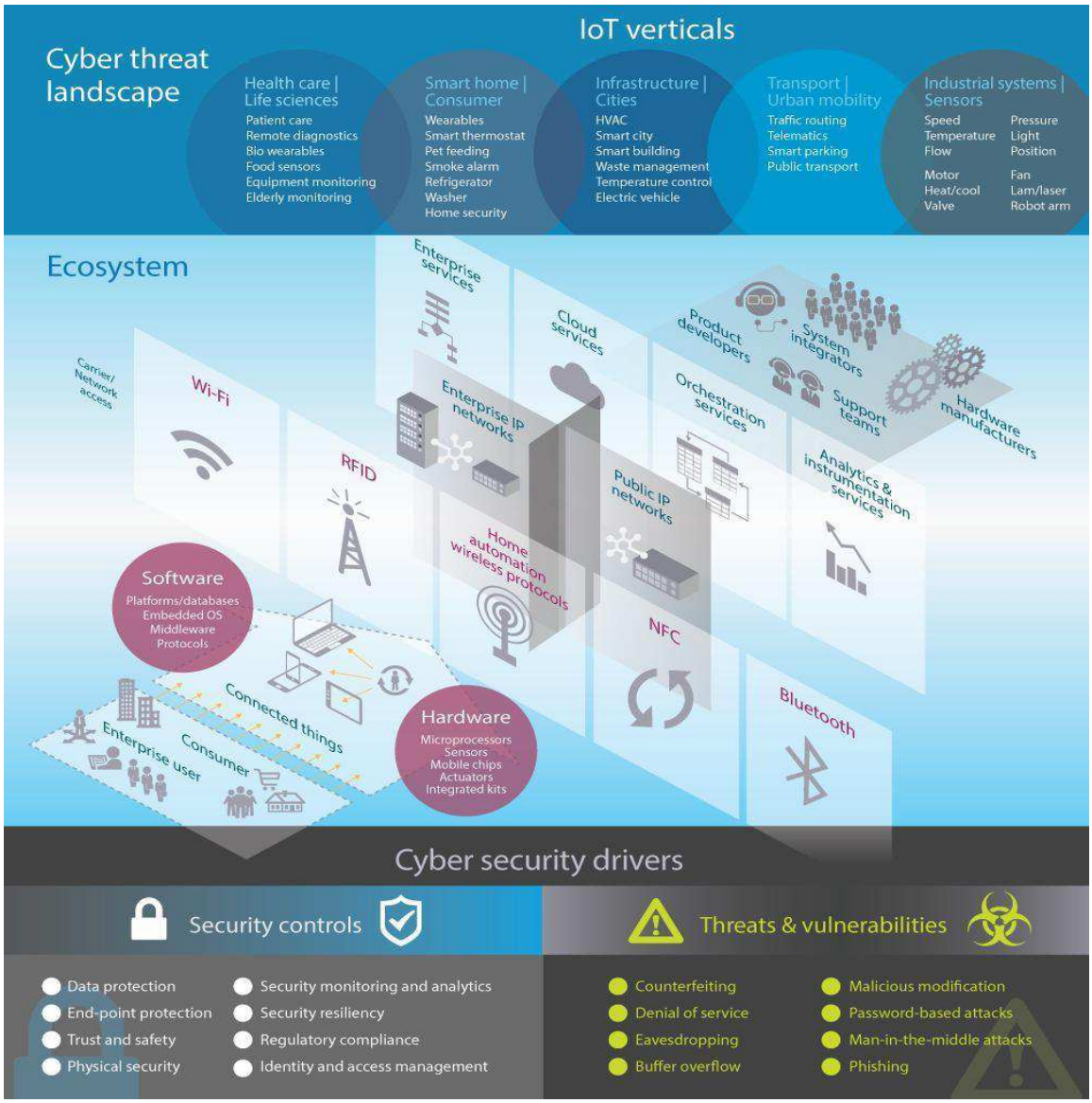
El aspecto de la privacidad de datos y quién es dueño de estos, es importante para cualquier tecnología, pero dada esta nueva tendencia donde se recolectan cantidades masivas en todos los ámbitos posibles, el tema obtiene una dimensión aún más crítica. Los datos permiten el acceso a ataques de ingeniería social, suplantación de identidad, accesos no autorizados, espionaje masivo, y demás.

En el ámbito empresarial o industrial, el tema se complica si los dispositivos IoT tienen acceso a la red corporativa. De acuerdo con Andrew Hay, investigador de seguridad de OpenDNS, existen dos riesgos asociados a esta práctica [12]:

- La creación de un nuevo vector y expansión de la superficie de ataque. Los dispositivos IoT orientadas al consumidor pueden ser manipulados para dar a los atacantes nuevas vías para explotar las redes empresariales de forma remota.
- Los dispositivos IoT podrían estar vinculados a servicios en la nube controlados por atacantes o administradores incompetentes, que podrían no parchear vulnerabilidades rápidamente, lo que resulta en las credenciales de usuario en peligro y la exposición de la información almacenada.

El infograma de la Figura 2, proveniente del reporte de Deloitte sobre ciber riesgos relacionados con el Internet de las Cosas, sintetiza de manera efectiva la situación enfocada en el tema de seguridad.

[6] La ley de Metcalfe establece que el valor de una red aumenta proporcionalmente al cuadrado del número de usuarios. [6]



Fuente: [13]

Figura 2. Infograma sobre ciber riesgos asociados a IoT.

Como se indica en la figura 2, otro punto por considerar radica en que, debido al aumento exponencial de dispositivos conectados a Internet que se está generando con la expansión de IoT, se han incrementado los ataques de denegación de servicios; es decir, más dispositivos conectados equivalen a más

equipos esclavos o bots para perpetrar un ataque de este tipo[7]. Además, debido a la movilidad ubicua y las cantidades masivas de dispositivos IoT, las labores de medidas de seguridad como seguimiento/gestión de equipos, resguardo de perímetro o actualización/mantenimiento de los mismos, se dificultan.

De hecho, como parte de los resultados de la IoT Developer Survey 2016, se determina que la seguridad es una de las principales preocupaciones que, junto con la interoperabilidad y la conectividad, desvelan a los desarrolladores de estas iniciativas. De acuerdo al especialista Rob Black [14] existen seis componentes relativos a la seguridad en IoT:

1. los dispositivos (que incluyen desarrollo, gestión y su seguridad),
2. la comunicación dentro y fuera del *firewall*,
3. las operaciones en la nube,
4. las plataformas en la nube,
5. las aplicaciones en la nube,
6. el desarrollo de las nubes.

Dentro de estos componentes, cabe resaltar preocupaciones concernientes al manejo y control de *software* de terceros [15], los datos que se recolectarán y la privacidad de ellos, así como los problemas que conllevan la movilidad de los equipos (casos de robo de equipos con datos sensibles o accesos a una red).

Otro factor de riesgo lo constituyen las interfaces *Front-End* (aplicaciones móviles o web) que se utilizan con los dispositivos IoT. Según una investigación de HP [16], se estima que el 80% de los dispositivos estudiados, no contaban con suficientes medios de autenticación y el código fuente de las aplicaciones no poseía ofuscación (es posible obtener las credenciales y claves de cifrado). Otro reto es actualizar regularmente los dispositivos IoT con arreglos y actualizaciones de seguridad. Desplegar parches de seguridad para todos los

[7] El 27 de setiembre del 2016 se dio a conocer la noticia del ataque DDoS a la empresa OVH por medio de una Botnet de 145 000 cámaras de seguridad, el cual generó un ataque de 1,5 Tbps (sumado todo el ancho de banda recibido).

dispositivos puede ser muy difícil en redes de dispositivos poco fiables o de ancho de banda bajo, y muchas de las medidas de seguridad existentes, como la seguridad de los navegadores Web, pueden no estar disponibles para las aplicaciones IoT.

Asimismo, se debe tomar en cuenta el desarrollo o mejora de mecanismos de seguridad en protocolos dedicados o creados para esta tecnología (como MQTT y CoAP) [17].

Por último, según las predicciones del reporte de Juniper Research, a mediano plazo, es probable que IoT presente un objetivo mucho más importante para los ataques que los hoy existentes, debido a la adopción generalizada, el aumento de las capacidades de los dispositivos y la interconexión con los sistemas que actualmente son atacados.

Hasta el momento, se conocen varios casos de malware que puede afectar tanto a *routers* como dispositivos IoT [18]. Nombrado KTN-Remastered o Reimaten, fue descubierto por ESET y se describe como una combinación de un bot IRC llamado Tsunami (usado para ataques DDoS) y el troyano Gafgyt, el cual realiza un escaneo del protocolo Telnet y busca equipos vulnerables. Una vez conectado, el troyano intenta adivinar las credenciales (en caso de configuraciones débiles). Si logra acceder al equipo, se ejecuta una Shell para descargar ejecutables del bot. Otro malware identificado que podría afectar estos dispositivos, es el gusano Linux.Darlloz, el cual infecta equipos para la minería de criptomonedas como Dogecoin y Mincoin [19]. Enfocado en atacar equipos de arquitectura Intel x86, ARM, MIPS y PowerPC que utilicen el sistema operativo Linux, el gusano se difunde al iniciar un servidor web HTTP en el puerto 58455 y abrir un *back door*; de modo tal, permite la descarga de archivos a través de este puerto. En los casos registrados hasta el momento, el gusano descarga el software CPUMINER, dedicado a la minería de criptomonedas que utilizan el algoritmo *scrypt*. Por ahora, el ataque afecta sólo a sistemas x86 por los requerimientos de memoria y procesamiento de CPU, pero no se descarta la posibilidad de que se adapte a equipos IoT.

Sumado a lo anterior, el Laboratorio de Investigaciones Level3 identificó una pequeña familia de malware denominados Lizkebab, BASHLITE, Torlus y Gafgyt (ya comentado) con la cual se ha construido una gran botnet que está siendo utilizada para llevar a cabo ataques de Denegación de Servicio[8]. Esta familia de malware se aprovecha del uso de credenciales por defecto de los equipos, de protocolos inseguros (como Telnet) y de las vulnerabilidades no parcheadas para acceder a los dispositivos, con el fin de bajar y ejecutar la aplicación maliciosa.

Según indican los investigadores de Level3, la mayoría de los puntos finales infectados están vinculados a un puñado de empresas que han implementado normas de seguridad ineficientes en sus dispositivos IoT.

Otro malware conocido es JS_Jiton, descubierto por Trend Micro el cual utiliza Javascript y ataca tanto dispositivos móviles (smartphones, tablets, etc) como dispositivos IoT. Este malware ejecuta una serie de rutinas al router, con el fin de modificar sus servidores DNS. Al igual que los otros casos descritos, este malware aplica un ataque de fuerza bruta con una lista de credenciales por defecto, para ganar acceso al router o dispositivos.

Por otro lado, un caso famoso de prueba de concepto de ataque a un equipo IoT, fue realizado por el investigador de seguridad argentino César Cerrudo, quién en el 2014 demostró que era posible acceder al sistema de tránsito de la ciudad de Nueva York al interceptar las comunicaciones no cifradas y sin mecanismos de autenticación entre los componentes del sistema (sensores, puntos de acceso y central de control). Igualmente, el caso de Jesús Molina, quien en el DEFCON 22 del 2015, demostró que era posible tomar control de todos los dispositivos de los cuartos de un hotel en China, al aprovechar vulnerabilidades en el protocolo de *home automation* KNX [21].

[8] Para septiembre 2016, investigadores de MalwareMustDie! encontraron la evolución de esta familia de troyanos: Mirai. Este troyano tiene como meta atacar los equipos con firmware Linux Busybox y se centra en plataformas como ARM, ARM7,MIPS, PPC,SH4, SPARC and x86 [22].El 01 de octubre del 2016 el código fuente de Mirai fue liberado y para el 21 de octubre del 2016, la botnet formada por Mirai fue la causa del ataque de DDoS contra la empresa de servicios de DNS Dyn.

Los supradescritos son solo algunos ejemplos de malware, pero el uso de motores de búsqueda (como Shodan) han aumentado la visibilidad de los peligros relacionado a IoT, pues permiten a cualquier persona encontrar equipos IoT conectados a Internet y confirmar la falta de métodos de seguridad básica en estos.

Múltiples pruebas de concepto similares para casos de sensores, equipos inalámbricos y dispositivos conectados a internet (desde casos de acceso a cámaras web, Smart TV, monitores para bebés, CCTV, etc.) han sido publicadas, lo cual demuestra las debilidades en esta tendencia.

1.3 Enfoque de la investigación: análisis de criticidad e impacto

Durante los últimos años, debido al aumento de ataques relacionados, la preocupación en temas de seguridad se ha centrado principalmente en accesos remotos no autorizados a equipos conectados que propician la ejecución de acciones no permitidas como filtraciones de datos.[9] Esta es una situación que sucede con los sistemas actuales y que la expansión de los dispositivos IoT incrementará.

Según un estudio de Dell SecureWorks [23], resulta posible estimar el valor de la data robada en aproximadamente $ 1 por récord de números de tarjetas de crédito con un código CVV, $ 20 a $ 200 para una cuenta de PayPal con un balance verificado, $ 1.000 para una cuenta bancaria en línea y por las credenciales de seguro de salud, puede variar de $ 20 cada uno a $1.000 cuando está empaquetado con otra información de identificación personal por cada paquete[10]. Además, gracias al uso de Big Data, se puede concluir que todo dato recolectado tiene relevancia en el tema de seguridad [24].

Asociado al valor de los datos, cabe resaltar el grado de seguridad que se aplica en cada sector. Durante el 2015 y parte del 2016, los casos de ataques al

[9] Sector salud (registros médicos de pacientes), retail/bancario (datos financieros clientes), el industrial (secretos comerciales).

[10] En junio 2016 se publicó la noticia de la puesta en venta de una base de datos clínicos de 9.3 millones de pacientes en la Dark Web a un precio de 151BTC (~100,000$) a 607BTC (~395,000$).

sector médico e industrial se incrementaron debido a la debilidad en las medidas de seguridad que estas áreas implementan. Estos sectores resultan más vulnerables, pues carecen de una cultura exigente de seguridad como la tienen otras áreas, por ejemplo, la Bancaria o *Retail*.

Con respecto al impacto, si bien no se ha cuantificado por sector en caso de un ataque con dispositivos IoT, es posible estimarlo con las posibles consecuencias que se generarán en caso de suceder. Por ejemplo, según un estudio presentado por Kaspersky [25], las clínicas u hospitales son potenciales objetivos de ataque debido a la valiosa información y costosos equipos que manejan. Entre los peligros que el estudio enlista se resaltan:

- El uso delictivo de los datos personales de los pacientes: la reventa de información a terceros o exigir a la clínica que pague un rescate para recuperar información sensible de los pacientes.
- La falsificación intencional de los resultados de los pacientes o diagnósticos.
- El daño al equipo médico puede causar daño físico a los pacientes y enormes pérdidas financieras a una clínica u hospital.
- Impacto negativo en la reputación de una clínica u hospital.

Con respecto a los puntos mencionados, estos se pueden aplicar a cualquier sector afectado por un ataque de seguridad, puesto que los peligros de corrupción de datos, extorsión, daños y pérdida de reputación o de imagen aplican para todas las áreas.

Estudios similares han sido realizados en ambientes industriales, tanto para empresas que utilizan equipos SCADA (*Supervisory Control and Data Acquisition*), las cuales administran infraestructura crítica, como compañías que utilizan equipos de menor rango o responsabilidad.

Asimismo, casos conocidos de acceso no autorizados se han registrado en plantas nucleares (el famoso asunto de Stuxnet en la planta de Irán y el último caso en Alemania), el robo de datos con el troyano Regin a empresas de

telecomunicaciones, aerolíneas, empresas de energía, etc. Todos los casos descritos son relativos a ataques en los sistemas actuales (PC, *smartphones*, equipos industriales, médicos y de entretenimiento, etcétera); no obstante, se puede extrapolar esta tendencia al IoT pues se considera que, a mediano plazo, estos dispositivos tendrán similar interconexión con estos equipos, convirtiéndolos en posibles vectores de ataques y puertas de entrada a las redes.

En consecuencia, debido a la carente o mínima adopción de medidas de seguridad y al impacto que un posible ataque puede tener, se selecciona como enfoque de estudio el área industrial o la denominada *Industrial Internet of Things*, principalmente la refererida a gestión de infraestructuras críticas.

1.4 Soluciones u opciones similares a la propuesta

Existen diferentes grupos de investigación, consorcios, alianzas e iniciativas de estandarización relativas al desarrollo seguro, *compliance* y auditoría de dispositivos y recursos relacionados al Internet de las Cosas.

Entre estas propuestas, la más cercana a la opción por desarrollar en este proyecto, es 'The Project' de Opale Security, quienes proponen diseñar, producir y brindar una herramienta para realizar pruebas de penetración a nivel de hardware y software, por medio de la creación del marco 'Hardsploit' para auditar la seguridad en los dispositivos IoT [26]. Es una iniciativa muy interesante, que utiliza una plataforma modular con una FPGA para realizar las pruebas de seguridad en las interfaces de comunicación en dispositivos embebidos.

Según los creadores de la herramienta preindicada: "El objetivo es proporcionar una herramienta equivalente a los de la empresa Qualys o Nessus (Vulnerability Scanner) o el marco de Metasploit, pero en el ámbito de los sistemas embebidos".

Otra opción consiste en la propuesta por Beyond Security's, con la herramienta de auditoría tipo *fuzzer blackbox* llamada 'beSTORM' [27], la busca

las vulnerabilidades de seguridad en los protocolos utilizados por el dispositivo IoT.

Ambas propuestas se enfocan a un nivel empresarial y para ser utilizado por especialistas del área de seguridad informática; razón por la cual se considera que no afectan la propuesta de esta tesis, ya que pretende centrarse en una opción a los usuarios no especializados.

Capítulo 2: Precedentes, análisis funcional y estructural de los dispositivos IoT

En este capítulo se definen los factores de seguridad por evaluar por medio del diseño de la aplicación. Para lograr este objetivo, primero se determinan los lineamientos básicos de seguridad; luego, se filtran los datos de los dispositivos más utilizados en el sector con mayor criticidad, con el fin de poseer un modelo de verificación basado en el caso con mayor impacto.

2.1. Buenas prácticas, estándares y recomendaciones para el desarrollo de IoT.

Para lograr la masificación de la tecnología IoT es necesario la homogeneización de ciertos aspectos técnicos para garantizar la interconectividad y la interoperabilidad, condición que se logra con la aplicación de estándares y acuerdos entre las partes involucradas. Además, es necesario que exista desde el inicio del diseño de las iniciativas IoT, la infraestructura de seguridad, privacidad y controles generales.

De hecho, como parte de las recomendaciones dadas por el World Economic Forum y Accenture en el reporte sobre *Industrial Internet of Things* en el 2015 [28], se incentiva a la actualización de las políticas y regulaciones para que funcionen acorde a los nuevos modelos propuestos con IoT, así como a la creación de entes reguladores.

Además, se recomiendan solicitar a los proveedores de tecnología por inventariar y compartir las mejores prácticas de seguridad, con el fin de establecer un marco común global, mientras que al sector industrial, gubernamental y académico, se les encomienda enfocar su área de estudio a buscar soluciones a los problemas de seguridad, compatibilidad e interoperabilidad de la tendencia IoT.

Con el propósito de alinear el proyecto por desarrollar con estos aspectos, se presenta una recolección de las principales recomendaciones de seguridad relacionadas con IoT emitidas hasta la fecha. Para ello, se utiliza como punto de referencia las recomendaciones dada por medios especializados como

empresas de anti-malware o entidades gubernamentales, convenios industriales o comerciales entre empresas, foros, propuestas de estándares o marcos, entre otras fuentes de información. Se examina brevemente en la Tabla 1 las recomendaciones de seguridad, a partir de la información recolectada en el Apéndice 1.

Tabla 1 Resumen de recomendaciones de seguridad

Uso recursos	**Seguridad mínima**
● Garantizar la habilitación sólo de recursos o servicios necesarios. ● Garantizar un diseño efectivo y seguro de la arquitectura del dispositivo ● Si el usuario tiene acceso al código, este se debe ofuscar. ● Garantizar el uso de protocolos seguros actualizados o parcheados. ● Proteger acceso a dispositivos tanto desde redes externas como internas. ● Dispositivos sólo deben utilizarse en las redes de control y nunca deben cruzarse con la red corporativa.	● Reglas de autorización y control. ● Mecanismos de acceso y autenticación robustos. ● *Booteo* seguro. ● Medidas básicas de seguridad en API y nubes asociadas. ● Políticas de seguridad sobre equipos que interactúan con dispositivos IoT. ● Evitar credenciales hardcodeadas en firmware o código fuente. ● Mecanismos de *backup* y recuperación en caso de desastres. ● Evitar la seguridad por la oscuridad. ● Configuración adecuada de reglas de firewall (NIST 800-82).
Información al usuario	**Gestión datos**
● Informar sobre riesgos asociados. ● Solicitar cambio de contraseñas por defecto. ● Brindar servicio de ayuda a cliente para adopción de medidas de seguridad. ● Enfatizar prevención relativa a métodos de ingeniería social: *spear-phishing,* etcétera.	● Solo recopilar los datos necesarios. ● Garantizar que no se transfieran sin autorización datos a terceros. ● Segregación adecuada del almacenamiento. ● Comunicaciones seguras y anonimización de datos cuando sea posible. ● Almacenamiento seguro y a prueba de modificaciones no autorizadas. ● Validación de la integridad de datos.

2.2. Estadísticas sobre IoT en el sector industrial

Aclaración sobre IIoT e ICS/SCADA

Industria Internet of Thing o el Internet de las Cosas enfocado a la industria está estrechamente relacionado con los sistemas de control industrial, porque aprovecha los sensores y otros elementos ya implementados para el monitoreo de los sistemas; con el fin de generar un valor agregado a la data recolectada. Es decir, IIoT es una mejora de las actuales ICS. Por este motivo, se utiliza información referente a ataques y estadísticas de ICS para el análisis de este apartado.

Precedentes en temas de seguridad

Según un reporte presentado por Laboratorios Kaspersky [29] en julio del 2016 con el uso de técnicas OSINT, se obtiene el siguiente panorama:

- La mayoría de los *hosts* con componentes ICS disponibles en remoto se encuentran en EEUU y Europa.
- Casi el 92% de los Sistemas de Control Industrial (ICS) están expuestos a ciberataques.
- La cantidad de *hosts* ICS vulnerables accesibles desde el exterior, que posiblemente pertenezcan a grandes organizaciones, es de 12.483 (91,1%), de los cuales 453 (3,3%), pertenecientes a las industrias de energía, transporte, gas, ingeniería, procesamiento de alimentos y bebidas, contienen vulnerabilidades críticas.
- El número de vulnerabilidades se ha multiplicado por diez en los últimos cinco años. En el 2015 se publicó un total de 189 vulnerabilidades en componentes ICS, y la mayor parte eran críticas (49%) o de riesgo medio (42%).
- La mayor cantidad de vulnerabilidades se detectaron en dispositivos Siemens, Schneider Electric y Hospira.

- Las vulnerabilidades en los componentes ICS tienen otra naturaleza. Los tipos más conocidos son los desbordamientos de buffer (9% de todas las vulnerabilidades detectadas), el uso de contraseñas predeterminadas (7%) y el *cross-site scripting* (7%).
- No todas las vulnerabilidades detectadas en 2015 han sido reparadas. Para el 85% de las vulnerabilidades publicadas hay parches y nuevo firmware disponibles. Las restantes no han sido reparadas o lo han sido sólo de forma parcial. La mayoría de las vulnerabilidades sin reparar, catorce de diecinueve, son de alto riesgo.
- Los componentes de ICS más vulnerables son Interfaces Hombre Máquina (HMI), dispositivos eléctricos y sistemas SCADA.

Este reporte confirma los resultados del Proyecto SHINE [11], el cual reveló que al menos dos millones de dispositivos relativos a ICS, estaban expuestos públicamente en Internet[12]. Debido a que muchos de estos dispositivos utilizan protocolos creados antes de la expansión del Internet, no poseen mecanismos de autenticación.

Por otra parte, los casos más conocidos de ataques a este sector son los causados por el malware BlackEnergy3 -culpable del corte eléctrico en Ucrania en el 2015-, y Havex, el cual ocasionó problemas en fábricas en Francia y Alemania.

Asimismo, un caso particular es el reportado por Trend Micro, sobre un troyano bancario que se disfrazaba como actualización de un software de actualización de ICS [30]. Sin olvidar los accesos no autorizados a reactores nucleares en Irán y Alemania, e inclusive el intento de saboteo de una planta potabilizadora de agua en Reino Unido.

Entonces, el tema de la ciberseguridad industrial está en auge. Blogs dedicados al tema de la seguridad ha demostrado mediante pruebas de

[11] SHodan INtelligence Extraction, proyecto que recolectó datos de ICS con el buscador SHODAN desde el 2012 al 2014.
[12] Después de la revisión por parte del ICS-CERT del Departamento de Estado de USA, la lista se redujo a 7200 dispositivos calificados como críticos.

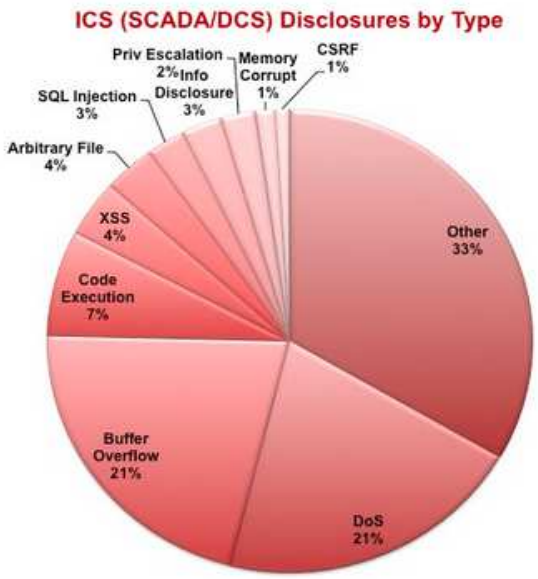
concepto cómo es posible realizar ataques de *man-in-the-middle* o de fuerza bruta para conseguir las claves de acceso (si existen métodos de autenticación). Incluso, han demostrado fallas como el intercambio de claves en texto plano, el uso de credenciales por defecto, *password hardcodeadas*, sistemas desactualizados y no fortificados,[13] entre otros [31].

Según Jordi Ubach, colaborador del blog de seguridad de Chema Alonso de Eleven Path, los principales ataques son:

- Ejecución remota de código.
- Extracción arbitraria de ficheros.
- XSS en varios de los sistemas.
- Denegación de servicio (alto porcentaje).
- *Buffer overflow* o desbordamientos de buffer.
- Restricción incorrecta de operaciones de la pila.
- Inyección SQL en plataformas web (del sistema o equipo).

Esta afirmación se confirma, tanto con los resultados del informe de Kaspersky [29], como con los resultados de SCADAhackers, los cuales se muestran en la Figura 3.

[13] Sistemas adjuntos al ICS como servidores (web, etc.) o servicios.



Fuente: SCADAhacker.com

Figura 3. Vulnerabilidades encontradas por SCADAhackers.

Protocolos, componentes y dispositivos IIoT con mayor visibilidad

Se profundiza en el reporte de Kaspersky [29] para determinar los protocolos, componentes y agentes inseguros de los dispositivos IIoT accesibles remotamente. Este análisis se sintetiza en la tabla 2, donde se detallan los principales elementos de los ICS en términos de seguridad.

Tabla 2. Principales elementos de ICS

Agente	Ejemplos
Componentes de la arquitectura ICS más comprometidos.	-Supervisory Control and Data Acquisition Systems (SCADA) -Distribution Control System (DCS).
Endpoint Críticos.	Dispositivos embebidos (PLC, RTU, IED, BPSC, SIS, paneles de operación), workstations, HMI e *historian*

	controllers, sistemas de control de comunicación, aplicaciones móviles.
Protocolos inseguros accesibles remotamente	HTTP, Niagara Fox, Telnet, EtherNet/IP, Modbus, BACnet, FTP, Omron FINS, Siemens S7.
Vulnerabilidades más expandidas.	-Sunny WebBox Hard-Coded Credentials (CVE-2015-3964) -Omron CJ2M PLC (CVE-2015-1015 y CVE-2015-0987) -Hard-Coded Credentials in Westermo Falcon y Lynx -Adcon Telemetry A840 multiples vulnerabilities -Siemens simatic S7-300 CPU DoS.
ICS accesibles remotamente por marca.	Tridium, Sierra Wireless, Beck IPC, Digi International. SMA, Siemens, Moxa, HMS industrial network AB, Lantronix, Westermo, Freescale, Rockwell Automation, TAC AB, Schneider Electronic, Conel, NetModule, Eister Energy ICT, Phoenix contact.

2.3 Componentes, servicios, estructura y funcionamiento.

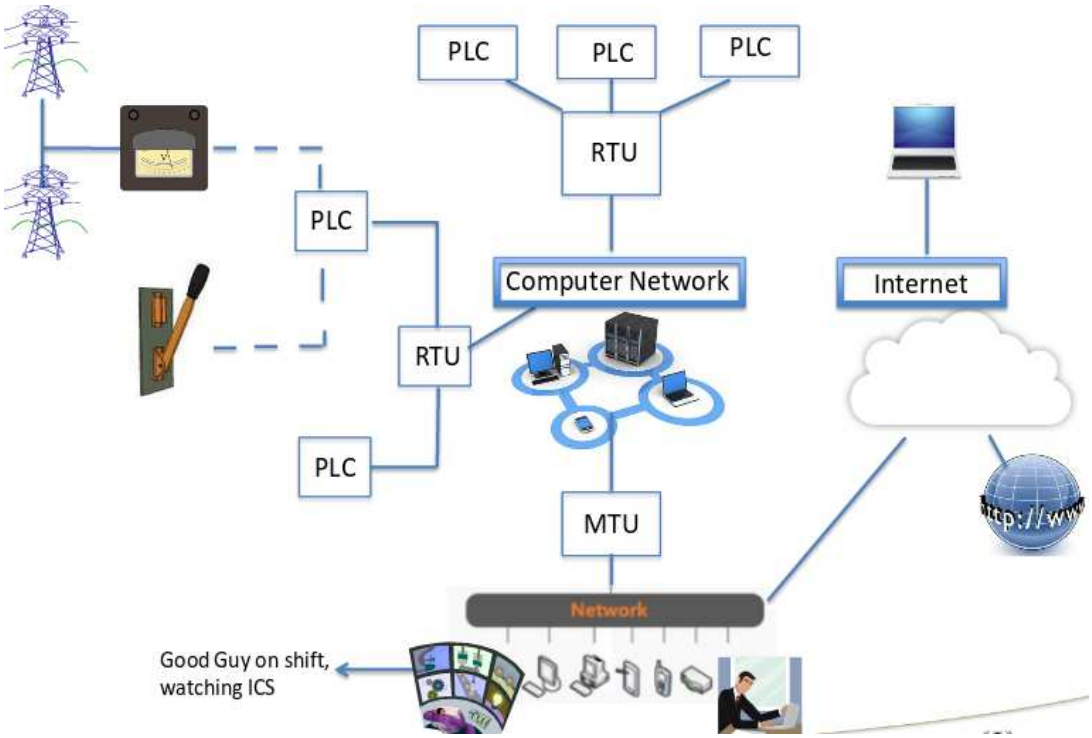
Se enfoca el estudio en los ICS y la subcategoría SCADA, pues implican las infraestructuras más críticas. Con la información analizada, se selecciona como área prioritaria la energética y proveedora de servicios básicos (reactores nucleares, plantas eléctricas, potabilizadores de agua, entre otros), pues son las industrias más afectadas con componentes más expuestos y cuyo impacto, en caso de ataque, es mayor.

Arquitectura y operaciones básicas de ICS

De acuerdo con el CERTSI [32], existen tres niveles principales en la arquitectura básica de los ICS, los cuales se pueden equiparar con las capas generales de los dispositivos IoT que se mencionaron en el capítulo 1.

- Campo y Control: contiene los RTU's, PLC's, IED's, sensores y actuadores finales, así como las comunicaciones entre ellos. A nivel de estructura de IoT, se puede considerar como la capa más baja o Física, la de las 'cosas'.
- Comunicación: contiene la interfaz de comunicación que conecta el nivel de campo y control con el nivel de supervisión. Es decir, la capa de *Communication y Gateway*.
- Supervisión: puede incluir las estaciones de trabajo (*workstation*), servidores SCADA (MTU's), servidores OPC, servidores de datos, etc. Este nivel se podría equiparar con las capas finales de la Figura 1. En el ámbito de la automatización industrial, esta etapa se puede dividir en dos: en Operaciones (que incluye *software* de visualización y supervisión de dispositivos, almacenamiento de datos, cálculo y paquetes de automatización) y Negocios (donde se encuentran las aplicaciones de visualización, análisis y reporte de datos para la generación de métricas y soporte para la toma de decisiones).

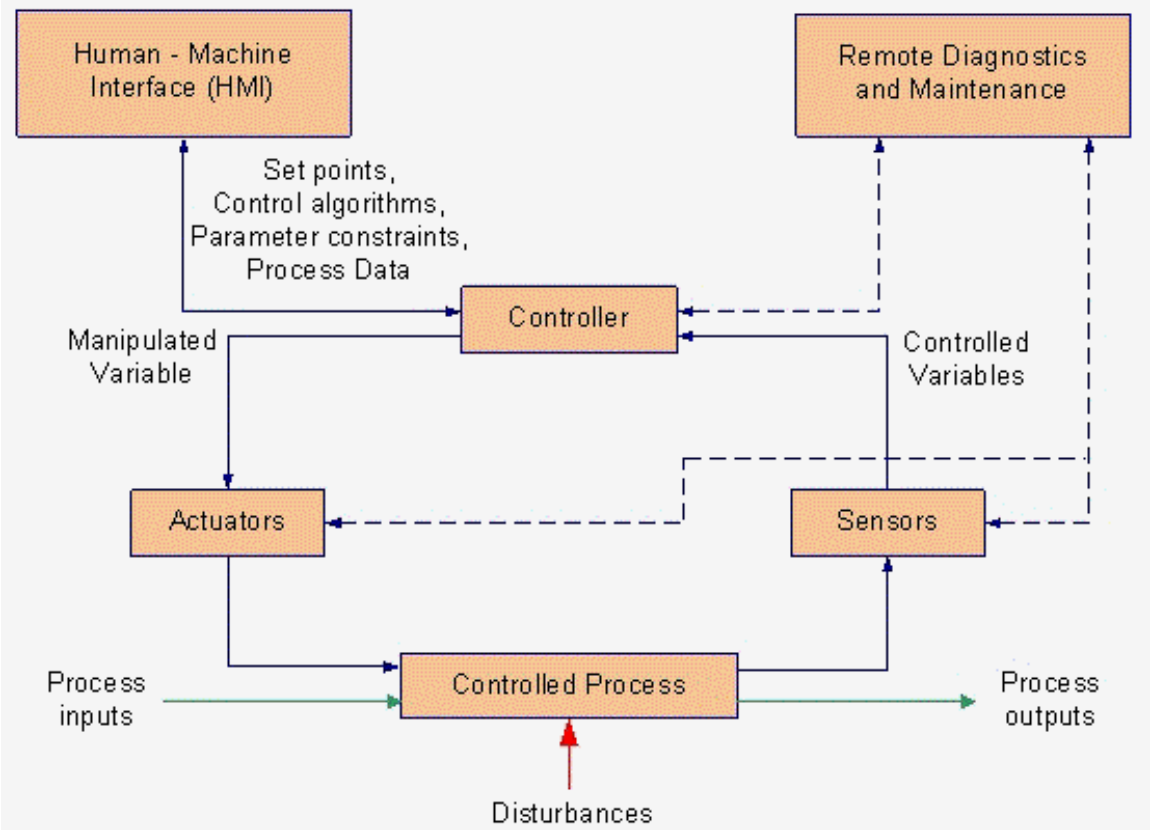
Con el propósito de lograr una mayor comprensión de la interacción entre los elementos que conforman las ICS, en la Figura 4 se muestra un diagrama de un típico sistema de control de infraestructura crítica con sus componentes principales,



Fuente: [33]

Figura 4. Típico sistema ICS.

Según el estándar de la NIST 800-82, la operación básica de las ICS está compuesta por tres elementos claves, a saber: el bucle de control (formado por los controladores, sensores y actuadores), la interfaz hombre-máquina (HMI) y las utilidades de diagnóstico remoto y mantenimiento. Para ilustrar mejor, en la Figura 5 se describe la interacción entre estos elementos.



Fuente: [34]

Figura 5. Proceso típico de un ICS.

Las 'cosas' en este ambiente son los controladores y sensores con capacidades de procesamiento y recolección de datos, cuya terminología depende del tipo del sistema al que estén asociados. Los más utilizados son las Unidades Terminales Remotas (RTU), los sensores inteligentes o actuadores conocidos como los IED y los controladores lógicos programables (PLC)[14]. Dentro del bucle de control, los sensores adquieren la data y transmiten a los controladores, quienes interpretan los datos según los parámetros definidos por el operario por medio del HMI y los manipulan para generar la información a enviar a los actuadores. Dentro de los actuadores, se pueden encontrar las válvulas de control, *breakers, switches* y motores.

[14] Dentro de un esquema SCADA, estos componentes son 'esclavos' del MTU.

Por otro lado, las RTU y los PLC se definen según la ICS-CERT [35], como las unidades de control computarizado, típicamente de rack o panel montado con tarjetas de procesamiento y de interfaz modulares. Las unidades están en el mismo emplazamiento del equipo de proceso y la interfaz, a través de módulos de entrada y de salida a los diversos sensores y dispositivos controlados. La mayoría utilizan una aplicación basada en la lógica programable que proporciona análisis y escritura de datos desde y hacia los módulos de interfaz IO, y se comunica con la red del sistema de control, por medio de diversos métodos de comunicación, generalmente por medio de redes LAN (excepto los SCADA que están diseñados para manejar comunicaciones a largas distancias).

Componentes

De acuerdo con el estándar NIST 800-82, los componentes claves de un ICS se clasifican en componentes de redes y componentes de control. En la Tabla 3 se enlistan algunos ejemplos de dichos componentes.

Tabla 3. Clasificación y ejemplo de componentes, según la NIST 800-82

Componentes de Control	Componentes de Redes
• Servidores: de control, SCADA o Master Terminal Unit (MTU) e Input/Output (IO) Server. • Controladores: RTU, PLC, IED. • *Data historian.* • HMI.	• *Remote access point*. • *Modems,* • *Firewalls,* • *Routers* de comunicación. • Redes *fieldbus.* • Redes de control.

Con respecto al tema de seguridad, existe la tendencia a utilizar interfaces abiertas y tecnologías basadas en servicios web, lo que puede

implicar una fácil puerta de ingreso a los ICS. Por ejemplo, una HMI puede ser una laptop conectada inalámbricamente, una plataforma dedicada o una página web desplegada desde un navegador; lo que puede generar puntos de acceso no autorizados a los sistemas si no se configuran adecuadamente. Además, el servidor IO puede ser utilizado como interfaz para controlar componentes de terceros, es decir, similar a un HMI.

De igual modo, el *Data Historian* es una base de datos centralizada de toda la información recolectada por el ICS, otro posible punto vulnerable por la falta de segmentación de la información.

En cuanto a las redes *fieldbus,* utilizan una variedad de protocolos para lograr la comunicación entre los sensores y otros dispositivos a un PLC u otro controlador, con la ventaja de que los mensajes enviados entre los sensores y el controlador, identifican a cada uno de los sensores. Con la intensión de lograr una mejor interoperabilidad, se establece el IEC 61158 *Fieldbus Standard*, el cual en realidad representa a varios estándares con requerimientos comunes, entre ellos Profibus, Foundation Fieldbus, WorldFIP, Modbus, y otros. Modbus es el más utilizado debido a su eficiencia, aunque, según los reportes de escaneo de puerto abiertos, es uno de los más expuestos en Internet.

Las redes de control, las cuales se encargan de conectar los niveles de supervisión con los niveles inferiores, tienen la difícil tarea de transportar y consolidar la información de varias fuentes. De modo que el protocolo de comunicación a utilizar, debe ser capaz de interactuar con múltiples marcas, sistemas y productos. El protocolo más utilizado es OPC, que es reconocido por su escalabilidad, versatilidad (es *open source*) e interoperabilidad con protocolos de capas inferiores y superiores, especialmente, si se utiliza OPC UA como protocolo integrador para la capa de negocio o final.

Servicios y protocolos

El principal error de los administradores de los ICS radica en aplicar la seguridad por la oscuridad; es decir, escudarse en que gracias al

desconocimiento de la dirección IP, el puerto y la marca del dispositivo, los sistemas utilizados están protegidos. Actualmente, existen varias herramientas, a disposición de todo el público, para escanear todo lo conectado a internet y obtener tanto los puertos abiertos, como información relativa al fabricante del equipo, entre otros datos.

Estos datos facilitan el trabajo de los atacantes, quienes suelen realizar un análisis de reconocimiento del objetivo para determinar los puntos débiles para perpetrar un ataque. Un paso típico en un ataque, es encontrar los puertos abiertos y verificar si es el puerto por defecto de un servicio u protocolo, para luego intentar una conexión remota y autenticarse (si el sistema posee este mecanismo) con las credenciales por defecto. En la Tabla 4 se presenta una breve descripción de los principales protocolos en uso y expuestos en Internet.

Tabla 4. Protocolos más utilizados en ICS

Protocolo	Área aplicación	Descripción
Tridium, Niagara Fox.	Automatización de edificios, datacenters, industrias, smart cities, gobierno.	Pertenece a Niagara Framework.
OMRON-FINS, Factory Interface Network Service.	Protocolo de red utilizado para redes físicas como Ethernet, Controller Link, DeviceNet and RS-232C.	Se utiliza para enviar mensajes entre controladores de la red del OMRON Framework.
EtherNetIP.	Automatización de procesos de manufactura.	Permite implementar IEEE 802.3 combinado con TCP/IP Suite.
DNP3 (Distributed Network Protocol).	Sistemas de automatización de procesos en *utilities* como compañías de agua y electricidad.	Set de protocolos de comunicación. Posee autenticación.
BACnet.	Automatización de edificios y redes de control.	Protocolo de comunicación.
Modbus.	Usado en la comunicación	Estructura de mensajería

	entre dispositivos y sensores inteligentes e instrumentos; para controlar los dispositivos de campo usando PC y HMI.	que establece la comunicación del tipo maestro-esclavo, sin autenticación, con checksum.

2.4. Análisis de similitudes y diferencias

Con base en la información recolectada, la mayoría de los ataques registrados se lograron debido a un acceso a la interfaz de control de los dispositivos, ya sea por medio de un servidor web o por un *software* propio del fabricante. No se necesita de ataques de fuerza bruta, pues los sistemas carecían de mecanismos de autenticación o tenía las credenciales predeterminadas. Es decir, el acceso a la consola de administración o configuración estaba libre de protecciones básicas, permitiendo el control total de los equipos al atacante.

De igual forma, los dispositivos o sistemas desactualizados y no parcheados permiten el uso de vulnerabilidades conocidas para ejecutar los ataques, mientras que la exposición de los puertos abiertos en Internet permite una peligrosa visibilidad de los dispositivos. Además, el uso de protocolos inseguros permite ataques del tipo *man-in-the-middle*, permitiendo otra vía para comprometer los equipos.

Adicionado a esta falta de medidas básicas de seguridad, algunos equipos presentan fallas de diseño, las cuales facultan al equipo o la consola de administración a realizar tareas o tener acceso a recursos innecesarios para sus funciones, pero que representan un problema de seguridad: acceso a la red corporativa, bases de datos, recursos compartidos de la compañía o realizar tareas innecesarias como posibilidad de subida de scripts, ejecución de código arbitrario, configuración de un segundo servidor DNS, entre otros.

Por ejemplo, con los detalles de los CVE referentes a los principales equipos y protocolos expuestos (ver anexo 1) es posible determinar que los

servidores web asociados a los IIoT suelen poseer medidas inadecuadas para contrarrestar ataques conocidos como XSS, CSRF, SQL injection...datos que se confirmaron en la sección 2.2. Así, la cantidad de dispositivos vulnerables a ataques de desbordamiento de búfer era de esperarse debido a las limitantes de procesamiento y configuración de los equipos ICS.

Otro factor predecible lo constituyen los ataques de denegación de servicio a estos equipos para afectar su disponibilidad, o bien, el uso de los dispositivos como zombies para la creación de una *botnet* gigantesca para realizar un DDoS a otro objetivo.

2.5 Definición de factores de seguridad y métodos de verificación

Con la información recolectada de las secciones anteriores, se determinan los principales factores de seguridad y los métodos de verificación para cada caso. Se asume que previamente se obtuvieron los datos básicos del *IIoT;* fabricantes, modelos, versiones, CPU, *clock*, ranuras de expansión, puertos específicos de cada fabricante, áreas de memoria, usuarios, *passwords*, directorios ocultos.

Tabla 5. Factores de seguridad y métodos de verificación

Factor de seguridad	Método de verificación	Recurso necesario
Exposición del dispositivo.	Validar visibilidad y disponibilidad de datos del dispositivo en Internet.	Shodan.
Credenciales por defecto.	Intento de inicio de sesión con credenciales por defecto disponibles.	Lista de credenciales por defecto por IIoT [37].
Comunicaciones no cifradas.	Analizar tráfico entre elementos durante transmisión de datos.	Security Onion.
Protocolos vulnerables.	Validar si los protocolos en uso de todos los elementos (tanto que conforman como asociados) estén actualizados (última versión o parcheado) o no tengan vulnerabilidades conocidas.	CVE database, Shodan.

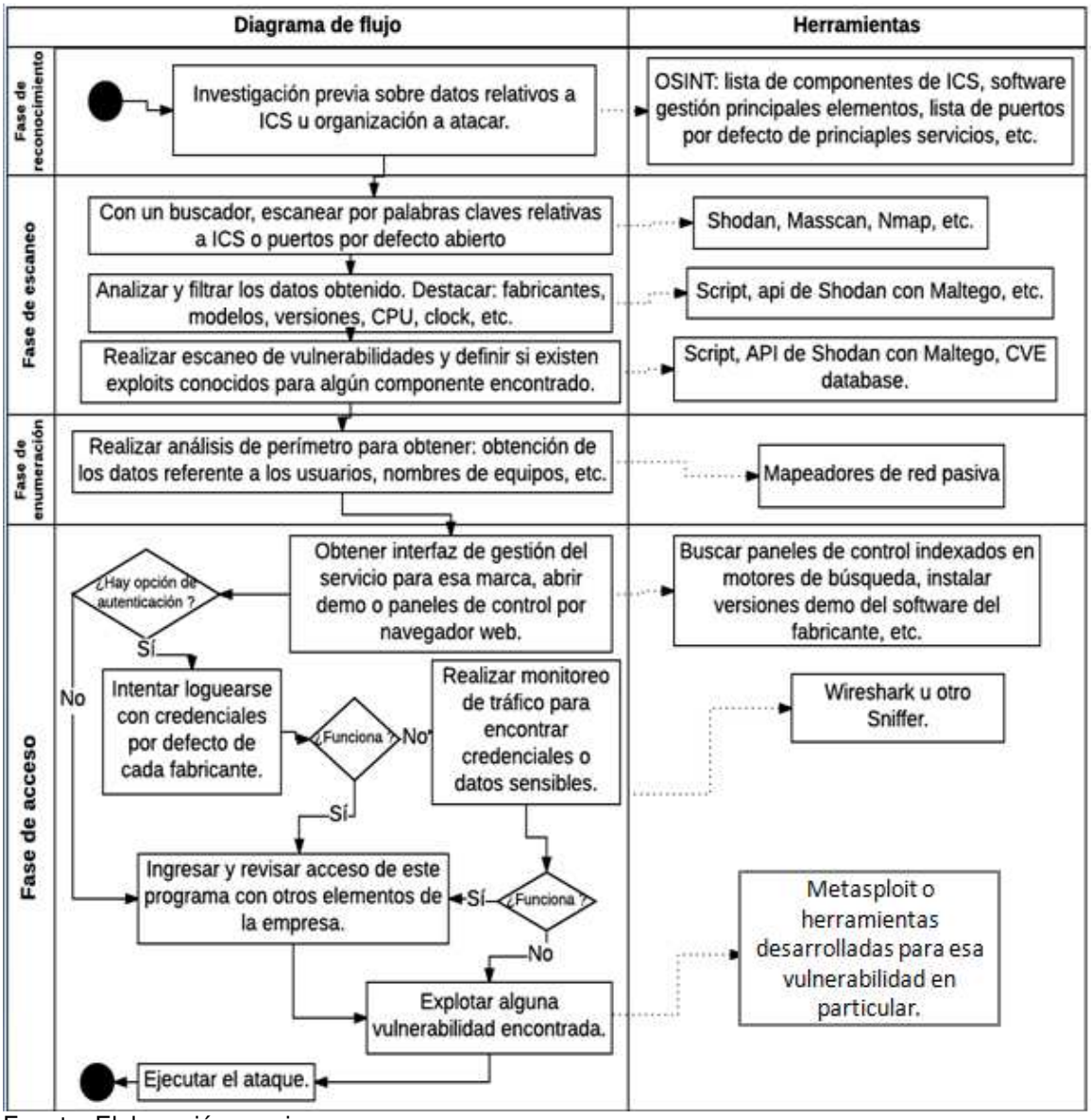
Nivel de seguridad del portal de gestión del IIoT,	Revisar nivel de seguridad de la interfaz: autenticación, autorización, servicios disponibles.	-Google dorking. -Lista *software* de gestión y fabricante. -Escáner de vulnerabilidades.
Transmisión de data.	Validar si: -Se está enviando información a una fuente desconocida o no válida (posible servidor C&C). -Data transmitida es la justa y necesaria.	Script que valida puertos abiertos vs puertos requeridos. Comparación con la lista de puertos más utilizados para ataques DDoS.

Se limita los factores de seguridad a verificar a la lista definida en la Tabla 5. En el siguiente capítulo, se muestra un diseño de aplicación y servicio asociado en la nube para el procesamiento de datos, con el fin de obtener un panorama del comportamiento del dispositivo y definir si cumple o incumple algún factor de seguridad citado en la tabla anterior.

Capítulo 3: Requerimientos y diseño lógico

3.1. Análisis de factores de seguridad y definición de medios de solución.

En la Figura 6 se presenta un diagrama de flujo donde se generaliza el procedimiento típico de un ataque a un ICS, a fin de analizar los factores de seguridad a tomar en cuenta.



Fuente: Elaboración propia.

Figura 6. Diagrama de un ataque típico a un ICS y las herramientas utilizadas.

De acuerdo con la Figura 6, el primer paso es de reconocimiento y recolección de datos sobre el objetivo (procedimiento típico en pruebas de penetración). Esta etapa se realiza con herramientas y técnicas OSINT, pero se añade como información adicional en el reporte final al cliente en la sección de nivel de exposición de los equipos o empresa.

Después, se aplica el escaneo de la red, con el propósito de seleccionar el equipo potencialmente vulnerable. Normalmente se utilizan buscadores de banner de servicio o *banner grabbing* como Shodan; de esta manera, se obtiene data del equipo y se determina el nivel de visibilidad/exposición del dispositivo en cuestión en Internet. En algunos casos estudiados, los especialistas en seguridad han creado scripts para facilitar el resumen de resultados, es decir, para filtrar los datos que consideran críticos de la información recolectada o utilizan funciones complementarias como Maltego o FOCA. Asimismo, en esta etapa se analizan los servicios y versiones utilizados por el objetivo para validar si están actualizados o son vulnerables.

La siguiente etapa es enumeración, en esta se determinan los datos del equipo y el entorno que lo rodea. Existen varias opciones de mapeadores de red pasiva que realizan esta tarea de manera automatizada.

Finalmente, en la etapa de acceso, se intenta llegar al objetivo utilizando la información recolectada. En anteriores secciones de este documento se detalla que, para los equipos ICS, la mayoría de los accesos remotos no autorizados se deben al uso de credenciales por defecto en interfaces de gestión de los equipos. Por ende, se toma esta maniobra como primer paso de intento de acceso. En caso de que falle, se analiza el tráfico de las comunicaciones entre los objetivos y otros equipos del entorno para validar si la transmisión es segura (cifrada y con condiciones mínimas de seguridad). Si no funciona, se procede a explotar alguna vulnerabilidad conocida de algún servicio que utilice el dispositivo o algún elemento del ambiente que interactúa con el equipo.

Herramientas útiles

Existen varias herramientas de acceso público que pueden facilitar algunas tareas de monitoreo y control de aspectos de seguridad, descritos anteriormente. En la propuesta, se incluyen estas herramientas como parte del servicio alojado en la nube. Por ejemplo, el distro de Linux llamado Security Onion [39] que reúne varias herramientas de monitoreo y mapeo de redes. Así, se utilizan los buscadores Google y Bing para determinar los portales de gestión de los equipos a verificar y el escáner de vulnerabilidades de páginas web VEGA, en caso de que el HMI sea una página web.

Se propone el uso de estas herramientas desde un servidor en la nube, con el fin de generar un reporte completo sin utilizar recursos de la aplicación móvil.

3.2 Propuesta del diseño lógico: documento de especificación del *software*

Para presentar el diseño de la propuesta, se utiliza como base el estándar de la IEEE 830 con el propósito de definir un documento de especificación de software. Se recalca que la propuesta se basa en este estándar, pero no contempla todos los puntos, pues algunos son redundantes con otras secciones del documento de investigación.

Sección 1-Introducción

Propósito

Este documento es parte de una investigación para determinar implicaciones de seguridad en dispositivos de Internet de las Cosas y buscar una solución para prevenir al usuario no especializado.

Con respecto a lo mencionado en la introducción de este documento, el propósito es crear el documento de especificación de requerimiento de *software* para su posterior desarrollo e implementación, por parte de una empresa o programador.

Criterios de éxito

- Crear un rol de Supervisor de ICS para el ingreso de parámetros de búsqueda de IIoT y selección del IIoT a verificar.
- Análisis de datos recolectados por la aplicación y procesamiento en un servidor en la nube, para determinar el nivel de cumplimiento de los factores de seguridad: uso de credenciales por defecto, comunicaciones no cifradas, transmisión de data, exposición IIoT en Internet y uso de protocolos/servicios vulnerables en IIoT o en el portal de gestión del IIoT.
- Generación de un reporte con el estado de los factores de seguridad definidos en el IIoT seleccionado por el Supervisor de ICS.

Alcance

El *software* a diseñar es una aplicación móvil y su servicio en la nube para procesamiento de datos. La aplicación móvil funcionará como un panel de control para definir los parámetros a validar (escoger el objeto de estudios o *IIoT*, definir localización y formato del reporte a generar, entre otros).

Es necesario recalcar que esta propuesta de aplicación no viene a suplantar las funciones de un anti malware, es decir, solo verificará los factores de seguridad definidos en la investigación (ver Capítulo 2, sección 2.5). Debido a que se selecciona como área de estudio las infraestructuras críticas, se determina que los usuarios serán los ingenieros o técnicos encargados de manejar los ICS de estas compañías.

El servicio en la nube será manejado por un administrador no local, es decir, que no pertenece al personal de la empresa que desea auditar el IIoT. El servicio de la nube se pretende que sea gestionado por la empresa que desarrolle la aplicación móvil como un servicio tercerizado de gestión de datos. Dicha asistencia deberá incluir la gestión, mantenimiento y actualización de los recursos utilizados para el análisis de datos contenidos en el servidor de la nube.

El principal beneficio del uso de esta aplicación por parte de estos usuarios es que les facilitará el diagnóstico de problemas típicos de seguridad

para alertar a las entidades responsables, sin la necesidad de contar con conocimientos profundos sobre el tema.

Definiciones, acrónimos y abreviaciones

Todas abreviaciones y acrónimos están enlistados en la sección 'Nómina de abreviaturas'. Las definiciones están descritas en la sección 'Glosario'. Ambas secciones son parte del documento de investigación del cual esta sección forma parte.

Referencias

Para la realización de este documento, se utiliza como base en la síntesis del documento IEEE-STD-830-1998 realizado por profesor Gregor Bochmann de la Universidad de Ottawa [40] y la guía de implementación del profesor Jaime Solano del Instituto Tecnológico de Costa Rica. Otras referencias utilizadas están enlistadas en la sección Bibliografía Específica y General.

Apreciación Global

Este documento de especificación de requerimientos de software está organizado en tres secciones: esta primera sección define la propuesta, el enfoque y los usuarios a los que está dirigido. En la sección 2 y la sección 3 se profundiza en el diseño de la solución propuesta, por medio de los diagramas requeridos según la especificación IEEE 830.

Sección 2- Descripción General

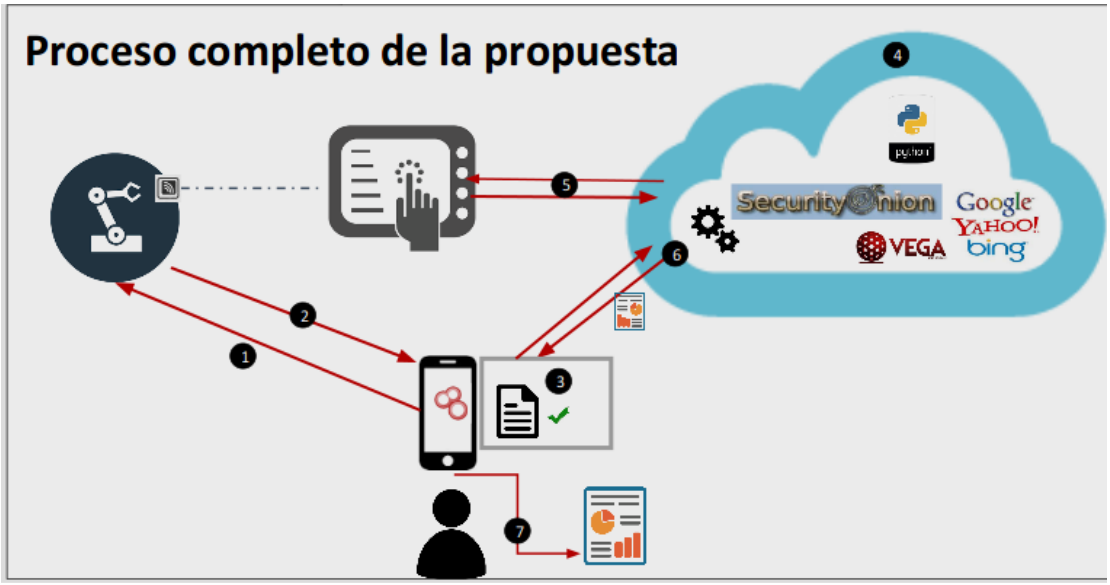
Los factores generales que afectan el producto descrito en esta propuesta son los factores de seguridad que se validan en el dispositivo IIoT. En el capítulo 2 se indica con detalle la justificación de la elección de estos factores.

Perspectivas del producto

A nivel general, la propuesta está conformada por dos partes: la aplicación móvil para configuración de parámetros mediante una interfaz sencilla, donde el usuario ingresa datos para realizar una búsqueda en Shodan. Una vez realizada la búsqueda, se despliega la información para que el usuario seleccione el equipo a verificar.

La otra parte, es el servicio en la nube para procesamiento de datos, con conexión a los siguientes servicios: recolección de información sobre la empresa o equipos con el uso de Google y Bing, escáner de vulnerabilidades de páginas web con la aplicación VEGA y el uso de varias herramientas reunidas en el distro de linux Security Onion. Además, dentro de los servicios de la nube, se contempla la ejecución de scripts en Python para la extracción y filtración de data, para el informe final al usuario.

Las interfaces externas de la propuesta se demuestran en los siguientes diagramas, donde se describen los sistemas, usuarios, hardware, software y medios de comunicación a utilizar a nivel conceptual, sin definir exactamente proveedor o fabricante, pues estos detalles se especifican con el arquitecto y el desarrollador en la etapa de implementación de la propuesta.



Fuente: Elaboración propia.

Figura 7. Diagrama de bloques del proceso completo de la propuesta

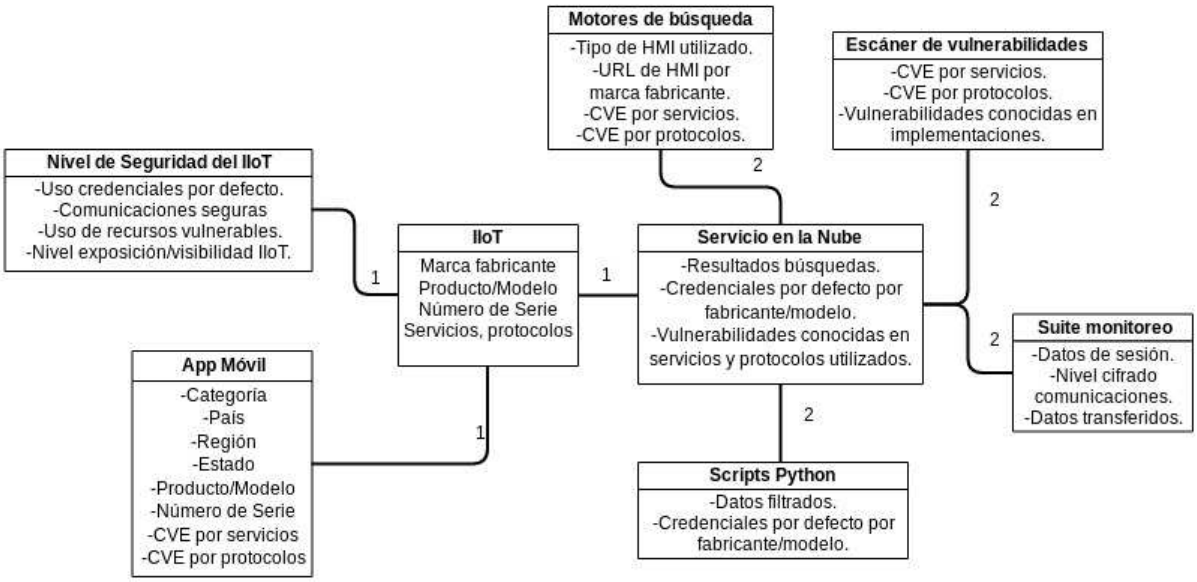
A partir del diagrama de la Figura 7, se determina el flujo esperado del proceso de la propuesta. A continuación se detallan los principales pasos del mismo, los cuales fueron representados por círculos negros con su respectivo número en el diagrama:

1. Se inicializa SHODAN y se busca, según parámetros ingresados por usuario.
2. Se encuentran dispositivos de acuerdo a la búsqueda y se despliega en la App un resumen de data para su confirmación por el usuario.
3. El usuario selecciona el IIoT que desea chequear y se envía la data obtenida al servidor en la nube.
4. Se verifican los factores de seguridad con las herramientas disponibles.
5. Se intenta iniciar la sesión con credenciales por defecto en HMI.
6. Se genera el reporte final con todos los datos procesados.
7. Se despliega el reporte final en la pantalla de la aplicación.

Funciones del producto

Las principales capacidades funcionales del producto son la recolección de datos y el *banner grabbing* de la aplicación móvil, así como los procesamientos de estos datos por parte de los distintos recursos disponibles en el servidor en la nube para la generación de un reporte final al usuario.

La Figura 8 muestra el modelo del dominio del sistema y en la Tabla 8 se detallan los elementos que lo conforman.



Fuente: Elaboración propia.

Figura 8. Modelo de dominio del sistema para la propuesta.

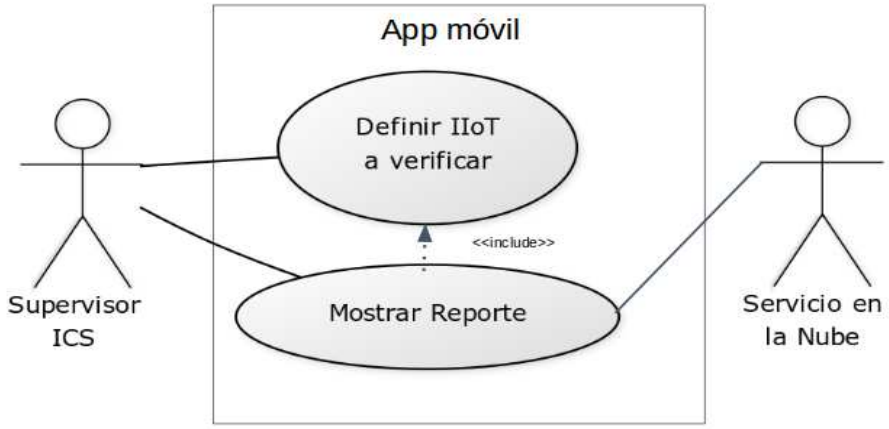
En la Tabla 6 se detallan los elementos presentados en la Figura 8.

Tabla 6. Descripción del modelo de dominio

Concepto	Intención	Extensión
IIoT	Existe un dispositivo IIoT del cual los demás conceptos del sistema usan los datos para definir su nivel de seguridad.	Se limita a equipos IIoT de infraestructuras críticas o ICS.
App Móvil	Ingreso de datos para definir parámetros de búsqueda del IIoT con el fin de determinar los servicios, protocolos y nivel de visibilidad del dispositivo.	Puede ser una aplicación Android o iOS que utilice la API de Shodan.
Servicios en la nube	Analiza datos obtenidos por la App para verificar el nivel de seguridad del dispositivo.	Servidor en la nube con varios recursos.
Scripts	Automatiza tareas como emparejar tipo de IIoT con credenciales por defecto y filtrar datos para el informe.	Scripts desarrollados en Python.

Motores de búsqueda	Busca información relativa a los tipos de interfaz de gestión del IIoT y vulnerabilidades conocidas de estos.	Motores de búsqueda como Google y Bing.
Suite Monitoreo	Determina el nivel de cifrado de la transmisión de datos y que los datos enviados sean los necesarios.	La distro de linux Security Onion.
Reporte del nivel de seguridad	Informe con datos recolectados y procesados por recursos de la aplicación móvil y los servicios de la nube.	Reporte en pantalla con posibilidad de exportar a formato pdf o csv.

Diagrama de Casos de Uso



Fuente: Elaboración propia.

Figura 9. Diagrama de Casos de Uso.

Restricciones de la propuesta

Las funciones de la aplicación móvil se restringen a la obtención de los parámetros para la búsqueda en Shodan, así como la presentación del informe al usuario. Se considera que todos los comandos y acciones que requieren más procesamiento se realizarán con las herramientas localizadas en el servidor de la nube asociado con esta propuesta. Como se indica previamente en este documento, el diseño de esta propuesta se enfoca en los dispositivos IIoT, pero valida solo algunos factores de seguridad, los cuales se aclaran en el capítulo 2.

Esto implica que la propuesta no abarca absolutamente todos los factores de seguridad, ni todos los tipos o variedades de IIoT del mercado.

Características del Usuario

El usuario de esta propuesta es el personal encargado del dispositivo IIoT, quien, se asume, posee conocimientos reducidos en el tema de seguridad informática. La finalidad es que sea utilizado por personas no especializadas en el tema de seguridad y que no sean necesariamente miembros del Departamento de TI o Seguridad Informática de la empresa, pero que puedan verificar los aspectos básicos de seguridad del dispositivo.

Supuestos y dependencias

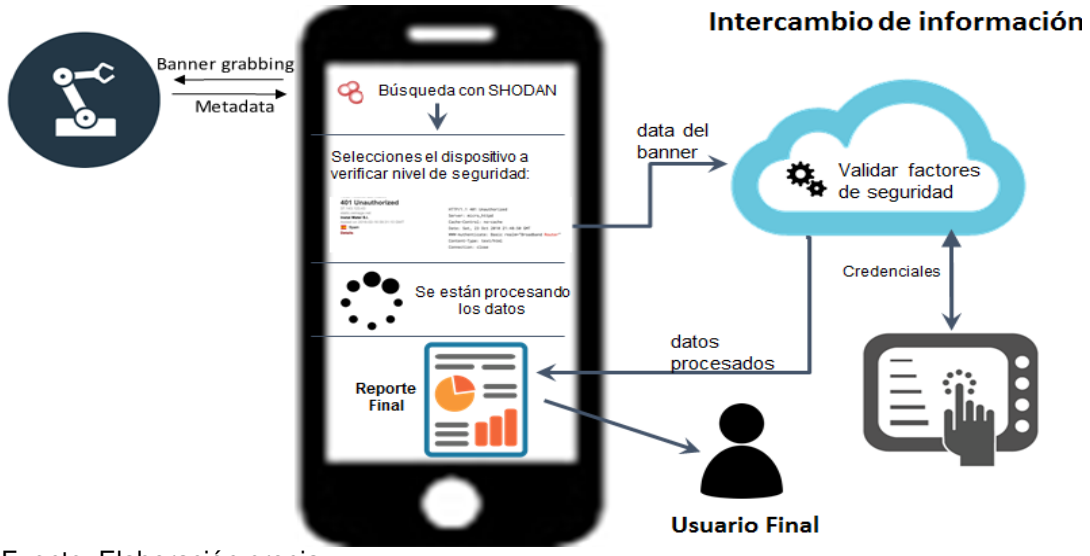
El usuario debe poseer la API key, la cual se genera al obtener una cuenta en Shodan, para tener acceso a todos los filtros de búsqueda y la API de Shodan.

Se supone que el usuario conoce la serie del equipo por verificar, ya sea por detección con herramientas (como Fing que enlista todos los equipos conectados a una red Wi Fi), o por obtención manual del dato. Dado esto, se recomienda que la aplicación se encuentre en la misma red local donde se encuentra el dispositivo IIoT.

Sección 3- Requerimientos específicos

Interfaces del sistema

Para explicar el comportamiento de la App con los otros elementos, se detalla el intercambio de información en la Figura 8.



Fuente: Elaboración propia.

Figura 10. Intercambio de información entre la App y otros elementos de la propuesta.

Desde la aplicación móvil, el usuario selecciona una serie de parámetros que se traducen en el comando de búsqueda en Shodan. Cabe resaltar, que es necesario que el usuario tenga una cuenta en shodan.io para utilizar todos los filtros disponibles. Los resultados de la búsqueda se muestran en la pantalla para que el usuario seleccione la opción que corresponde al dispositivo a verificar. Si el usuario ingresa el número de serie del dispositivo, solo el resultado que concuerde con este dato se mostrará en pantalla, en caso contrario, se mostrarán todos aquellos que concuerden con los parámetros de búsqueda.

Una vez elegido el dispositivo, se envían los datos al servicio en la nube para la búsqueda del tipo de interfaz de gestión que utiliza ese IIoT y analice el nivel de seguridad de ese elemento, así como buscar las credenciales por defecto asociadas a ese dispositivo.

En las Figuras 11 y 12 se muestran las pantallas propuestas para la aplicación móvil, mientras que en la Figura 13 se muestran los mensajes de error a desplegar.

Fuente: elaboración propia.

Figura 11. Propuesta de las pantallas de la aplicación móvil (1 de 3).

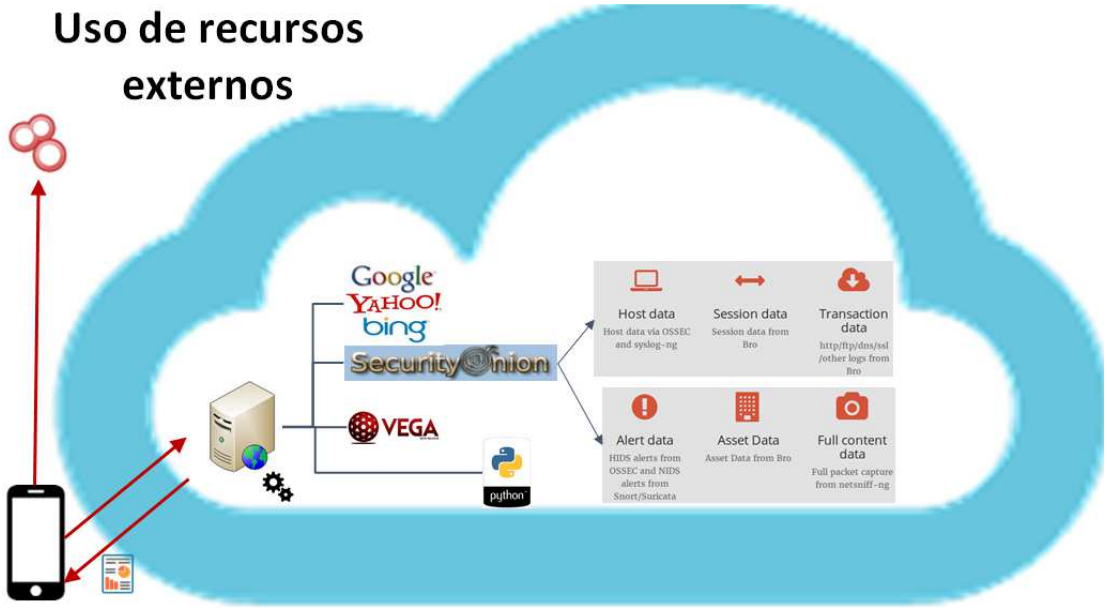
Fuente: Elaboración propia.

Figura 12. Propuesta de las pantallas de la aplicación móvil (2 de 3).

Fuente: Elaboración propia.

Figura 13. Pantallas de error para diferentes casos y opciones del menú.

Por otro lado, en la Figura 14 se diagrama la interacción entre la aplicación móvil y el servidor en la nube, con los respectivos recursos que este contiene.



Fuente: Elaboración propia.

Figura 14. Uso de recursos externos contemplados en la propuesta.

Los recursos externos detallados en la Figura 14, muestran ejemplos de herramientas por utilizar para validar los factores de seguridad de la propuesta. En la tabla 7 se describe la función de cada herramienta con respecto al factor de seguridad.

Tabla 7. Descripción de los recursos de terceros y los factores de seguridad

Plataforma	**APP**	**Servicios en la nube**			
Aplicación	*Shodan*	*Google, Bing, etc.*	*Scripts en Python*	*Security Onion*	*VEGA*
Input	-Tipo de dispositivo. -Región/Estado/País -Número de serie -Producto	Metadata (información Banner de Servicios).	-Modelo y fabricante. -Puerto Abiertos.	-Intercambio de claves de cifrado.	URL del portal de gestión
Factor de Seguridad	-Exposición/ visibilidad a Internet. -Versiones, servicios, etc. desactualizados o vulnerables.		-Credenciales por defecto. -Puertos abiertos innecesarios.	-Transmisión de data. -Comunicaciones no cifradas.	Nivel seguridad portal de gestión del IIoT.
Data a mostrar en el reporte (output)	Metadata (información Banner de Servicios). Versiones, servicios, etc vulnerables o desactualizados.	-URL del portal de gestión o nombre del software propietario.	-Uso credenciales por defecto. -Puertos abiertos vs puertos requeridos. Comparación con lista de puertos más utilizados para ataques DDoS.	-Nivel de cifrado en las comunicaciones -Tráfico inusual o envío innecesario datos.	Versiones, servicios, etc. vulnerables o desactualizados en portal de gestión

A continuación, se detalla cada recurso externo por utilizar en la propuesta para justificar su elección.

Shodan: se pretende utilizar la REST API de Shodan, la cual proporciona métodos para búsquedas en Shodan: buscar nodos, obtener información resumida sobre las consultas y una variedad de métodos de utilidad, para hacer más fácil el desarrollo e integración con la aplicación móvil de la propuesta. Además, se aplicará la API *Exploits* de Shodan, el cual proporciona acceso a varias fuentes de datos de vulnerabilidades a explotar. Por el momento, esta opción busca a través de los siguientes recursos: *Exploit DB, Metasploit y Common Vulnerabilities and Exposures* (CVE).

Google/**Bing**: se utilizan estos motores de búsqueda para encontrar detalles de los portales de gestión y vulnerabilidades asociadas a los dispositivos IIoT a verificar. Se aplica la técnica conocida como Google Dorking donde se ingresan comandos especiales en el motor de búsqueda para encontrar información indexada.

Security Onion: distro de linux con una suite de herramientas reunidas en el Network Security Monitor (NSM) dedicado al control de eventos de seguridad en una red. Se puede utilizar de manera proactiva (para identificar las vulnerabilidades o los certificados SSL que expiran) o reactiva (respuesta a incidentes y análisis forense de red).

VEGA: escáner de vulnerabilidades en páginas web. Se utiliza para validar si el portal de gestión del dispositivo IIoT posee servicios, protocolos, etcétera, con vulnerabilidades conocidas o desactualizados.

Scripts de Python: los *scripts* automatizan la recolección y filtración de datos de los distintos recursos para la elaboración de informes. También se utilizan para:

- Buscar las credenciales por defecto de la lista del archivo .csv de la página de Scada Strangelove con los datos del dispositivo IIoT por verificar.
- Comparar los puertos abiertos contra los requeridos por el fabricantes y los más utilizados en ataque DDoS (archivo port.cvs).
- Para la traducción de los datos enviados por la Shodan API para realizar el comando del Google Dorking (inurl: XXXXX)
- Realizar un intento de inicio de sesión y captura de resultados (éxito o fracaso).
-Enviar datos a VEGA (URL a escanear) y datos de transmisión de data a Security Onion.

Resulta imperioso recalcar que, dentro de la estructura de la propuesta, todas las comunicaciones entre los elementos deben de contar con comunicaciones cifradas end-to-end con métodos criptográficos robustos,

mecanismos de autenticación adecuados: solicitud de credenciales, políticas contra ataques de fuerza bruta, doble factor, entre otros.

Sumado a lo anterior, se recomienda seguir el *checklist* de OWASP para garantizar los puntos básicos de seguridad en los sistemas, tanto del servicio en la nube como en la aplicación móvil.

Funciones

A continuación se presenta una descripción detallada de los Casos de Uso indicados en la sección 2.

Caso de prueba 1: Definición de IIoT a verificar

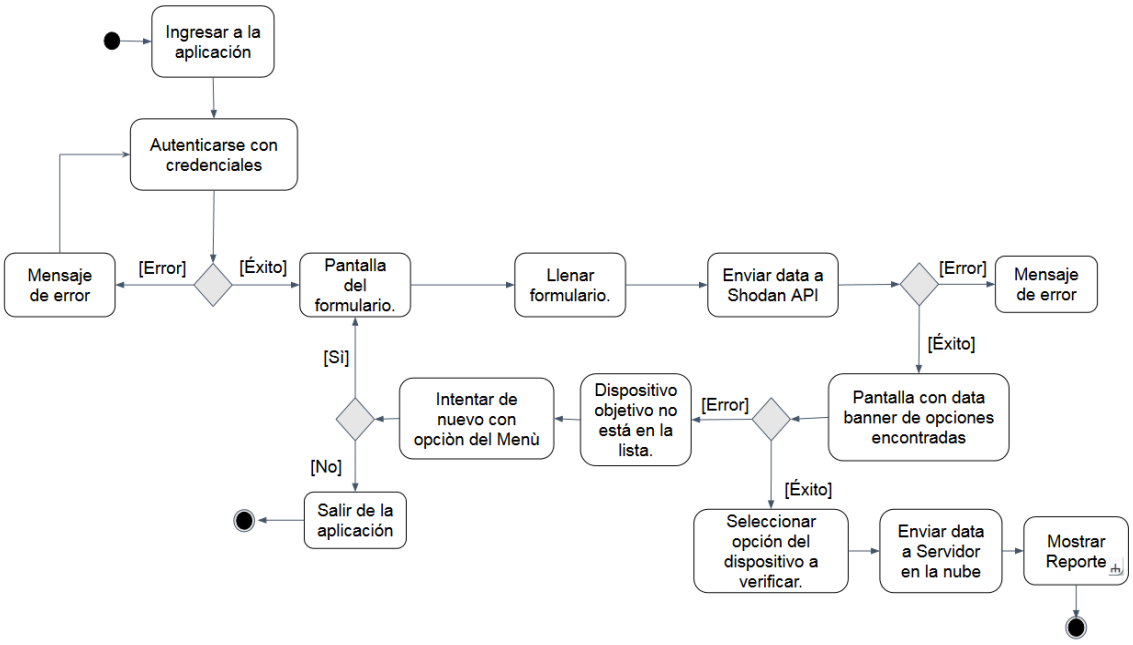
- **Texto**

Tabla 8. Texto del caso de uso 1

Actores	Supervisor ICS
Descripción	Ingresar los parámetros para la búsqueda en Shodan del IIoT y selección del mismo en caso de que se desplieguen varios resultados.
Inicio	Cuando se abre la aplicación.
Terminación	Cuando se llega a la pantalla de espera de los datos procesados.
Precondiciones	-Usuario con credenciales válidas. -Conocer el número de serie del IIoT o modelo/ fabricante para poder identificarlo entre los resultados de la búsqueda. -Contar con acceso a Internet. -Contar con API key de Shodan y previamente haberla ingresado a la aplicación.
Pos condiciones	-Información del banner de servicios de los puertos abiertos del IIoT. -Vulnerabilidades de los servicios/protocolos utilizados por el IIoT.

Flujo Normal	1. Ingresar a la aplicación. 2. Iniciar sesión. 3. Ingresar datos del formulario. 4. Presionar botón con ícono de 'siguiente' 5. Esperar presentación de resultados. 6. Seleccionar IIoT que se desea verificar.
Flujo Alternativo	6. No se encuentran IIoT con esa descripción y se muestra un mensaje de información.
Excepciones	-Si se ingresan credenciales incorrectas, se muestra un mensaje de error. -Si no se ingresan los datos en los espacios obligatorios, se muestra un mensaje de error. -Si se ingresan datos no alfanuméricos, se muestra un mensaje de error.

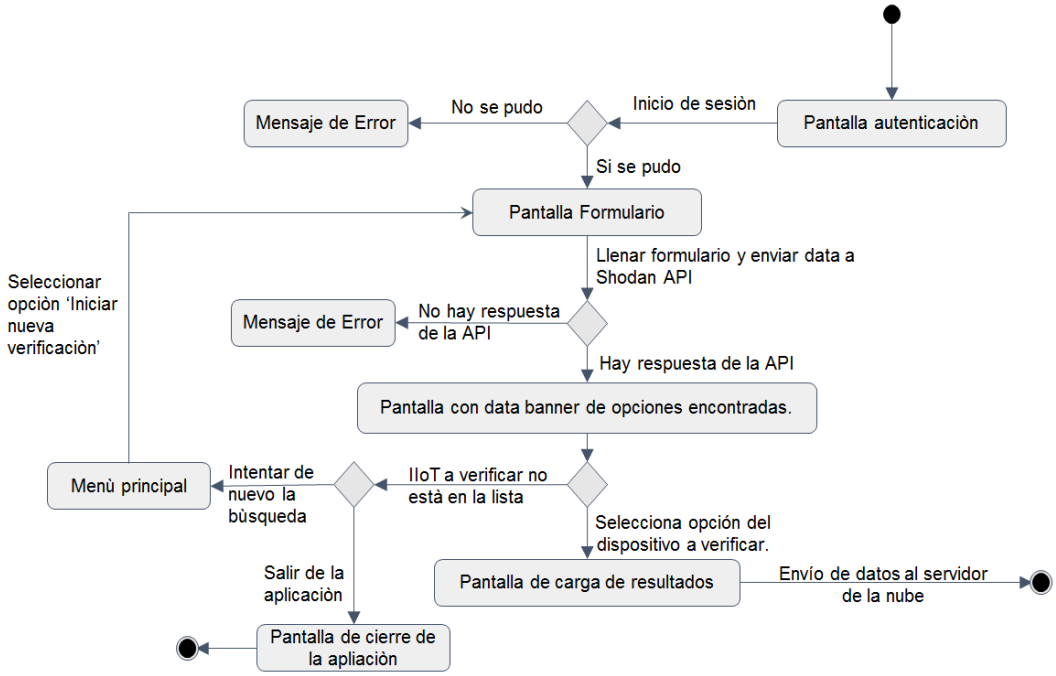
- **Diagrama de actividades**



Fuente: Elaboración propia

Figura 15. Diagrama de actividades del caso de uso 1.

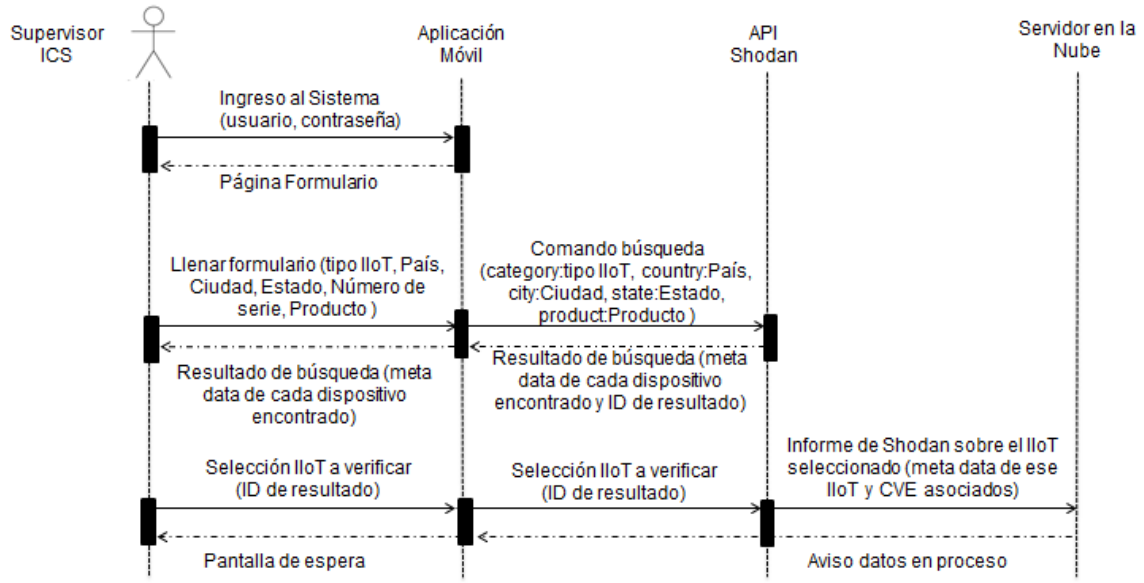
- **Diagrama de Estado**



Fuente: Elaboración propia.

Figura 16. Diagrama de estado del caso de uso 1.

- **Diagrama de secuencia del sistema**



Fuente: Elaboración propia.

Figura 17. Diagrama de secuencia del caso de uso 1.

- **Casos de prueba**

Tabla 9. Casos de Prueba para caso de uso 1

#	Entrada	Salida
1	Ingresar todos los parámetros obligatorios.	Mostrar en pantalla los resultados de la búsqueda.
2	Ingresar todos los parámetros obligatorios.	Mostrar mensaje de información en caso de encontrar dispositivos.
3	Seleccionar una opción.	Mostrar pantalla de espera de resultados.
4	Seleccionar una opción.	Mostrar en pantalla el reporte final con todas las secciones: información general, alertas, vulnerabilidades, recomendaciones y detalle de factores de seguridad.

Caso de Uso 2: Ver reporte

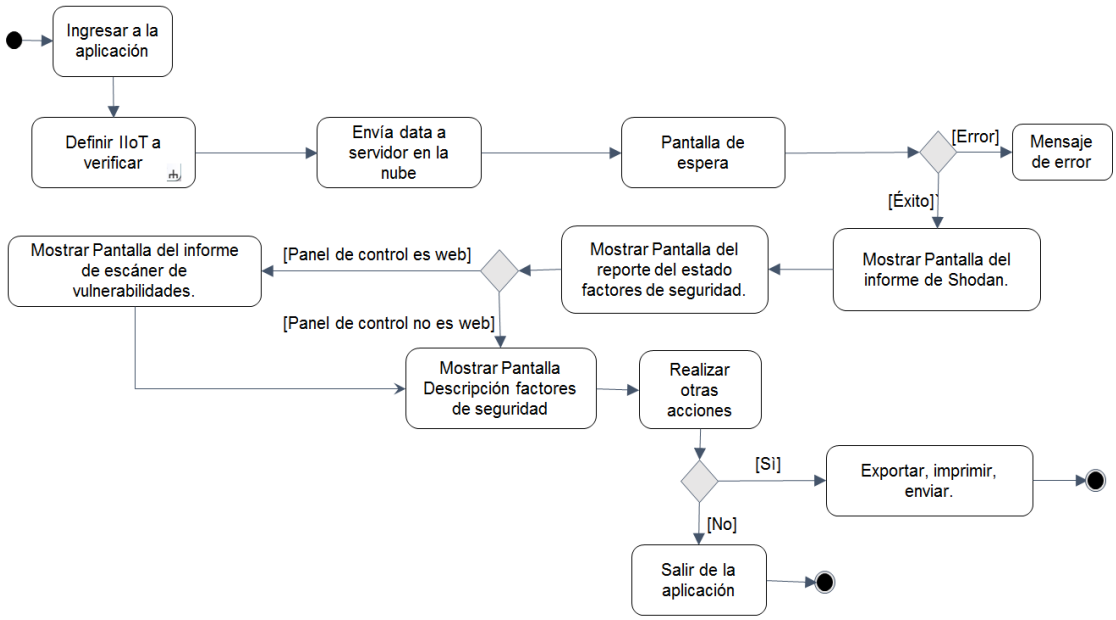
- **Texto**

T**abla 10. Texto de caso de uso 2**

Actores	Supervisor ICS
Descripción	Desplegar el informe sobre el nivel de seguridad del dispositivo IIoT obtenido del procesamiento de los datos en el servicio en la nube.
Inicio	Envío de datos al servidor en la nube.
Terminación	App recibe reporte y despliega en pantalla los datos.
Precondiciones	-Conexión a Internet. -Conexión con el servicio en la nube. -Envío de parámetros, resultados de búsqueda y selección de IIoT al servicio en la nube. -Funcionamiento correcto de todas las herramientas disponibles en la nube.
Pos condiciones	Reporte con nivel de cumplimiento de los factores de seguridad definidos a evaluar, lista de vulnerabilidades, recomendaciones y detalle de los factores de seguridad.

Flujo Normal	1. Ingresar a la aplicación. 2. Ingresar datos del formulario. 3. Esperar presentación de resultados. 4. Seleccionar IIoT que se desea verificar. 5. Esperar por presentación de datos en pantalla.
Flujo Alternativo	NA
Excepciones	-En caso de fallo en envío de datos desde la App al servidor en la nube, se muestra un mensaje de error. -En caso de fallo de procesamiento de datos en el servidor de la nube, se muestra un mensaje de error.

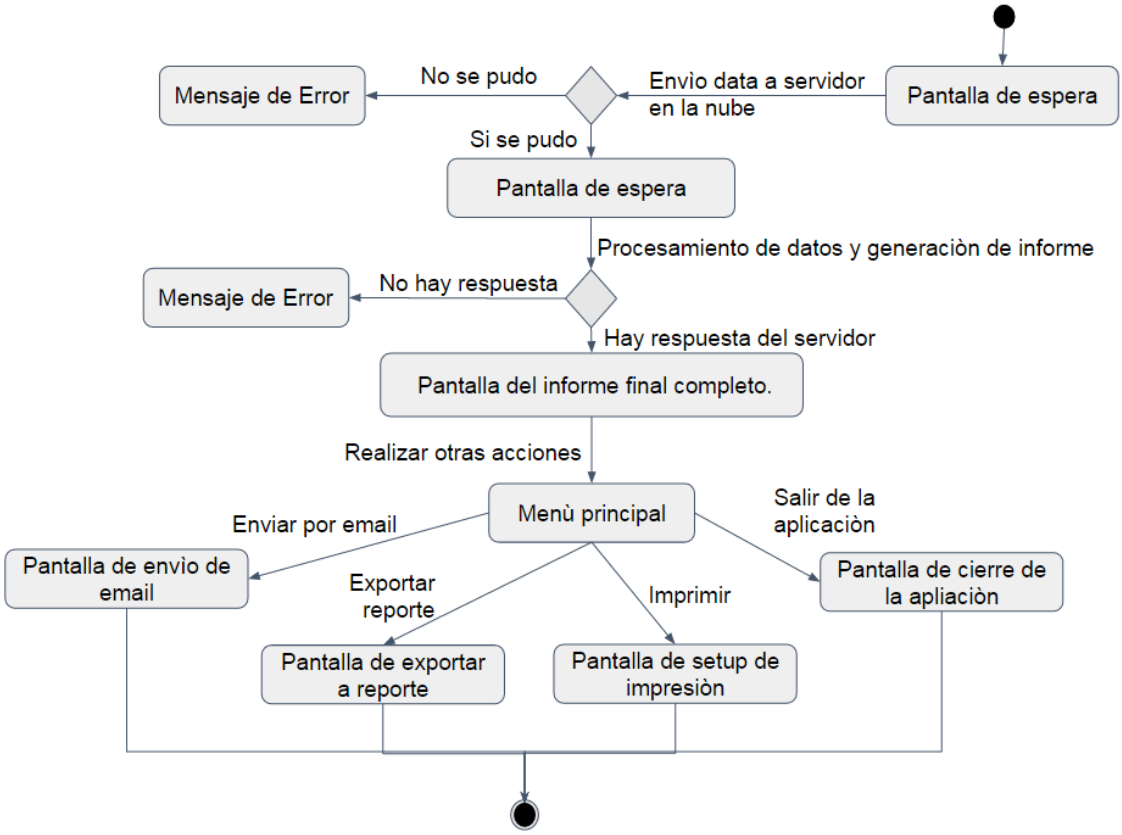
- **Diagrama de actividades**



Fuente: Elaboración propia.

Figura 18. Diagrama de actividades del caso de uso.

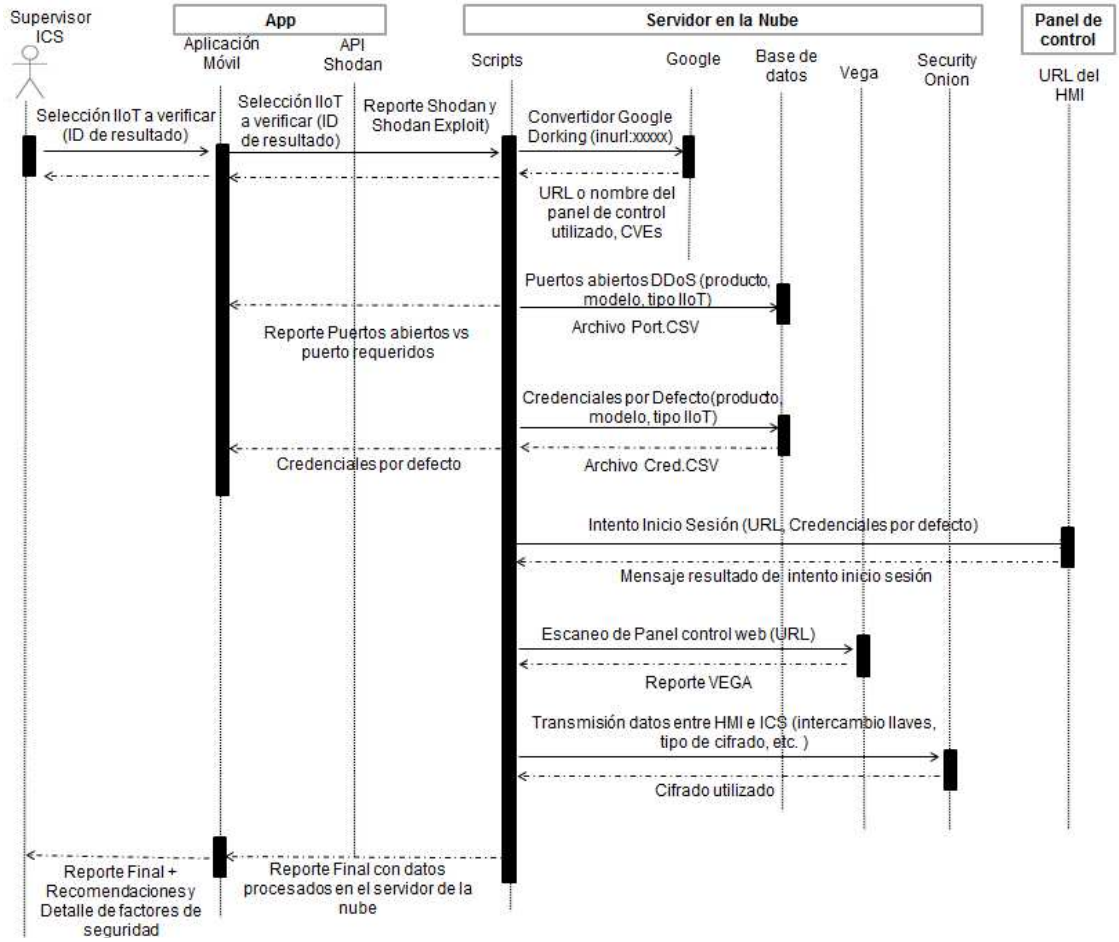
- **Diagrama de Estado**



Fuente: Elaboración propia.

Figura 19. Diagrama de estado del caso de uso 2.

- **Diagrama de secuencia del sistema**



Fuente: Elaboración propia.

Figura 20. Diagrama de secuencia del caso de uso 2.

- **Casos de prueba**

#	Entrada	Salida
1	Ingresar todos los parámetros obligatorios y seleccionar dispositivo IIoT a verificar.	Mostrar en pantalla el reporte final.
2	Ingresar todos los parámetros obligatorios y seleccionar dispositivo IIoT a verificar.	Mostrar mensaje de error en caso de problemas de envío de datos o de su procesamiento.

Restricciones de diseño

En otros apartados de este documento se detalla la carencia de un estándar de uso generalizado por parte de los fabricantes de dispositivos IoT. Sin embargo, los factores de seguridad escogidos se basan en recomendaciones y guías de buenas prácticas para el desarrollo de dispositivos IoT, aplicaciones móviles y sistemas informático. Principalmente, se recomienda seguir las indicaciones de desarrollo seguro de OWASP e incluir procedimientos de Auditoría, como registro de actividades para el posterior análisis en caso de presentarse algún problema.

Capítulo 4: Análisis, recomendaciones y conclusiones.

4.1. Viabilidad del proyecto, recomendaciones y cumplimiento de objetivos

Se considera que el proyecto es viable, pues hace uso de recursos especializados ya desarrollados y cuya eficacia es conocida, por lo que las tareas a realizar son de integración de esta con un *endpoint* para recolectar la información inicial y envío de esta al servidor en la nube.

A nivel económico, se deberá crear un análisis de costo-beneficio en caso de que la idea le interese a un desarrollador; no obstante, a nivel cualitativo, los beneficios a la seguridad son evidentes.

Por otro lado, se recomienda aplicar mayores medidas de seguridad en compañías que adopten el uso de dispositivos IoT; es decir, como mínimo aplicar las siguientes contramedidas:

- Analizar, diseñar, elaborar e implementar un marco normativo de ciberseguridad específico para la naturaleza de su industria.
- Detectar brechas de ciberseguridad respecto a las buenas prácticas internacionales.
- Identificar y clasificar los ciberactivos.
- Identificar y analizar la interdependencia entre los distintos sistemas de control industrial.
- Diseñar e implementar métricas de seguridad con el objetivo de medir proactivamente el estado de eficiencia de los controles de seguridad.
- Evaluar las vulnerabilidades de los sistemas de control industrial y médico.

En cuanto al cumplimiento del objetivo de la investigación, se determina que la propuesta cumple con lo establecido, puesto que se presenta un diseño lógico de una solución a los problemas de seguridad en dispositivos de Internet de las Cosas, de acuerdo con los factores de seguridad y ámbito de aplicación escogidos.

4.2. Análisis de los resultados, retos, iniciativas y conclusiones.

En el presente, la seguridad en IIoT se basa en seguridad por la oscuridad: los responsables de los dispositivos confían en que están seguros mientras nadie sepa en qué dirección IP, en qué puerto y con qué *software* se puede acceder a los equipos. A pesar de ello, los últimos ataques de DDoS, considerados entre los más grandes registrados en la historia y perpetrados con la botnet Mirai, que aprovecha principalmente el uso de credenciales por defecto o en algunos casos hardcodeadas en el *firmware* para tomar control sobre los dispositivos de Internet de las Cosas, han demostrado que este enfoque es totalmente ineficiente y peligroso. De modo que propuestas como la presente resultan útiles para identificar vulnerabilidades básicas, pero que pueden desencadenar ataques a estos dispositivos.

Sin embargo, es claro que esta propuesta debe ser reforzada para que abarque todos los factores de seguridad que afectan los dispositivos IoT. Para ello, se debe incorporar nuevas herramientas o mejorar los recursos contemplados en los servicios en la nube. En este momento, se están desarrollando varias iniciativas para encontrar y prevenir problemas específicos con los IoT. Tales cambios se pueden incorporar en una segunda versión de la propuesta, una vez que se determine el nivel de aceptación de esta con el público.

A nivel general, existen varios retos que la industria alrededor de la fabricación de dispositivos IoT debe tener en cuenta. Como se afirma en el Capítulo 2 de este documento, el tema de la estandarización, la interoperabilidad y la fragmentación entre fabricantes afecta el desarrollo y adaptación de esta tecnología con los lineamientos de seguridad. Además, las soluciones disponibles están atadas a los fabricantes y en caso que estos desaparezcan, se pierde el soporte, al exponer a los usuarios a equipos desactualizados y vulnerables.

Existen varias iniciativas para la mejora de la seguridad en IoT, como la elaboración de *test beds* estandarizados entre fabricantes para garantizar la

interoperabilidad, la creación de estándares de desarrollo e implementación, una etiqueta de hardware seguro propuesto por la Comisión Europea o la creación de equipos de respuesta de emergencia informática (CERT) centrada en ICS como la creada por Kaspersky; los cuales son solo algunos ejemplos de medidas para mejorar los actuales problemas de la tecnología IoT.

Repitiendo lo indicado por John Villasenor [41], la conectividad no debe sobrepasar el tema de la seguridad, es decir, el ideal de todos los equipos interconectados no debe sobreponer u obviar el tema de la seguridad. Es un punto de sentido común, pero particularmente en este aspecto las empresas fabricantes han fallado. Además, es necesario centrarse en los nuevos desafíos de ciberseguridad que presentan los sistemas distribuidos, en el sentido de que los dispositivos IoT incrementan la superficie de ataque y potencian la creciente tendencia de sistemas descentralizados difíciles de proteger.

Asimismo, las soluciones tanto de IoT como de ciberseguridad, necesitan ser diseñadas contemplando a los vínculos no deseados como la regla y no la excepción, por lo tanto, se debe incluir, desde el primer bosquejo de la solución, las contramedidas a esta situación.

Otro punto por incluir desde la primera etapa del ciclo de desarrollo del *software,* consiste en la adopción de un enfoque hacia la ciberseguridad en múltiples niveles, con el fin de evitar que los sistemas tengan puntos de fallo únicos.

En resumen, el principal reto es lograr un Internet de las Cosas seguras, donde la tendencia se desarrolle en conjunto con la seguridad en todos los aspectos y que no sea relegada hasta que exista una vulnerabilidad conocida.

Glosario

IoE: *Internet of Everything* se define como la reunión de personas, procesos, datos y cosas para hacer conexiones en red más relevantes y valiosas que nunca, convirtiendo la información en acciones que crean nuevas capacidades, experiencias más ricas, y oportunidades económicas sin precedentes para las empresas, los individuos y los países.

***Sentient Spaces* o espacios sensibles**: espacios por los cuales, por medio de cámaras y otros sensores ambientales, el sistema identifica a los ocupantes de los espacios sensibles y determina sus preferencias y las acciones que pueden ocurrir, con el fin de hacer sugerencias o preventivamente tomar acciones.

***Open Hardware/Open source hardware*:** "Hardware abierto" o "hardware de código abierto", se refiere a las especificaciones de diseño de un objeto físico que tienen licencia de tal manera que dicho objeto puede ser estudiado, modificado, creado y distribuido por cualquier persona.

***Fuzzer* black box:** fuzz testing o Fuzzing es una técnica de pruebas de software de caja negra, que básicamente consiste en encontrar errores de implementación mediante la inyección mal formado / semi-malformación de datos de forma automatizada.

***Test beds* o Bancos de Prueba:** un banco de pruebas es una plataforma para la realización de la prueba rigurosa, transparente y replicable de las teorías científicas, las herramientas computacionales y las nuevas tecnologías. El término se utiliza en muchas disciplinas para describir la investigación experimental y nuevas plataformas y entornos de desarrollo de productos.

***Checksum*:** es un valor matemático calculado basándose en el contenido del mensaje. El checksum es creado por una serie de cálculos que convierte el

payload del mensaje en una cadena de dígitos fija llamados un valor hash. Ejemplos: MD5, comprobación de redundancia cíclica (CRC), y Segurity Hash Algorithm (SHA) 1 y 2. Este valor de suma de comprobación podría ser añadido al comienzo del mensaje del payload y la aplicación que recibe los mensajes calcula el checksum para verificar la integridad del mensaje.

Servidor de control: el servidor de control aloja el *software* de control de supervisión DCS o PLC que se comunica con los dispositivos de control de nivel inferior. El servidor de control de accesos de los módulos de control subordinados través de una red ICS. SCADA Server o Unidad Terminal Maestra (MTU). El servidor SCADA es el dispositivo que actúa como el maestro en un sistema SCADA. Unidades a distancia de terminales y dispositivos PLC (como se describen a continuación) situados en los sitios de campo remotas suelen actuar como esclavos.

Remote Terminal Unit (RTU): la RTU, también llamada una unidad de telemetría remota, es una unidad especial de control y adquisición de datos de uso diseñado para soportar estaciones remotas SCADA. Son dispositivos de campo que a menudo están equipados con interfaces de radio inalámbricas para soportar situaciones remotas donde las comunicaciones basadas en cable no están disponibles. A veces, los PLC se implementan como dispositivos de campo para servir como estaciones remotas; en este caso, el PLC se refiere a menudo como una RTU

.

Programmable Logic Controller (PLC): el PLC es una pequeña computadora industrial originalmente diseñado para realizar las funciones lógicas ejecutadas por el hardware eléctrico (relés, interruptores y temporizadores / contadores mecánicos). Los PLCs han evolucionado hasta convertirse en controladores con la capacidad de controlar procesos complejos, y se utilizan sustancialmente en los sistemas SCADA y DCS. Otros controladores utilizado sobre el terreno son

controladores de proceso y estaciones remotas; que proporcionan el mismo control que los PLC, pero están diseñados para aplicaciones de control específicos. En entornos SCADA, los PLCs a menudo se utilizan como dispositivos de campo, ya que son más económicos, versátiles, flexibles y configurable de lo que para fines especiales RTU.

Intelligent Electronic Devices **(IED):** Un IED es un sensor/actuador "inteligente" que contiene la inteligencia necesaria para adquirir datos, comunicarse con otros dispositivos, realizar el procesamiento y el control local. Un IED puede combinar un sensor analógico de entrada, salida analógica, capacidades de control de bajo nivel, un sistema de comunicación, y la memoria de programa en un solo dispositivo. El uso de los IED en los sistemas SCADA DCS y permite el control automático a nivel local.

Human-Machine Interface (HMI): Es el software y hardware que permite a los operadores humanos supervisar el estado de un proceso bajo control, modificar los ajustes de control para cambiar el objetivo de control, y de forma manual tienen prioridad sobre las operaciones de control automático en caso de una emergencia. El panel de operador también permite que un ingeniero de control o del operador para configurar los puntos de ajuste o algoritmos de control y los parámetros en el controlador. El operador también muestra la información del proceso de estado, información histórica, informes y otra información de operadores, administradores, gerentes, socios comerciales y otros usuarios autorizados. La ubicación, la plataforma y la interfaz pueden variar mucho. Por ejemplo, un HMI podría ser una plataforma dedicada en el centro de control, un ordenador portátil en una red LAN inalámbrica o un navegador en cualquier sistema conectado a Internet.

Data Historian: el historial de datos es una base de datos centralizada para el registro de toda la información de proceso dentro de un ICS. La información

almacenada en esta base de datos se puede acceder para apoyar diversos análisis, desde el control estadístico de procesos de planificación a nivel de empresa.

Input/Output (IO) Server: el servidor IO es un componente de control responsable de la recogida, el almacenamiento en búfer y el acceso a procesar la información de control de sub-componentes, tales como PLC, RTU y artefactos explosivos improvisados. Un servidor IO puede residir en el servidor de control o en una plataforma informática separada. Servidores IO también se utilizan para interconectar los componentes de control de terceros, tales como un HMI y un control servidor.

Fieldbus Network: la red de bus de campo enlaza sensores y otros dispositivos a un PLC u otro controlador. El uso de tecnologías de bus de campo elimina la necesidad de cableado punto a punto entre el controlador y cada dispositivo. Los dispositivos se comunican con el controlador de bus de campo usando una variedad de protocolos. Los mensajes enviados entre los sensores y el controlador identifican de forma exclusiva cada uno de los sensores.

Apéndice

Apéndice 1: recomendaciones de seguridad

1. The Cloud Security Alliance (CSA) Summary Guidance on Identity and Access Management: recomendaciones dadas por FTC para aplicar en plan de respuesta en caso de data *breach*.
    - Las medidas de seguridad tendrán que abordar el tipo de dispositivo, el tipo de datos que tiene acceso a, y el valor de esos datos.
    - Los empleados deben ser entrenados en los riesgos y medidas de seguridad para dispositivos IoT.
    - Los fabricantes de productos vinculados entre sí debe practicar minimización de los datos, así como recopilar y utilizar el mínimo necesario para que la función del producto según lo previsto.
    - Las empresas deben informar a los consumidores de posibles riesgos asociados a los dispositivos conectados, y proporcionarles orientación en el uso de productos de una manera segura.
    - Las amenazas de la IoT podrían aumentar el riesgo de entrar en conflicto con las leyes de privacidad y de violación de datos de una empresa.

2. Security Alliance [42].

Los desarrolladores de productos de la IO deben comenzar con las siguientes prácticas de ingeniería de seguridad:

- Diseñar e implementar un proceso de actualización / *software* de *firmware* seguro.
- Las interfaces de productos de seguridad con autenticación, protección de integridad y cifrado.
- Obtener una evaluación independiente de la seguridad de sus productos de IO.

- Asegurar las aplicaciones y / o puertas de enlace del compañero móviles que se conectan con sus productos de la IO (por ejemplo, el cifrado , privilegios y la autenticación).
- Implementar una raíz de confianza segura para las cadenas de raíces y claves privadas en el dispositivo.

3. Recomendaciones de seguridad de Symantec: Medidas de seguridad para mitigar/prevenir infecciones por Linux. Darlloz [19].
    - Aplicar los parches de seguridad para todo el software instalado en los equipos o dispositivos IoT.
    - Actualización de *firmware* en todos los dispositivos
    - Cambiar la contraseña de forma predeterminada en todos los dispositivos
    - Bloquear la conexión en el puerto 23 o 80 desde el exterior si no es necesario.
    - Desactivar la reproducción automática para evitar la puesta en marcha automática de los archivos ejecutables de las redes y unidades extraíbles, y desconectar las unidades cuando no es necesario. Si no se requiere acceso de escritura, habilitar el modo de sólo lectura si la opción está disponible.
    - Desactivar el uso compartido de archivos si no es necesario. Si se requiere el intercambio de archivos, utilizar las ACL y la protección de contraseña para limitar el acceso. Deshabilitar el acceso anónimo a las carpetas compartidas. Conceder acceso únicamente a cuentas de usuario con contraseñas a las carpetas que deben ser compartidos.
    - Desconectar y anular los servicios innecesarios. De forma predeterminada, muchos sistemas operativos instalan servicios auxiliares que no son críticos. Estos servicios son avenidas de

ataque. Si se quitan, las amenazas tienen menos posibilidades de ataque.

- Configurar el servidor de correo electrónico para bloquear o eliminar los mensajes que contengan archivos adjuntos que se utilizan comúnmente para distribuir virus, como archivos .vbs, .bat, .exe, .pif y .scr.
- Si Bluetooth no es necesario, debe estar apagado. Si se requiere su uso, la visibilidad del dispositivo debe estar en modo "indetectable", por lo tanto, no puede ser explorado por otros dispositivos Bluetooth. Si se debe utilizar el emparejamiento del dispositivo, se debe asegurar que todos los dispositivos se ajustan a "no autorizado", que exige una autorización para cada solicitud de conexión. No se debe aceptar aplicaciones que no estén firmadas o sean enviadas desde fuentes desconocidas.

3. ElevenPath, empresa de Telefónica: Empresa que forma parte de la IoT Security Foundation, indica estas recomendaciones en su reporte sobre inseguridad en IoT [16].
    - Usar conexiones seguras para la comunicación, hay métodos criptográficos eficientes diseñados para dispositivos a pequeña escala, tales como criptografía de curva elíptica (ECC).
    - Uso de datos de anónimos, cuando sea posible.
    - Permitir y fomentar el uso de contraseñas seguras alejándose PIN de 4 números.
    - Requerir que el usuario cambie la contraseña por defecto, no usar contraseñas codificadas de forma rígida.
    - Ofuscación de código si los usuarios puedan acceder a él.
    - Incluir reglas de autorización y control de acceso.
    - Proporcionar un proceso de actualización sencilla y segura con una cadena de confianza.

- Sólo recopilar datos que sean estrictamente necesarios.
- No obligar a los usuarios a utilizar una interfaz de nube si la funcionalidad del dispositivo no lo justifique.
- Prevenir los ataques de fuerza bruta en la etapa de inicio de sesión a través de medidas de bloqueo de cuentas.
- Evaluar el uso del dispositivo y examinar orientaciones tales como la verificación de seguridad de aplicaciones según OWASP.
- Evaluar la calidad de "vecindad" de Internet IP antes de la creación de la plataforma.
- Garantizar la mutua comprobación del certificado SSL y la lista de revocación de certificados.
- Implementar un mecanismo de seguridad inteligente cuando la conexión o la energía se pierde o se atascan.
- Evitar los puertos de entrada abiertos.
- El código debe ser verificado a través de una cadena de confianza.
- Usar cadena de arranque seguro para verificar todo el software que se ejecuta en el dispositivo.
- No transferir datos a terceros para otros fines sin la aprobación explícita y separar los datos de forma adecuada, a menos agregación para fines de análisis.

4. Recomendaciones de la IIRA[15]: recomendaciones para las Industrial Internet of Things (IIoT) [43].

- Recomienda desarrollar bancos de pruebas tecnológicos (*Test beds)* para demostrar como soluciones de diferentes organizaciones pueden trabajar juntas.
- Si bien a nivel industrial existe la serie de estándares de seguridad IEC 62443: Network and system security for industrial-process measurement and control, se recomienda enfatizar en estos puntos:

[15] Industria Internet Reference Architecture

- Aumentar la red de seguridad- para incluir opciones para el cifrado, con mayor énfasis en la seguridad de la aplicación y control de acceso para definir qué dispositivos se pueden conectar a la red y los permisos que tienen esos dispositivos.
- Los dispositivos habilitados para IIoT tendrán que soportar los protocolos de seguridad.
- Los dispositivos y el software deben implementar funciones de seguridad certificadas, pero estas características tienen que ser consistente.
- Garantizar la seguridad de la cadena de suministros, donde la codificación de productos, fabricación, suministro, instalación, mantenimiento y eliminación, todo en conjunto, se convertirá en una parte clave de la seguridad del mantenimiento del sistema.
- Servicios de seguridad para ayudar a los clientes en la adopción del nuevo sistema de seguridad.

6. ITU-T: Recomendación Y 2060 [1]

Según la recomendación Y 2060 existen dos tipos de capacidades de seguridad: genéricas y específicas. Las capacidades de seguridad genéricas son independientes de la aplicación y son, entre otras:

- En la capa de aplicación: autorización, autentificación, confidencialidad de datos de aplicación y protección de la integridad, protección de la privacidad, auditorías de seguridad y antivirus;
- En la capa de red: autorización, autentificación, confidencialidad de datos de señalización y de datos de uso, y protección de la integridad de señalización;

- En la capa de dispositivo: autenticación, autorización, validación de la integridad del dispositivo, control de acceso, confidencialidad de datos y protección de la integridad.
- Las capacidades de seguridad específicas están estrechamente relacionadas con los requisitos específicos de la aplicación, por ejemplo, los requisitos de seguridad para el pago con el móvil.

7. IBM: de acuerdo a IBM, las soluciones IoT están conformadas por tres niveles principales y cada nivel debe incorporar medidas de seguridad específicas [17].

•Nivel de Dispositivos/Gateways: proteger contra un servidor "falso" que envía comandos maliciosos, o proteger contra un *hacker* quien intenta escuchar los datos de sensores privados, los cuales están siendo enviados desde los dispositivos. Las consideraciones de seguridad son: llamada API segura, seguridad de Node-RED, cifrado de mensajes y aplicar mensaje de verificación de checksum.

•Nivel de Red/Transporte: proteger contra un dispositivo "falso" que envía mediciones falsas que pueden corromper los datos que persisten en la aplicación.

Las consideraciones de seguridad para este nivel son: autenticando dispositivos (sólo pueden enviar datos los dispositivos confiables), autorización, seguridad de la API (utilizar REST APIs, HTTPS (puerto 443), en lugar de HTTP (puerto 80), la clave de la API de la aplicación como el nombre de usuario y el correspondiente *token* de autenticación como la contraseña), configuración de seguridad y transporte seguro, como por ejemplo utilizar MQTT sobre TLS o MQTT Payload encryption, en caso de que no sea posible implementar TLS por restricciones de capacidad del dispositivo.

Para mejorar el desempeño de TLS en dispositivos con poca capacidad se recomienda establecer *long-lived connections* para evitar el

uso innecesario de ancho de banda, incluir TLS *session resumption*[16] para reanudar la sesión previa y garantizar uso de última versión de TLS. Asimismo, se exhorta a utilizar comprobaciones de integridad de mensaje cuando la comunicación no es de confianza entre los dispositivos y el *broker* (c*hecksums*, MAC o firmas digitales).

•Nivel de aplicaciones: proteger contra el uso inválido de datos, o proteger contra la manipulación de los procesos analíticos que se ejecutan en el nivel de la aplicación. Las consideraciones de seguridad para este nivel son: autenticación, cifrado del mensaje de carga, suministro y verificación de certificados, transporte MQTT seguro, arranque seguro, aplicación y actualización de firewalls y parches.

8. ICS-CERT [35]: *Industrial Control System Cyber Emergency Response Team* es un área del Departamento de Seguridad Interna de los Estados Unidos. En un reporte del 2012 sobre monitoreo de los sistemas de control industrial y los casos de malware encontrados, se indican las siguiente recomendaciones al sector:
    - Dueños y operadores de infraestructuras críticas deben de implementar políticas de seguridad para mantener los antivirus actualizados, gestionar el sistema de parches y controlar el uso de medio removibles (USB, CD, entre otros).
    - Implementar medidas de backup y recuperación de desastres.
    - Se debe minimizar la exposición de todos los dispositivos de control. Estos no deben estar directamente conectados a Internet.

[16] Session resumption es una técnica de reanudación de sesión de TLS que permite la reutilización de una sesión TLS ya negociada. Después de que vuelve a conectarse al servidor, el cliente y el servidor no necesitan realizar de nuevo *handshake* completo.

- Ubique las redes y los dispositivos de control de sistema detrás de un contrafuego. Esta área debe estar separada de la red de la empresa.
- Si el acceso remoto es necesario, se debe utilizar VPN para asegurar la comunicación.
- Borrar, deshabilitar o renombrar las cuentas por defecto.
- Implementar políticas de bloqueo de cuentas para evitar ataques de fuerza bruta.
- Monitorear la creación de cuentas administrativas o con privilegios por parte de terceros.
- Implementar políticas de contraseñas robustas.
- Adoptar ciclos regulares de parcheo para garantizar la implementación de las actualización de seguridad.

Anexos

Anexo 1: ejemplo de CVE

Siemens : Security Vulnerabilities (CVSS score between 7 and 7.99)

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2016-7113	399		DoS	2016-09-05	2016-09-06	7.8	None	Remote	Low	Not required	None	None	Complete
	The EN100 Ethernet module before 4.29 for Siemens SIPROTEC 4 and SIPROTEC Compact devices allows remote attackers to cause a denial of service (defect-mode transition) via crafted HTTP packets.													
2	CVE-2016-6486	264		+Priv	2016-08-07	2016-08-10	7.2	None	Local	Low	Not required	Complete	Complete	Complete
	Siemens SINEMA Server uses weak permissions for the application folder, which allows local users to gain privileges via unspecified vectors.													
3	CVE-2016-3949	399		DoS	2016-06-27	2016-08-18	7.8	None	Remote	Low	Not required	None	None	Complete
	Siemens SIMATIC S7-300 Profinet-enabled CPU devices with firmware before 3.2.12 and SIMATIC S7-300 Profinet-disabled CPU devices with firmware before 3.3.12 allow remote attackers to cause a denial of service (defect-mode transition) via crafted (1) ISO-TSAP or (2) Profibus packets.													
4	CVE-2016-2200	20		DoS	2016-02-08	2016-02-18	7.8	None	Remote	Low	Not required	None	None	Complete
	Siemens SIMATIC S7-1500 CPU devices before 1.8.3 allow remote attackers to cause a denial of service (STOP mode transition) via crafted packets on TCP port 102.													
5	CVE-2015-5698	352		CSRF	2015-08-30	2015-08-31	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	Cross-site request forgery (CSRF) vulnerability in the web server on Siemens SIMATIC S7-1200 CPU devices with firmware before 4.1.3 allows remote attackers to hijack the authentication of unspecified victims via unknown vectors.													
6	CVE-2015-5374	19		DoS	2015-07-18	2015-07-21	7.8	None	Remote	Low	Not required	None	None	Complete
	The EN100 module with firmware before 4.25 for Siemens SIPROTEC 4 and SIPROTEC Compact devices allows remote attackers to cause a denial of service via crafted packets on UDP port 50000.													
7	CVE-2015-2177	20		DoS	2015-03-06	2016-08-24	7.8	None	Remote	Low	Not required	None	None	Complete
	Siemens SIMATIC S7-300 CPU devices allow remote attackers to cause a denial of service (defect-mode transition) via crafted packets on (1) TCP port 102 or (2) Profibus.													
8	CVE-2014-9369	20		DoS	2015-03-06	2015-03-09	7.8	None	Remote	Low	Not required	None	None	Complete
	Siemens SPC controllers SPC4000, SPC5000, and SPC6000 before 3.6.0 allow remote attackers to cause a denial of service (device restart) via crafted packets.													
9	CVE-2014-8478	22		DoS Dir. Trav.	2015-01-21	2015-01-23	7.8	None	Remote	Low	Not required	None	None	Complete
	The web server on Siemens SCALANCE X-300 switches with firmware before 4.0 and SCALANCE X 408 switches with firmware before 4.0 allows remote attackers to cause a denial of service (reboot) via malformed HTTP requests.													
10	CVE-2014-5074			DoS	2014-08-17	2014-08-28	7.1	None	Remote	Medium	Not required	None	None	Complete
	Siemens SIMATIC S7-1500 CPU devices with firmware before 1.6 allow remote attackers to cause a denial of service (device restart and STOP transition) via crafted TCP packets.													
11	CVE-2014-2259			DoS	2014-03-16	2014-03-25	7.8	None	Remote	Low	Not required	None	None	Complete
	Siemens SIMATIC S7-1500 CPU PLC devices with firmware before 1.5.0 allow remote attackers to cause a denial of service (defect-mode transition) via crafted HTTPS packets.													

Schneider Electric : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2016-2278	284		Exec Code	2016-03-02	2016-03-04	9.0	None	Remote	Low	Single system	Complete	Complete	Complete
	Schneider Electric Struxureware Building Operations Automation Server AS 1.7 and earlier and AS-P 1.7 and earlier allows remote authenticated administrators to execute arbitrary OS commands by defeating an msh (aka Minimal Shell) protection mechanism.													
2	CVE-2015-6485	200		+Info	2016-03-11	2016-03-22	5.0	None	Remote	Low	Not required	Partial	None	None
	Schneider Electric Telvent Sage 2300 RTUs with firmware before C3413-500-S01, and LANDAC 8-2, Sage 1410, Sage 1430, Sage 1450, Sage 2400, and Sage 3030M RTUs with firmware before C3414-500-S0212, allow remote attackers to obtain sensitive information from device memory by reading a padding field of an Ethernet packet.													
3	CVE-2015-3962	310			2015-09-18	2015-09-23	5.0	None	Remote	Low	Not required	Partial	None	None
	Schneider Electric StruxureWare Building Expert MPM before 2.15 does not use encryption for the client-server data stream, which allows remote attackers to discover credentials by sniffing the network.													
4	CVE-2015-0999	200		+Info	2015-03-29	2015-03-30	2.1	None	Local	Low	Not required	Partial	None	None
	Schneider Electric InduSoft Web Studio before 7.1.3.4 SP3 Patch 4 and InTouch Machine Edition 2014 before 7.1.3.4 SP3 Patch 4 store cleartext OPC User credentials in a configuration file, which allows local users to obtain sensitive information by reading this file.													
5	CVE-2015-0998	200		+Info	2015-03-29	2015-03-30	3.3	None	Local Network	Low	Not required	Partial	None	None
	Schneider Electric InduSoft Web Studio before 7.1.3.4 SP3 Patch 4 and InTouch Machine Edition 2014 before 7.1.3.4 SP3 Patch 4 transmit cleartext credentials, which allows remote attackers to obtain sensitive information by sniffing the network.													
6	CVE-2015-0997	200		+Info	2015-03-29	2015-03-30	5.0	None	Remote	Low	Not required	Partial	None	None
	Schneider Electric InduSoft Web Studio before 7.1.3.4 SP3 Patch 4 and InTouch Machine Edition 2014 before 7.1.3.4 SP3 Patch 4 provide an HMI user interface that lists all valid usernames, which makes it easier for remote attackers to obtain access via a brute-force password-guessing attack.													
7	CVE-2015-0996	200		+Info	2015-03-29	2015-03-30	2.1	None	Local	Low	Not required	Partial	None	None
	Schneider Electric InduSoft Web Studio before 7.1.3.4 SP3 Patch 4 and InTouch Machine Edition 2014 before 7.1.3.4 SP3 Patch 4 rely on a hardcoded cleartext password to control read access to Project files and Project Configuration files, which makes it easier for local users to obtain sensitive information by discovering this password.													
8	CVE-2015-0982	119		Exec Code Overflow	2015-03-13	2015-03-16	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	Buffer overflow in an unspecified DLL in Schneider Electric Pelco DS-NVs before 7.8.90 allows remote attackers to execute arbitrary code via unspecified vectors.													
9	CVE-2014-9188	119		Exec Code Overflow	2014-12-27	2014-12-29	9.0	None	Remote	Low	Not required	Complete	Partial	Partial
	Buffer overflow in an ActiveX control in MDraw30.ocx in Schneider Electric ProClima before 6.1.7 allows remote attackers to execute arbitrary code via unspecified vectors, a different vulnerability than CVE-2014-8513 and CVE-2014-8514. NOTE: this may be clarified later based on details provided by researchers.													
10	CVE-2014-8514	119		Exec Code Overflow	2014-12-27	2014-12-29	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	Buffer overflow in an ActiveX control in MDraw30.ocx in Schneider Electric ProClima before 6.1.7 allows remote attackers to execute arbitrary code via unspecified vectors, a different vulnerability than CVE-2014-8513 and CVE-2014-9188. NOTE: this may be clarified later based on details provided by researchers.													
11	CVE-2014-8513	119		Exec Code Overflow	2014-12-27	2014-12-29	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	Buffer overflow in an ActiveX control in MDraw30.ocx in Schneider Electric ProClima before 6.1.7 allows remote attackers to execute arbitrary code via unspecified vectors, a different vulnerability than CVE-2014-8514 and CVE-2014-9188. NOTE: this may be clarified later based on details provided by researchers.													
12	CVE-2014-8512	119		Exec Code Overflow	2014-12-27	2014-12-29	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	Buffer overflow in an ActiveX control in Ats45.ocx in Schneider Electric ProClima before 6.1.7 allows remote attackers to execute arbitrary code via unspecified vectors, a different vulnerability than CVE-2014-8511. NOTE: this may be clarified later													

Bibliografía Específica

[1] Recomendación Y.2060 de la ITU, Uniòn Internacional de Telecomunicaciones, https://www.itu.int/rec/T-REC-Y.2060-201206-I/es (consultada el 2/6/2016)

[2] What is Internet of Things, TechTarget, http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT (consultada el 2/6/2016)

[3] Gary Barnett, Chief Analyst de OVUM. How IoT will change our world, and how to start building it- Digital Futures 2025. http://pt.slideshare.net/OvumTelecoms/how-iot-will-change-our-world-and-how-to-start-building-it-digital-futures-2025-56212160 (consultada el 4/6/2016)

[4] IoT Developer Survey, Eclipse *IoT Working Group*, IEEE IoT y AGILE IoT, http://iot.ieee.org/images/files/pdf/iot-developer-survey-2016-report-final.pdf (consultada el 4/6/2016)

[5] Gartner, "Gartner Identifies Four Fundamental Usage Models to Unlock Value from the Internet of Things, http://www.gartner.com/newsroom/id/2699017. (consultada el 9/6/2016)

[6] Cisco, Internet de las Cosas y la evolución de Internet. http://www.cisco.com/web/ES/campaigns/internet-de-las-cosas/index.html#~tab-tab3, (consultada el 9/6/2016)

[7] Gartner, Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020, http://www.gartner.com/newsroom/id/2636073, (consultada el 9/6/2016)

[8] Federal Trade Commission, https://www.ftc.gov/news-events/press-releases/2015/01/ftc-report-internet-things-urges-companies-adopt-best-practices (consultada el 9/6/2016)

[9] SAP. IoT and Digital Transformation: A Tale of Four Industries, http://go.sap.com/documents/2016/05/0eea93b3-707c-0010-82c7-eda71af511fa.html (consultada el 9/6/2016)

[10] TechTarget, http://searchdatacenter.techtarget.com/es/cronica/Los-primeros-usuarios-de-las-plataformas-IoT-de-SAP-reportan-beneficios-reales-de-negocio?utm_medium=EM&asrc=EM_EDA_62797649&utm_campaign=20160817_Hay%20beneficios%20reales%20de%20negocio%20en%20plataformas%20IoT%20de%20SAP,%20reportan%20usuarios_&utm_source=EDA (consultada el 9/6/2016)

[11] McKinsey Global Institute, Unlocking the potential of IoT, http://www.mckinsey.com/business-functions/business-technology/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world (consultada el 9/6/2016)

[12] Bank Info Security, The 'Internet of Things' as a Security Risk, http://www.bankinfosecurity.com/interviews/toys-in-attic-iot-as-security-risk-i-2783 (consultada el 10/6/2016)

[13] Deloitte, Flashpoint: Cyber risk in an Internet of Things world http://www2.deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/flashpoints-cyber-risk-in-an-internet-of-things-world-emerging-trends.html?id=us:2em:3na:fp04nsl:awa:tmt:110515 (consultada el 10/6/2016)

[14] TechTarget, Q&A: Address Internet of Things security concerns across an enterprise, http://internetofthingsagenda.techtarget.com/feature/QA-Address-Internet-of-Things-security-concerns-across-an-enterprise (consultada el 10/6/2016)

[15] Cloud Tweaks. Secure Third Party Access Still Not An IT Priority, http://cloudtweaks.com/2016/06/breaches-secure-third-party-access-still-not-priority/
(consultada el 10/6/2016)

[16] ElevenPath, Insecurity in the IoT https://www.elevenpaths.com/wp-content/uploads/2015/10/TDS_Insecurity_in_the_IoT.pdf (consultada el 10/6/2016)

[17] IBM, Diseñar y construir soluciones IoT seguras, http://www.ibm.com/developerworks/ssa/library/iot-trs-secure-iot-solutions1/iot-trs-secure-iot-solutions1-pdf.pdf (consultada el 10/6/2016)

[18] The Hacker News, http://thehackernews.com/2016/03/internet-of-thing-malware.html (consultada el 11/6/2016)

[19] Symantec, IoT worm, https://www.symantec.com/connect/blogs/iot-worm-used-mine-cryptocurrency (consultada el 11/6/2016).

[20] Kaspersky Lab, Bashlite Family Malware, https://threatpost.com/bashlite-family-of-malware-infects-1-million-iot-devices/120230/ (consultada el 11/6/2016)

[21] DEFCON 22, https://www.youtube.com/watch?v=RX-O4XuCW1Y (consultada el 11/6/2016)

[22] MalwareMustDie, Mirai DDos Troyano, http://news.softpedia.com/news/mirai-ddos-trojan-is-the-next-big-threat-for-iot-devices-and-linux-servers-507964.shtml (consultada el 12/6/2016)

[23] Experian, http://www.experian.com/blogs/data-breach/2015/08/11/following-personal-identifying-information-pii-down-the-black-net-road/ (consultada el 12/6/2016)

[24] Information Week, http://www.informationweek.com/government/big-data-analytics/data-protection-in-internet-of-things-era/d/d-id/1204428 (consultada el 12/6/2016)

[25] Kaspersky, https://securelist.lat/blog/investigacion/82864/hospitals-are-under-attack-in-2016/ (consultada el 12/6/2016)

[26] Hardsploit, The Project: a Framework to audit IoT devices security, https://hardsploit.io/the-project/ (consultada el 10/6/2016)

[27]Security Testing the Internet of Things –IoT .http://www.beyondsecurity.com/security_testing_iot_internet_of_things.html (consultada el 13/6/2016)

[28] World Economic Forum, Industrial Internet of Things: Unleashing the Potential of Connected Products and Services http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf (consultada el 16/6/2016)

[29] Kaspersky, Panorama de las amenazas contra la ciberseguridad industrial https://securelist.lat/analysis/publicaciones/83507/industrial-cybersecurity-threat-landscape/ (consultada el 16/6/2016)

[30] Dark Reading, Banking Trojans Disguised as ICS/SCADA Software Infecting Plants, http://www.darkreading.com/attacks-breaches/banking-trojans-disguised-as-ics-scada-software-infecting-plants/d/d-id/1318542 (consultada el 16/6/2016)

[31] Blog, Un informático en el lado del mal. http://www.elladodelmal.com/2016/10/hacking-de-dispositivos-iiot-industrial.html (consultada el 16/6/2016)

[32] CERTSI, Seguridad enICS, https://www.certsi.es/blog/scadalab-seguridad-ics (consultada el 16/6/2016)

[33] Ecole Polytechnique Montrèal, Typical ics/scada system, https://secsi.polymtl.ca/rsa2013/RSA-SCADA-JMF.pdf (consultada el 16/6/2016)

[34] NIST, NIST SP 800-82, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82.pdf (consultada el 27/7/2016)

[35] US-CERT, Secure Architecture Design Definition, https://ics-cert.us-cert.gov/Secure-Architecture-Design-Definitions#Field_Controller_RTU_PLC_IED (consultada el 27/7/2016)

[36] Shodan, https://www.shodan.io/explore/category/industrial-control-systems (consultada el 28/7/2016)

[37] SCADA Strange Love, Projecto SCADAPASS, http://scadastrangelove.blogspot.com.ar/2015/12/scadapass.html https://github.com/scadastrangelove/SCADAPASS (consultada el 28/7/2016)

[38] Blog Segu-Info, Mapeador pasivo de redes, http://blog.segu-info.com.ar/2016/07/mapeador-de-red-pasivo-de-entornos.html, (consultada el 28/7/2016)

[39] Dark Reading,http://www.darkreading.com/perimeter/using-free-tools-to-detect-attacks-on-ics-scada-networks/d/d-id/1318527?itc=edit_in_body_cross (consultada el 29/7/2016).

[40] Universidad de Ottawa, Gregor Bochmann, www.site.uottawa.ca/~bochmann/SEG3101/Notes/SEG3101-ch3-2 - Requirements documentation standards - IEEE830.ppt (consultada el 30/8/2016)

[41] Tecnología y Negocios, https://www.tecnologiaynegocios.com/topic/tecnologia/5-claves-para-que-iot-sea-el-internet-de-las-cosas-seguras (consultada el 10/10/2016)

[42]SecurityAlliance,https://downloads.cloudsecurityalliance.org/assets/research/internet-of-things/future-proofing-the-connected-world.pdf (consultada el 30/8/2016)

[43] Recomendaciones IIRA, http://blog.iiconsortium.org/2015/11/recap-of-the-industrial-internet-security-forum.html
http://blog.iiconsortium.org/2016/05/cyber-security-the-cornerstone-of-iiot-adoption.html (consultada el 01/9/2016)

Bibliografía General

Curso IoT de ECI 2016, Universidad de Buenos Aires. Profesor Gabriel Carro. https://ioteci2016.wordpress.com/acerca-de/

Matherly, J. Complete Guide to Shodan, Lean Publishing, 2016.

SANS Institute InfoSec Reading Room, Using and Configuring Security Onion.