Lineamientos de Seguridad IoT para el Ecosistema de Dispositivos Periféricos IoT

Lineamientos de Seguridad IoT para el Ecosistema de Dispositivos Periféricos IoT

Versión 2.0

26 de Octubre 2017

Non-binding Permanent Reference Document

Clasificación de Seguridad: No Confidencial

El acceso y distribución de este documento está restringido a las personas permitidas por la clasificación de seguridad. Este documento es confidencial para la Asociación y está sujeto a la protección de derechos de autor. Este documento se utilizará únicamente para los fines para los que ha sido suministrado y la información contenida en él no debe divulgarse ni ponerse a disposición en ninguna otra forma posible, en su totalidad o en parte, a personas distintas a las permitidas bajo la clasificación de seguridad sin la aprobación previa por escrito de la Asociación.

Aviso Legal

La Asociación GSM ("Asociación") no acepta ninguna responsabilidad por la representación, garantía o compromiso (expreso o implícito) con respecto al contenido de este documento así como por la exactitud o integridad o actualidad de la información. La información contenida en este documento puede estar sujeta a cambios sin previo aviso.

Aviso Antimonopolio

La información aquí contenida es conforme a la política de cumplimiento antimonopolio de la Asociación GSM.

Tabla of Contenidos

1 Introducción 5
1.1 Introducción al Conjunto de Documentos sobre la Seguridad en IoT de la GSMA 5
1.2 Objetivo del Documento 6
1.3 Audiencia a la que se Dirige el Documento 6
1.4 Definiciones 7
1.5 Abreviaciones 8
1.6 Referencias 9
2 El Reto de la Seguridad en Dispositivos Periféricos de IoT 11
2.1 Bajo Consumo de Energía 11
2.2 Barato 11
2.3 Larga Duración (>10 Años) 11
2.4 Accesible Físicamente 12
3 El Modelo IoT de un Dispositivo Periférico 12
3.1 El Dispositivo Periférico Ligero 13
3.2 El Dispositivo Periférico Complejo 14
3.3 La Pasarela (o “Hub”) 14
3.4 El Modelo Global 15
4 El Modelo de Seguridad 16
4.1 Ataques a la red de Comunicaciones 17
4.2 Ataques a los Servicios de Red Accesibles 17
4.3 Ataques sobre el Acceso a la Consola 18
4.4 Ataque a las Comunicaciones del Bus Local 19
4.5 Ataques de Acceso Físico al Chip 19
5 Preguntas Frecuentes de Seguridad 20
5.1 ¿Cómo se Combate la Clonación? 20
5.2 ¿Cómo Protejo la Identidad de un Dispositivo Periférico? 20
5.3 ¿Cómo Reduzco el Impacto de un Ataque contra el Ancla de Confianza? 21
5.4 ¿Cómo Reduzco la Probabilidad de que se Suplante un Dispositivo Periférico? 21
5.5 ¿Cómo Evito la Posibilidad de que se Suplanten Servicios o Dispositivos (Pares)? 22
5.6 ¿Cómo Evito la Manipulación del Firmware y Software? 22
5.7 ¿Cómo Reduzco la Posibilidad de que se Ejecute Código Remotamente? 22
5.8 ¿Cómo Deshabilito la Depuración no Autorizada o la Instrumentación de la Arquitectura? 23
5.9 ¿Cómo debo Manejar los Ataques de Canal Lateral? 23
5.10 ¿Cómo Debo Implementar una Gestión Remota Segura? 24
5.11 ¿Cómo Detecto Dispositivos Periféricos Comprometidos? 24
5.12 ¿Cómo implemento de Manera Segura un Dispositivo sin una Conexión al Back-end? 24

5.13 ¿Cómo Aseguro la privacidad del Consumidor/Usuario? 25
5.14 ¿Cómo Aseguro la Protección de un Usuario Mientras Fuerzo la Privacidad y la Seguridad? 25
5.15 ¿Qué Problemas no Podría Esperar Resolver? 26
6 Recomendaciones Críticas 26
6.1 Implementar una Base de Computación Confiable para los Dispositivos Periféricos 26
6.2 Usar un Ancla de Confianza 31
6.3 Usar un Ancla de Seguridad contra Manipulaciones Físicas 33
6.4 Utilizar una API para Acceder a la TCB 34
6.5 Definir una Raíz de Confianza Organizativa 35
6.6 Personalizar cada Dispositivo Periférico antes de Ponerlo en Marcha Comercialmente 36
6.7 Implementar una Plataforma de Ejecución Mínima Viable (Recuperación de la Aplicación) 38
6.8 Provisionar cada Dispositivo Periférico de Manera Única 39
6.9 Gestionar las Contraseñas en los Dispositivos Periféricos 40
6.10 Utilizar un Generador de Números Aleatorios Comercial 41
6.11 Firmar Criptográficamente las Imágenes de las Aplicaciones 42
6.12 Gestionar de Manera Remota el Dispositivo Periférico 43
6.13 Implementar Funciones de Registro y Diagnóstico 44
6.14 Forzar la Protección de Memoria 45
6.15 Forzar el Arranque Fuera de la EEPROM Interna 45
6.16 Bloquear las Secciones de Memoria Críticas 46
6.17 Evitar los Gestores de Arranque Inseguros 47
6.18 Implementar la Transmisión de Claves con Secreto Perfecto hacia Adelante (PFS) 48
6.19 Implementar Comunicaciones Seguras entre los Dispositivos Periféricos 49
6.20 Autenticar la Identidad de los Dispositivos Periféricos 50
7 Recomendaciones de Alta Prioridad 52
7.1 Uso de la Memoria Interna para los Secretos 52
7.2 Detección de Anomalías 53
7.3 Usar un Encapsulado o Carcasa de Producto a Prueba de Manipulaciones 54
7.4 Forzar la Confidencialidad y la Integridad, desde y hacia el Ancla de Confianza 56
7.5 Actualizaciones OTA de las Aplicaciones 57
7.6 Autenticación Mutua Mal Diseñada o Sin Implementar 59
7.7 Gestión de la Privacidad 61
7.8 Privacidad e Identidades Únicas para los Dispositivos Periféricos 62
7.9 Ejecutar las Aplicaciones con Niveles de Privilegio Apropiados 63
7.10 Hacer Cumplir la Separación de Funciones en la Arquitectura de las Aplicaciones 63
7.11 Hacer Cumplir la Seguridad de los Lenguajes de Programación 65

7.12 Implementar Auditorias de Seguridad Tipo “Pentesting” (Pruebas de Penetración) Persistente 65
8 Recomendaciones de Prioridad Media 66
8.1 Imponer las Mejoras sobre el Nivel de Seguridad en los Sistemas Operativos 66
8.2 Deshabilitar las Tecnologías de Depuración y Pruebas 67
8.3 Corrupción del Contenido de la Memoria a través de Ataques a los Periféricos 68
8.4 Seguridad de la Interfaz de Usuario 69
8.5 Auditorías de Código Externas 70
8.6 Utilizar un APN Privado 71
8.7 Implementar Umbrales de Bloqueo Relativos a las Condiciones del Entorno (Ambientales) 72
8.8 Forzar Umbrales del Consumo de Energía 74
8.9 Entornos sin Conectividad al Back-end 74
8.10 Desactivación y Retirada del Mercado de Dispositivos 75
8.11 Captura de Metadatos no Autorizada 77
9 Recomendaciones de Baja Prioridad 78
9.1 Denegación de Servicio Intencional e Involuntaria 78
9.2 Análisis sobre la Protección Crítica de un Dispositivo 79
9.3 Como Frustrar los Ataques de Imitación de Componentes y Pasarelas no Fiables 80
9.4 Frustrar un Ataque de Arranque en Frío 81
9.5 Riesgos de seguridad no Obvios (“Ver a Través de las Paredes”) 82
9.6 Lucha contra Haces de Iones Focalizados (FIB) y Rayos X 83
9.7 Asegurar la Cadena de Suministro 85
9.8 Interceptación Legal 86
10 Resumen 87
Anexo A Ejemplo de Uso de una Arquitectura de Arranque Genérica (“Bootstrap”) 88
Anexo B Tutorial sobre el Uso de Tarjetas UICC en un Servicio de IoT 90
Anexo C Gestión del Documento 91
C.1 Historia del Documento 91
C.2 Otra Información 91

1 Introducción

1.1 Introducción al Conjunto de Documentos sobre la Seguridad en IoT de la GSMA

Este documento es una parte de un conjunto de documentos sobre lineamientos de seguridad IoT de la GSMA que están destinados a ayudar a la pujante industria del "Internet de las cosas" (IoT) a establecer una comprensión común alrededor de los problemas de seguridad del IoT. El conjunto de documentos de lineamientos no vinculantes promueve una metodología para el desarrollo de servicios de IoT seguros que permite aplicar las mejores prácticas con respecto a la seguridad durante todo el ciclo de vida del servicio. Los documentos proporcionan recomendaciones sobre cómo reducir las amenazas comunes de seguridad y vulnerabilidades dentro de los servicios de IoT.

A continuación, se muestra la estructura del conjunto de documentos de lineamientos de seguridad IoT de la GSMA. Se recomienda que el documento 'CLP.11 Descripción General de los lineamientos de Seguridad IoT de la GSMA' [1] se lea primero como base para entender los conceptos básicos antes de leer los otros documentos CLP.12 [2] y CLP.13 [3] (este documento).

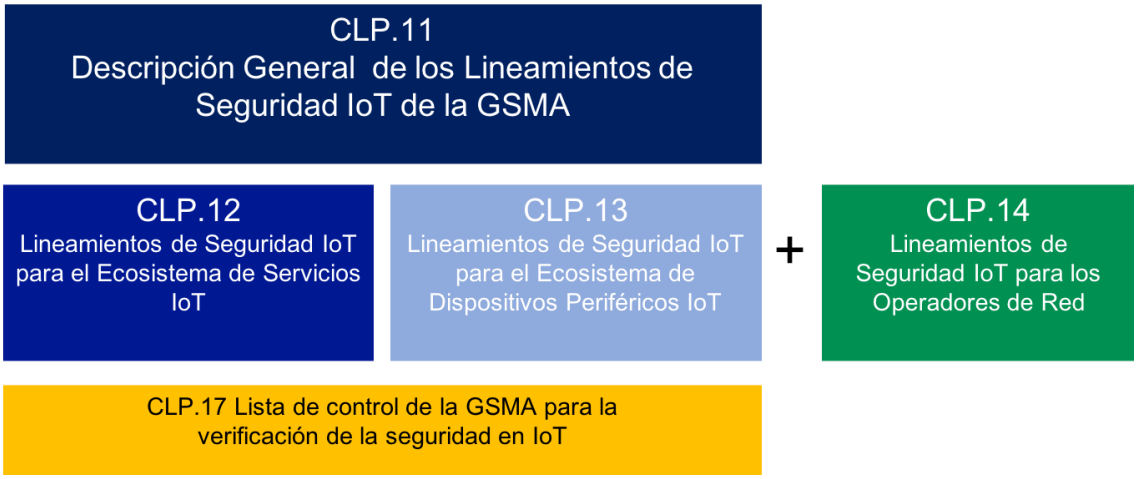

Figura 1 - Estructura del Conjunto de Documentos

Se aconseja a los operadores de red, proveedores de servicios IoT y otros socios en la cadena de valor de IoT, leer el documento GSP CLP.14 "Lineamientos de seguridad del IoT para operadores de red" [4] que proporciona lineamientos de seguridad de alto nivel a operadores de red que pretenden proporcionar servicios a proveedores de servicios IoT, para garantizar la seguridad del sistema y la privacidad de los datos.

1.1.1 Lista de control de la GSMA para la verificación de la seguridad en IoT

Una lista de control y verificación de seguridad IoT se puede encontrar en el documento CLP.17 [19]. Este documento permite a los proveedores de productos, servicios y componentes de IoT comprobar que sus productos son conformes a los lineamientos de seguridad IoT de la GSMA.

Al rellenar la lista de verificación arriba mencionada permitirá a cualquier entidad o empresa demostrar las medidas de seguridad que han tomado para proteger sus productos, servicios y componentes de los riesgos de Cyber-seguridad.

Se pueden obtener constancias de verificación enviando a la GSMA una declaración rellena con los puntos de la lista. Por favor vea el siguiente link en el portal de la GSMA para más información:

https://www.gsma.com/iot/future-iot-networks/iot-security-guidelines/

1.2 Objetivo del Documento

Esta guía se utilizará para evaluar todos los componentes en un producto de IoT o servicio desde la perspectiva del Ecosistema de Dispositivos Periféricos. Un Dispositivo Periférico, desde una perspectiva de IoT, es un dispositivo físico de computación que realiza una función o tarea como parte de un producto o servicio conectado a Internet. Un Dispositivo Periférico, por ejemplo, podría ser un dispositivo de “fitness” portátil, un sistema de control industrial, una unidad telemática de un automóvil o incluso un dron personal. Todas las tecnologías utilizadas para construir el dispositivo físico se evaluarán en cuanto a los riesgos de seguridad. El resultado es un conjunto práctico de lineamientos de diseño que permiten al lector identificar y evitar casi todos los riesgos potenciales de seguridad para el servicio de IoT.

El alcance de este documento se limita a las recomendaciones relacionadas con el diseño y la implementación de Dispositivos Periféricos de IoT.

Este documento no pretende impulsar la creación de nuevas especificaciones o estándares de IoT, sino que hará referencias a soluciones, estándares y mejores prácticas actualmente disponibles.

Este documento no pretende acelerar la obsolescencia de los Servicios de IoT existentes. La compatibilidad con versiones anteriores de los Servicios de IoT existentes del Operador de Red debe mantenerse cuando se consideren adecuadamente protegidas.

Se hace notar que el cumplimiento de las leyes y reglamentos nacionales para un territorio en particular pueden, en cualquier momento, anular las pautas establecidas en este documento.

1.3 Audiencia a la que se Dirige el Documento

A quien se dirige este documento:

- Proveedores de servicios de IoT: empresas u organizaciones que buscan desarrollar productos y servicios conectados nuevos e innovadores. Algunos de los muchos campos en los que operan los proveedores de servicios IoT incluyen hogares inteligentes, ciudades inteligentes, automoción, transporte, salud, servicios públicos y productos electrónicos de consumo.
- Fabricantes de dispositivos IoT: desde los desarrolladores de HW IoT a proveedores de servicios IoT para hacer posibles los servicios IoT.

- Desarrolladores de IoT que crean servicios de IoT para los proveedores de servicios de IoT.
- Operadores de red que brindan servicios a proveedores de servicios de IoT o que son a su vez proveedores directos de servicios IoT.

1.4 Definiciones

Término	Descripción
Nombre del punto de acceso	Identificador de un punto de conexión de red al que se conecta un dispositivo periférico. Están asociados a diferentes tipos de servicio, y en muchos casos son configurados por el operador de red.
Atacante o “hacker”	Un pirata informático, un agente inteligente (atacante), un atacante, un estafador u otra amenaza maliciosa para un servicio de IoT. Esta amenaza podría provenir de un solo delincuente, del crimen organizado, por terrorismo, de gobiernos hostiles y sus agencias, por espionaje industrial, de grupos de piratería, de activistas políticos, de hackers 'aficionados', investigadores, así como infracciones de seguridad y privacidad no intencionadas.
Celular	Cualquier tecnología de red móvil estándar 3GPP (por ejemplo, GSM, UMTS, LTE (inc. LTE-M) y NB-IoT).
La Nube	Una red de servidores remotos en Internet que aloja, almacena, administra y procesa aplicaciones y sus datos.
Dispositivo Periférico Complejo	Este modelo de Dispositivo Periférico tiene una conexión persistente a un servidor centralizado a través de un enlace de comunicaciones WAN, como un enlace Celular, por satélite o a través de una conexión cableada, como Ethernet. Ver la sección 3.
SIM Embebido	Una SIM que no puede ser eliminada o sustituida físicamente dentro de un dispositivo y permite el cambio seguro de perfiles según la especificación de la GSMA SGP.01 [2].
Dispositivo Periférico IoT	Un dispositivo con capacidad de computo que realiza una función o tarea como parte de un producto conectado o servicio de Internet. Ver sección 3 para una descripción de las tres clases comunes de dispositivos de IoT y ejemplos de cada clase de dispositivo periférico.
Internet de las cosas	El Internet de las cosas (IoT) describe la coordinación entre múltiples máquinas, dispositivos y aparatos conectados a Internet a través de múltiples redes. Estos dispositivos incluyen objetos cotidianos tales como tabletas y electrónica de consumo y otros dispositivos o máquinas tales como vehículos, monitores y sensores equipados con capacidades de comunicación que les permitan enviar y recibir datos.
Servicio IoT	Cualquier programa de computadora que utiliza datos desde dispositivos de IoT para prestar el servicio.
Ecosistema de servicios IoT	El conjunto de servicios, plataformas, protocolos y otras tecnologías necesarias para proporcionar capacidades y recopilar datos de puntos periféricos (servidores) implementados a “pie de campo””. Ver CLP11 [1] para más información.
Proveedor de un Servicio IoT	Las empresas u organizaciones que buscan desarrollar nuevos productos y servicios conectados innovadores.

Término	Descripción
Operador de Red	El operador y propietario de la red de comunicaciones que conecta un dispositivo periférico de IoT a un ecosistema de servicios IoT.
Raíz de Confianza Organizativa	Un conjunto de políticas criptográficas y procedimientos que dictan cómo las identidades, las aplicaciones y comunicaciones pueden y deben asegurarse mediante cifrado.
Punto de Acceso al Servicio	Un punto de entrada en la infraestructura de back-end de un Servicio IoT a través de una red de comunicaciones.
Módulo de identificación de suscriptor (SIM)	La tarjeta inteligente utilizada por una red móvil para autenticar dispositivos para su conexión a la red móvil y acceso a servicios de red.
Ancla de Confianza	En sistemas criptográficos con estructura jerárquica, un ancla de confianza es una entidad para la autorización de la cual se asume la confianza y no se deriva.
Base de Computador Confiable	Una Base de Computación Confiable (TCB) es un conglomerado de algoritmos, políticas y secretos dentro de un producto o servicio. La TCB actúa como un módulo que permite que el producto o servicio mida su propia fiabilidad, mida la autenticidad de los pares de la red, verifique la integridad de los mensajes enviados y recibidos a/o desde el producto o servicio, y más. La TCB funciona como la plataforma de seguridad principal sobre la cual se pueden construir productos y servicios seguros. Los componentes de una TCB cambiarán según el contexto (una TCB hardware para Dispositivos Periféricos o una TCB software para servicios en la nube), pero los objetivos, servicios, procedimientos y políticas abstractas deberían ser muy similares.
Entorno de Ejecución Confiable (TEE)	Un entorno que se ejecuta a través de un sistema operativo complejo y le proporciona servicios de seguridad. Existen múltiples tecnologías que se pueden usar para implementar un TEE, y el nivel de seguridad que se alcanza depende de la implementación en cuestión.
UICC	Elemento seguro entendido como plataforma especificada en ETSI TS 102 221 que puede soportar múltiples aplicaciones de autenticación estandarizadas para una red o servicio dentro de distintos dominios de seguridad. Puede ser integrada y encapsulada en varios formatos especificados en ETSI TS 102 671.

1.5 Abreviaciones

Término	Descripción
3GPP	Asociación de proyectos de 3ra generación (“3rd Generation Project Partnership”)
AC	Corriente Alterna (“Alternating Current”)
API	Interfaz del programa de Aplicación (“Application Program Interface”)
APN	Nombre del punto de Acceso (“Access Point Name”)
BLE	Bluetooth Baja Potencia (“Bluetooth Low Energy”)
BT	Bluetooth
CLP	Programa de la Vida Conectada (“GSMA’s Connected Living Programme”)
CPE	Equipo de Cliente (“Customer Premises Equipment”)

Término	Descripción
CPU	Unidad de Proceso Central (“Central Processing Unit”)
EEPROM	Memoria de Sólo Lectura Programable con Borrado Eléctrico (“Electrically Erasable Programmable Read-Only Memory”)
eUICC	UICC Integrado (“Embedded UICC”)
FIB	Haz de Iones de Barrido Ultrasónico (“Focused Ion Beam”)
GBA	Arquitectura genérica de Bootstrapping (“Generic Bootstrapping Architecture”)
GPS	Sistema de Posicionamiento Global (“Global Positioning System”)
GSMA	Asociación GSM (“GSM Association”)
IoT	Internet de la cosas (“Internet of Things”)
IP	Protocolo Internet (“Internet Protocol”)
ISM	Industrial, Científico y Médico (“Industrial, Scientific and Medical”)
LAN	Red de Area Local (“Local Area Network”)
LPWA	Bajo Consumo Area Extendida (“Low Power Wide Area”)
LTE-M	Evolución del Largo Plazo para Máquinas (“Long Term Evolution for Machines”)
MCU	Unidad de Micro-Controlador (“MicroController Unit”)
NB-IoT	Banda Estrecha-Internet de la Cosas (“Narrowband-Internet of Things”)
NVRAM	Memoria de Acceso Aleatorio No-Volátil (“Non-Volatile Random Access Memory”)
OMA	Alianza de moviles abierta (“Open Mobile Alliance”)
PAN	Red de Area Personal (“Personal Area Network”)
PSK	LLave Pre-compartida (“Pre-Shared Key”)
RAM	Memoria de Acceso Aleatorio (“Random Access Memory”)
ROM	Memoria de Solo Lectura (“Read Only Memory”)
SCADA	Supervisión, Control y Adquisición de Datos (“Supervisory Control And Data Acquisition”)
SPI	Interfaz Periférica Serie (“Serial Peripheral Interface”)
SSH	intérprete de órdenes seguro (“Secure Shell”)
SIM	Módulo de Identificación de Usuario (“Subscriber Identity Module”)
SRAM	Memoria Estática de Acceso Aleatorio (“Static Random Access Memory”)
TCB	Base de Computación Confiable (“Trusted Computing Base”)
TTL	Lógica Transisto-Transistor (“Transistor–Transistor Logic”)
UART	Receptor/Transmisor Universal Asíncrono (“Universal Asynchronous Receiver/Transmitter”)

1.6 Referencias

Ref	Número de Documento	Título
[1]	CLP.11	IoT Security Guidelines Overview Document
[2]	CLP.12	IoT Security Guidelines for IoT Service Ecosystem

Ref	Número de Documento	Título
[3]	CLP.13	IoT Security Guidelines for IoT Dispositivo PeriféricoEcosystem
[4]	CLP.14	IoT Security Guidelines for Network Operators
[5]	OMA FUMO	OMA Firmware Update Management Object www.openmobilealliance.org
[6]	na	ST-LINK/V2 in-circuit debugger/programmer http://www.st.com/
[7]	na	Mobile IoT Initiative https://www.gsma.com/iot/mobile-iot-initiative/
[8]	na	Nmap Security Scanner https://nmap.org/
[9]	CLP.03	IoT Device Connection Efficiency Guidelines https://www.gsma.com/iot/gsma-iot-device-connection-efficiency-guidelines/
[10]	na	Federal Information Processing Standards www.nist.gov/itl/fips.cfm
[11]	na	EMVCo www.emvco.com/
[12]	na	SIM Alliance - Open Mobile API simalliance.org/key-technical-releases/
[13]	GPD_SPE_013	GlobalPlatform Secure Element Access Control www.globalplatform.org/specificationsdevice.asp
[14]	GPD_SPE_024	GlobalPlatform Trusted Execution Environment API Specification www.globalplatform.org/specificationsdevice.asp
[15]	GPC_SPE_034	GlobalPlatform Card Specification www.globalplatform.org/specificationscard.asp
[16]	ISO/IEC 29192-1	Information technology -- Security techniques -- Lightweight cryptography www.iso.org/obp/ui/#iso:std:iso-iec:29192:-1:ed-1:v1:en
[17]	TS 33.220	Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) www.3gpp.org
[18]	TS 33.222	Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) www.3gpp.org
[19]	CLP.17	GSMA IoT Security Assessment Checklist https://www.gsma.com/iot/iot-security-assessment/
[20]	TS-0003	oneM2M Security Solutions www.onem2m.org

Ref	Número de Documento	Título
[21]	3GPP TS33.163	Battery efficient Security for very low Throughput Machine Type Communication (MTC) devices (BEST) www.3GPP.org
[22]	na	http://www.blackhat.com/presentations/bh-usa-08/McGregor/BH_US_08_McGregor_Cold_Boot_Attacks.pdf

2 El Reto de la Seguridad en Dispositivos Periféricos de IoT

El desafío de seguridad característico de un servicio IoT está, en muchos casos, directamente relacionado con las características específicas del Dispositivo Periférico IoT utilizado por el servicio. Por ejemplo, muchos Dispositivos Periféricos IoT tienen desafíos de seguridad particulares, presentan las siguientes características:

2.1 Bajo Consumo de Energía

- Es posible que se requiera un consumo de energía muy bajo para prolongar la duración de la batería (varios años) dentro de un Dispositivo Periférico remoto, inaccesible y sin un suministro de energía permanente, o porque el dispositivo tiene un suministro de energía constante, pero limitado, como por ejemplo a través de energía solar.
- Los Dispositivos Periféricos de bajo consumo normalmente solo pueden realizar operaciones criptográficas computacionalmente simples (por ejemplo, el Dispositivo solo admite las operaciones criptográficas ligeras definidas en ISO / IEC 29192 [16]) debido a los altos requisitos de consumo de energía asociados con las operaciones criptográficas más avanzadas y por otro lado solo admite comunicaciones de ancho de banda limitado, disminuyendo nuevamente la capacidad criptográfica.

2.2 Barato

- El argumento comercial para muchos Servicios de IoT exige que el costo del Dispositivo Periférico de IoT se mantenga bajo. Esto a menudo da como resultado que el dispositivo tenga una capacidad de procesamiento limitada, pequeñas cantidades de memoria y un sistema operativo restringido. El resultado final es que el dispositivo no es apto para implementar una criptografía como la utilizada en el Internet actual.

2.3 Larga Duración (>10 Años)

- Muchos Dispositivos, particularmente para aplicaciones públicas e industriales (por ejemplo, un medidor de gas inteligente), deben estar activos durante mucho tiempo (varios años sin cambiarles la batería). Esto presenta un desafío porque las elecciones de diseño criptográfico realizadas cuando se diseña el dispositivo tendrán que ser lo suficientemente robustas durante la vida útil del dispositivo. Por ejemplo, es probable que la capacidad de proceso por Dólar disponible para un atacante durante un período de 10 años haya aumentado 16 veces, mientras que las capacidades del dispositivo probablemente permanezcan estáticas.

- La gestión de dispositivos con una vida útil muy larga (varios años) también es un reto, especialmente si se encuentra una vulnerabilidad de seguridad que no se puede modificar dentro del Dispositivo Periférico.

2.4 Accesible Físicamente

- Muchos Dispositivos Periféricos de IoT son físicamente accesibles para el atacante. Por lo tanto, todos los componentes de hardware e interfaces en estos dispositivos están potencialmente sujetos a ataques y el desarrollador debe protegerlos (por HW y/o SW).

El resultado de todo esto es que se encuentran en muchos servicios IoT limitaciones: los Dispositivos Periféricos IoT no están normalmente conectados directamente a redes de comunicaciones tipo WAN (Red de Área Extendida) y muchos Dispositivos Periféricos de IoT no tienen posibilidad de ejecutar protocolos de Internet (IP). Por ejemplo, un Dispositivo Periférico de IoT puede usar un transceptor de radio industrial, científico y médico (ISM) para transferir datos a una pasarela IoT de servicios localmente que luego almacena los datos y los transmite a la red de comunicaciones mediante un protocolo IP, esto complica el proceso de hacer segura la comunicación de extremo a extremo.

Dependiendo de las capacidades del Dispositivo Periférico de IoT y los riesgos de seguridad asociados, es posible que sea necesario aplicar diferentes métodos para la seguridad con diversos grados de complejidad, como se explica más adelante en este documento.

3 El Modelo IoT de un Dispositivo Periférico

El modelo IoT de Dispositivos Periféricos, alguna vez considerado como un conjunto de tecnologías muy dispares, interactuando con el mundo físico y conectándose a un servidor en “algún lugar” de Internet para obtener información de cómo actuar y enviar parámetros provenientes de sensores, ha cambiado drásticamente. En la ingeniería moderna, la tecnología del IoT se ha integrado en un modelo predecible compuesto solo por varias soluciones características. El Dispositivo Periférico de IoT también se está volviendo más predecible, y se espera que tome una de las siguientes configuraciones:

- Un Dispositivo Periférico ligero
- Un Dispositivo Periférico complejo
- Una Pasarela (o "Hub")

En el siguiente diagrama se muestran algunas configuraciones típicas de un Dispositivo Periférico:

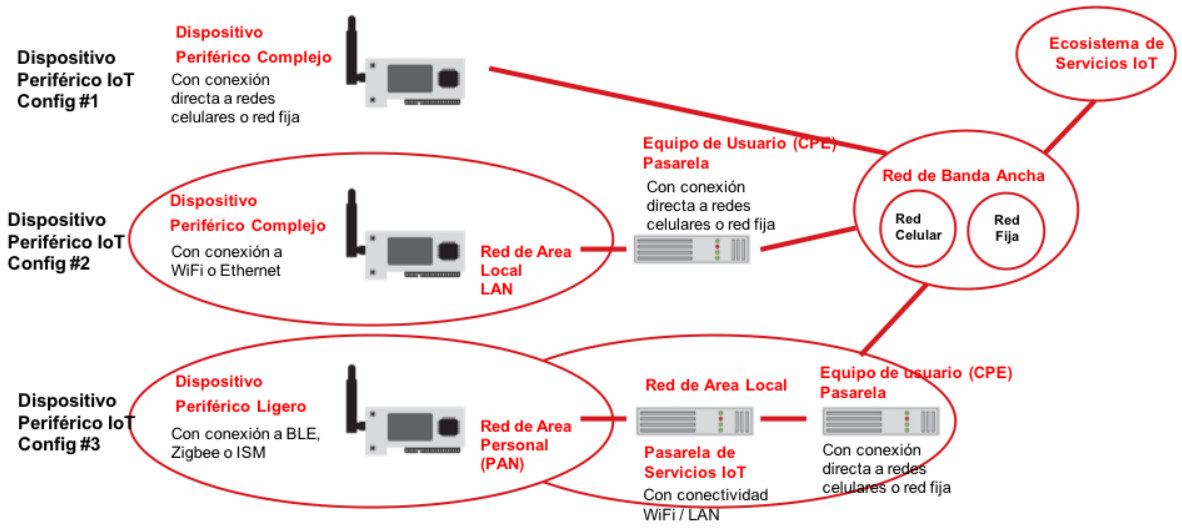

Figura 2 - Ejemplos de Configuraciones para Dispositivos Periféricos

3.1 El Dispositivo Periférico Ligero

Este tipo de Dispositivo Periférico es típicamente un sensor o dispositivo físico simple, como un interruptor de luz o un cerrojo de una puerta que tiene pocas funcionalidades. Su objetivo es servir a un propósito físico específico (comando) y proporcionar medidas/parámetros al ecosistema de servicios de IoT o al usuario final. Comúnmente utiliza una unidad de procesamiento (CPU) muy económica, posiblemente un microcontrolador de ocho bits, y una red de área personal (PAN) o un protocolo de red tipo capilar para la conectividad, como por ejemplo, Bluetooth Low Energy (BLE), Thread o Zigbee. Por lo general, es un dispositivo de baja potencia y puede funcionar con una batería o pila tipo botón, con energía solar o con una batería pequeña de polímero de litio. Estos dispositivos generalmente están conectados al Ecosistema de Servicios a través de pasarelas de servicios IoT y a una pasarela tipo router de cliente en serie, como se muestra en el ejemplo de configuración de 'Dispositivo Periférico IoT Config #3' en la figura 2.

Ejemplos de Dispositivos Periféricos ligeros son:

- "Wearables" (Dispositivos Personales)
- Dispositivos Periféricos en forma de sensores para servicios de seguridad en el hogar
- Balizas/sensores de proximidad
- Dispositivos para redes capilares no celulares (actuadores o sensores)

Debido al bajo costo de los Dispositivos Periféricos ligeros, las tecnologías para la implementación de la seguridad disponibles para estos dispositivos son mínimas. Las tecnologías de seguridad que requieren una cantidad significativa de consumo de corriente, costo o espacio en la placa de circuito no suelen ser apropiadas para estos dispositivos. Sin embargo, los Dispositivos Periféricos ligeros pueden de todas formas utilizar e integrar anclas de confianza pequeñas y rentables para implementar un entorno seguro y robusto.

3.2 El Dispositivo Periférico Complejo

Este modelo de Dispositivo Periférico generalmente tiene una conexión persistente a un servidor de back-end a través de un enlace de comunicaciones tipo WAN, como por ejemplo a través de la red celular (incluyendo redes LPWA) (ver el ejemplo de 'Dispositivo Periférico IoT Config #1' en la Figura 2) o con una conexión local tipo wifi o Ethernet a través de una pasarela de cliente final (consulte el ejemplo de configuración del 'Dispositivo Periférico IoT Config #2' en la Figura 2). El dispositivo puede tener un procesador rudimentario, incluso un microcontrolador de ocho bits, pero es normalmente una unidad de procesamiento más robusta ya que está directamente conectado a una fuente de alimentación de Corriente Alterna (CA) o lleva una batería recargable. Algunos Dispositivos Periféricos complejos se comunican a través de protocolos para redes capilares, pero requieren más capacidad de procesamiento para ejecutar la aplicación local de manera eficiente, como un dispositivo de audio en tiempo real.

Ejemplos de Dispositivos Periféricos complejos son:

- Sistemas de iluminación conectados de IoT
- Electrodomésticos como refrigeradores o lavadoras "inteligentes" y conectados
- Sistemas de control industrial (por ejemplo, SCADA)
- Dispositivos de supervisión y seguimiento "retro-fit OBD2 Celular" para automóviles conectados

Los Dispositivos Periféricos complejos pueden consumir mas corriente y por lo tanto tienen mas potencia, por lo general incluyen procesadores más robustos y tienen más espacio en la placa de circuito impreso disponible para implementar las tecnologías de seguridad. Como resultado, se puede hacer mucho más con los Dispositivos Periféricos complejos. Estos dispositivos pueden usar casi cualquier tipo de Ancla de Confianza. Como resultado, pueden implementar fácilmente una clave personalizada pre-compartida (PSK) o un modelo asimétrico de Base de Computación Confiable (TCB) como se describe más adelante en este documento.

3.3 La Pasarela (o "Hub")

Una pasarela es un dispositivo, generalmente conectado a una fuente de alimentación dedicada, que comúnmente gestiona la comunicación entre los Dispositivos Periféricos ligeros y los sistemas de back-end que los manejan. La pasarela gestiona las comunicaciones WAN de larga distancia, normalmente con tecnologías celulares (incluyendo LPWA), por satélite, par trenzado, fibra o Ethernet. Acepta comandos de los sistemas de back-end que forman parte del Ecosistema de Servicios y los traduce a su vez en mensajes aptos para los Dispositivos Periféricos ligeros.

Si bien la función principal de una pasarela de IoT es enrutar mensajes hacia y desde Dispositivos Periféricos ligeros, también es capaz de realizar tareas críticas, tales como:

- Descubrimiento de Dispositivos
- Despliegue del controlador de red
- Funcionalidad de gestión
- Supervisión en tiempo real

- Autenticación y seguridad utilizando por ejemplo GBA o TLS

Si bien las pasarelas son técnicamente Dispositivos Periféricos, es posible que no necesariamente sean administradas por el usuario final, y pueden ser administradas por el proveedor de servicio de IoT o el operador de red (consulte la información mas adelante). Independientemente de esto, las pasarelas también pueden diseñarse como Dispositivos Periféricos complejos para hacer un uso más eficiente de la distribución de un enlace ascendente a múltiples Dispositivos Periféricos ligeros en una red local.

Al igual que los Dispositivos Periféricos complejos, las pasarelas tienen más capacidad de procesamiento, consumen mas corriente y, por lo general, tienen más espacio disponible en la placa de circuito impreso. Esto permite a las pasarelas de IoT implementar soluciones complejas para una Base de Computación Confiable y tecnologías como clientes de autenticación GBA, con relativa facilidad.

Estos atributos de la pasarela también les permiten incorporar múltiples tecnologías de comunicación para enrutar mensajes entre diferentes tipos de dispositivos de red. Esto permite la comunicación entre Dispositivos Periféricos que normalmente no podrían intercambiar mensajes eficientemente. De esta manera, las pasarelas funcionan como un punto de agregación para los dispositivos dentro del ecosistema local, lo que les permite comunicarse entre sí y, si es necesario, con los Ecosistemas de Red y de Servicios.

Normalmente hay dos tipos de pasarela: una "Pasarela de servicios de IoT" y una "Pasarela de equipos de instalaciones de cliente (CPE)". La diferencia se explica a continuación:

1. El Proveedor de servicios de IoT proporciona una "Pasarela de servicios IoT". Puede ser propiedad del usuario final, pero normalmente es administrado por el proveedor de servicios IoT. Dicha pasarela se utiliza normalmente como un concentrador para conectar Dispositivos Periféricos ligeros al ecosistema de servicios (ya sea directamente a través de una conexión cableada o celular o a través de una puerta de enlace CPE), donde el usuario final compra un servicio administrado por un proveedor de servicios IoT.
2. Un operador de red proporciona una "Pasarela CPE". Este suele ser un router de banda ancha conectado a Internet por redes cableadas o celulares. Esto se puede usar en entornos residenciales o empresariales. En esta configuración, la pasarela generalmente se administra y se configura desde los sistemas del operador de red.

3.4 El Modelo Global

Independientemente del tipo de Dispositivo Periférico que se evalúe o diseñe, todos tienen modelos de subcomponentes similares desde una perspectiva hardware y de logística:

- Una Unidad Central de Procesamiento (CPU) debe ejecutar el código de la aplicación
- La CPU debe cargar/almacenar datos y código ejecutable desde/hacia el almacenamiento permanente
- La CPU debe calcular y procesar los datos en el almacenamiento temporal
- Se debe usar una Base de Computación Confiable para autenticar el entorno
- El dispositivo debe comunicarse con su ecosistema de servicios IoT

Se debe remarcar que los Dispositivos Periféricos ligeros tienen menos capacidad de almacenamiento y computación que los Dispositivos Periféricos complejos o pasarelas. Por lo general, también tienen menos capacidades para implementar soluciones seguras.

El aspecto más importante del modelo general es que cada tipo de dispositivo periférico tiene una función principal: establecer una plataforma fiable, de alta calidad y segura para ejecutar una aplicación en particular. En otras palabras, como ocurre en las plataformas informáticas más complejas como los teléfonos inteligentes (“Smartphones”), servidores en la nube o mainframes, antes de permitir que se ejecute una aplicación avanzada con fiabilidad o de interactuar de forma segura con sus pares, el equipo de ingeniería debe asegurarse de que el hardware sea una plataforma fiable y robusta para las aplicaciones.

Los Dispositivos Periféricos IoT, por naturaleza, son parte de una red de Dispositivos Periféricos. No son dispositivos independientes que realizan una acción sin la influencia o sin el control de un servicio de gestión. Para aumentar la fiabilidad de un dispositivo determinado y disminuir la posible responsabilidad debido a brechas de seguridad o falta de fiabilidad, cada Dispositivo Periférico debe diseñarse con la idea de que la fiabilidad dentro del ecosistema de IoT comienza con la implementación apropiada del hardware del Dispositivo Periférico.

Con esta perspectiva en mente, está claro que incluso el tipo de Dispositivo Periférico más fácil de desarrollar debe comportarse de manera fiable, con una calidad de diseño muy elevada y segura, ya que se espera que participe en una red que podría llegar a millones de dispositivos en el futuro. La manera en que se comporta un dispositivo periférico sin duda tendrá un efecto en todo el ecosistema de IoT en el que funciona. Como resultado, los ingenieros deben considerar las implicaciones del diseño de la arquitectura mucho más allá de los atributos físicos asociados con un dispositivo embebido determinado. Los ingenieros deben pensar en términos de seguridad, fiabilidad y necesidades de calidad de todo el ecosistema de IoT.

4 El Modelo de Seguridad

La seguridad en un Dispositivo Periférico se puede evaluar desde la perspectiva de sus componentes. Al evaluar cada componente que se requiere para construir un Dispositivo Periférico determinado, un ingeniero o un atacante puede llegar a concebir un conjunto de ataques con buenas perspectivas de éxito que darán lugar a un compromiso total del sistema sin un esfuerzo muy grande.

Utilizando el modelo de Dispositivo Periférico global definido anteriormente, los componentes utilizados se pueden evaluar desde una perspectiva general de sistema. Desde esta perspectiva para cada componente, un analista se fijará en las tecnologías que se usan comúnmente y que probablemente no sean las mas adecuadas desde el punto de vista de seguridad. Al priorizar la selección de estos componentes con poca experiencia práctica de diseño, desde la perspectiva del hardware fácilmente disponible y mínimo costo para implementar la funcionalidad requerida, un analista o atacante puede construir un modelo de ataque que pondrá en evidencia rápidamente cualquier Dispositivo Periférico por los fallos de seguridad.

En el Ecosistema de Dispositivos Periféricos, hay varios puntos susceptibles a ataques que serán investigados por los atacantes dependiendo de sus recursos, acceso a la infraestructura y experiencia. Estos puntos son:

- Las redes de comunicaciones utilizadas
- Los servicios de red disponibles
- El acceso a la consola
- Las comunicaciones internas en un Bus de datos
- El acceso físico al “Chip” (Circuito Integrado)

4.1 Ataques a la red de Comunicaciones

El primer paso y el más simple para intentar comprometer un Dispositivo Periférico de IoT generalmente implica el análisis de las deficiencias en el modelo de comunicaciones. Los analistas observarán si el modelo de comunicación incorpora las mejores prácticas de seguridad. Si el analista puede capturar fácilmente las credenciales de inicio de sesión, los tokens de comunicaciones u otros identificadores que el Ecosistema de Servicios utilizará para identificar el Dispositivo Periférico, entonces esto implica que el dispositivo ha sido comprometido.

Esta estrategia puede ser muy fácil o muy difícil dependiendo de las características de las comunicaciones empleadas entre los dispositivos. Especialmente si el analista puede acceder a los datos de las comunicaciones y estos no están codificados y utilizan texto plano. Un atacante suficientemente preparado tendrá muy probablemente tecnología para interceptar las comunicaciones a través de Bluetooth, de la tecnología 802.15.4 y de otros protocolos comerciales. Dado que la observación o la realización de un ataque de intermediario (“Man-in-the-Middle”) en las comunicaciones de un Dispositivo Periférico generalmente requieren poco o ningún cambio en el Dispositivo Periférico, el adversario se encuentra en una posición muy ventajosa. Requiere muy poco esfuerzo y trabajo para implementar este tipo de ataque.

Sin embargo, si el modelo de comunicaciones utiliza las mejores prácticas para hacer cumplir la confidencialidad y la integridad de los datos, el adversario lo tendrá especialmente difícil para acceder a secretos valiosos. Esto hará que el adversario cambie de modelo de ataque por el siguiente modelo que será menos sencillo de implementar.

4.2 Ataques a los Servicios de Red Accesibles

El siguiente paso para atacar un Dispositivo Periférico de IoT es evaluar los servicios de red que estén “abiertos” en el dispositivo. En el primer paso, se capturan los mensajes que salen del Dispositivo Periférico para identificar si hay secretos que puedan utilizarse inmediatamente. Esto permite a un adversario reducir la cantidad de trabajo requerido para extraer secretos del Dispositivo Periférico. Si el modelo de seguridad de las comunicaciones salientes es robusto, los servicios de red se escanean para evaluar si se puede acceder al sistema operativo del Dispositivo Periférico o se puede controlar desde la red de comunicaciones.

Se realizará una evaluación con una herramienta como NMap [8] para determinar si los puertos de red están abiertos. Si la topología de red no tiene capacidad de IP, lo cual es

común en las redes BLE o IEEE 802.15.4, el adversario aún puede usar herramientas de fácil acceso para conectarse al Dispositivo Periférico mediante el protocolo de radio apropiado.

El adversario intentará enviar mensajes al Dispositivo Periférico para determinar si el Dispositivo Periférico se puede manipular para ejecutar comandos o para proporcionar acceso remoto desde la consola del sistema operativo. Un método común es evaluar si está disponible una interfaz de inicio de sesión de red, como Secure Shell (SSH) o Telnet. Si se utilizan credenciales de inicio de sesión predeterminadas, el adversario puede iniciar sesión en el Dispositivo Periférico. Esto permitirá al adversario manipular el sistema operativo local, y potencialmente aprovecharse de las posibles vulnerabilidades locales para escalar privilegios y extraer secretos del dispositivo.

Otro ejemplo común incluye el ataque a servicios web mal diseñados, donde comandos pueden ser inyectados dentro de scripts de la interfaz, “Common Gateway Interface” (CGI), que no eliminan correctamente caracteres de control introducidos en los campos de entrada del usuario, esto da pie a la ejecución de código en el sistema operativo local.

4.3 Ataques sobre el Acceso a la Consola

El acceso a la consola no es exactamente un ataque, es una estrategia. Por lo general, las consolas deben estar habilitadas en Dispositivos Periféricos para proporcionar a los desarrolladores y técnicos de control de calidad (QA) la capacidad de diagnosticar anomalías en el hardware o el software. Sin embargo, la información proporcionada desde una consola es muy valiosa para un atacante. Además, la consola puede proporcionar a un atacante la capacidad de iniciar una sesión en el sistema de Dispositivos Periféricos de forma local y remota.

Por lo general, las consolas de hardware locales se pueden encontrar en los dispositivos Dispositivo Periférico mediante los siguientes métodos:

- Buscando un conector de 5 pines en la placa de circuito impreso que indique la existencia de un puerto serie TTL
- Buscando las especificaciones de la CPU o MCU utilizada en la PCI e identificar los pines de la UART

Se puede usar un multímetro para identificar un puerto TTL, ya que los pines se ajustarán a la especificación de voltaje típica para TTL. Alternativamente, se puede usar un analizador lógico para calcular la velocidad en baudios de cualquier pin de hardware donde se transmitan datos en serie. El analista podrá discernir rápidamente si una consola está disponible en el hardware local.

En muchos casos, el simple hecho de acceder a un puerto de la consola permite que un atacante tenga acceso directo a la línea de comandos provista para el Dispositivo Periférico. En otros casos, se requieren credenciales de inicio de sesión, pero generalmente se pueden adivinar. Si otra persona en Internet ha averiguado las credenciales de inicio de sesión, y todas las credenciales de acceso al Dispositivo Periférico son las mismas para todos los dispositivos existentes, todo lo que el atacante tiene que hacer es realizar una búsqueda en Google para ver si alguien ha publicado dichas credenciales.

El acceso a la consola remota se puede lograr a través de protocolos de red de diagnóstico, protocolos de acceso a la consola (por ejemplo, SSH o telnet) u otros medios. Estas metodologías de acceso deben evaluarse para determinar si un adversario puede manipular el canal de acceso, si esto es así se le otorga acceso a una consola remota.

4.4 Ataque a las Comunicaciones del Bus Local

Si no se puede acceder a la línea de comandos del sistema a través de una consola, el adversario o analista deberá comenzar a inspeccionar el hardware para determinar si el Dispositivo Periférico se puede comprometer fácilmente. Esto puede hacerse de muchas maneras, pero seguro se puede empezar por:

- Observar si hay dispositivos programables en la PCI que se puedan alterar
- Observar si secretos criptográficos se transmiten a través de los buses del hardware sin codificar
- Inyectar mensajes en el hardware que influyan en el comportamiento de la aplicación o sistema operativo a favor del atacante

El ataque más simple es identificar si existen dispositivos programables. El contenido de estos podría ser modificado, como en una tarjeta de memoria externa grabable (SD / MMC). O bien, un chip NVRAM o una EEPROM que se puede modificar haciendo cambios en la aplicación o en la configuración para permitir el acceso a la línea de comandos o a tokens almacenados de forma segura.

Si estos posibles puntos de “entrada” están protegidos correctamente, el atacante determinará si los secretos criptográficos se transmiten a través de los buses hardware. Esto podría necesitar del uso de un analizador lógico para interceptar mensajes entre una EEPROM y la CPU, un microcontrolador y un adaptador de red conectado a la interfaz serie (SPI, “Serial Peripheral Interface”) u a través de otros ataques. Estos pueden ser extremadamente simples y rápidos o muy complejos y costosos, dependiendo de la complejidad del ataque y de la tecnología utilizada en el Dispositivo.

Si el adversario no puede interceptar secretos valiosos utilizando el método anterior, puede intentar inyectar mensajes en los buses hardware para cambiar el comportamiento de una aplicación que se ejecuta en el Dispositivo Periférico. Este es un ataque difícil que requiere mucha experiencia, de equipos relativamente avanzados y la capacidad de evaluar los datos específicos de la aplicación y su contexto.

4.5 Ataques de Acceso Físico al Chip

Si los ataques anteriores son demasiado complejos o costosos, el adversario debe intentar realizar ataques aún más complejos contra el hardware. Esto generalmente implica abusar de la seguridad de algún chip o de los diversos componentes en la PCI. Esto puede hacerse:

- Eliminando el encapsulado del microcontrolador o CPU utilizado para acceder físicamente a pines o buses
- Extrayendo secretos de la EEPROM interna o NVRAM
- Interceptando los mensajes internos de la SRAM
- Haciendo un análisis con rayos X o aplicando ingeniería inversa con FIB

Todos estos ataques requieren un alto grado de conocimientos en ingeniería electrónica y equipos normalmente muy caros. Si bien la mayoría de las empresas no tendrán que temer a un atacante que utilice estas metodologías para realizar ingeniería inversa de sus productos, sigue siendo una posibilidad importante para considerar. La razón es que estos ataques solo se necesitan realizar una sola vez si los Dispositivos Periféricos no están protegidos con secretos criptográficos únicos e individuales.

Si esto es así, un ataque de esta clase extraerá secretos que pueden afectar a toda la línea de productos. Ese es un riesgo importante, porque si los datos se publican por cualquier motivo, la tecnología estará sujeta a ataques y abusos hasta que se instale un parche, si es que se puede crear y desplegar.

5 Preguntas Frecuentes de Seguridad

La seguridad de los Dispositivos Periféricos se presenta en forma de recomendaciones priorizadas en este documento. Pero, para su utilización, es mejor evaluarlas desde un punto de partida práctico. Los ingenieros generalmente comienzan a elaborar una lista de recomendaciones basándose en un objetivo tecnológico o en objetivos influenciados por el negocio empresarial. Esta sección describe los objetivos comunes desde una perspectiva de Dispositivo Periférico y qué recomendaciones son relevantes para lograr esos objetivos.

5.1 ¿Cómo se Combate la Clonación?

La protección de la propiedad intelectual es un objetivo importante para las empresas modernas. El hardware, el firmware y las tecnologías de comunicación utilizadas para construir un Dispositivo Periférico requieren de tiempo para su diseño, experiencia y de presupuesto que las empresas no quieren malgastar, permitiendo que otras empresas o marcas con pocos escrúpulos lo copien fácilmente. Sin embargo, no importa lo que haga una empresa, alguien puede usar exactamente los mismos componentes de hardware para construir un clon o copia muy similar de un producto determinado. No hay nada que la compañía pueda hacer para evitar esto fuera de establecer contratos legales para proteger su tecnología (Propiedad intelectual,) y alianzas empresariales. Sin embargo, existen formas rentables de evitar que alguien use un clon de un producto.

El establecimiento de una autenticación en las comunicaciones de los Dispositivos Periféricos garantizará que cada dispositivo se podrá probar criptográficamente para corroborar que ha sido fabricado por el proveedor de servicios de IoT en cuestión. Siempre que los servicios de back-end, o un Dispositivo Periférico similar, se comunique con otro Dispositivo Periférico, se podrá diferenciar entre un equipo original y un clon, forzando la autenticación entre dispositivos. Si el dispositivo no puede autenticarse, el par o servicio lo rechazará. Esto requiere de las siguientes recomendaciones para funcionar:

- Autenticación de la identidad de un Dispositivo Periférico
- Detección de una autenticación mutua defectuosa o inexistente

5.2 ¿Cómo Protejo la Identidad de un Dispositivo Periférico?

Para autenticar correctamente un Dispositivo Periférico, el ingeniero debe poder confiar en la identidad criptográfica del este. Esto es más complejo de lo que parece, y requiere de una

combinación de procesos, políticas y tecnología para lograrlo. Esto se detalla en la recomendación, "*Implemente una base de computación confiable"*, pero la forma en que se codifican los tokens de autenticación en un Dispositivo Periférico determinará cuán seguro es el sistema en general.

En muchas arquitecturas pensadas para Dispositivos Periféricos, un atacante simplemente puede copiar tokens criptográficos (si los hay) del dispositivo destino para suplantarlo. Si cada Dispositivo Periférico fabricado por el Proveedor de servicios IoT utiliza el mismo conjunto de tokens criptográficos, el adversario puede suplantar cualquier dispositivo simplemente comprometiendo un solo conjunto de tokens.

Por lo tanto, construir la TCB adecuada requiere cumplir con las siguientes recomendaciones:

- Implementar una base de computación confiable (TCB)
- Utilizar un ancla de confianza
- El ancla de confianza tiene que ser resistente a manipulaciones
- Utilizar una API para la TCB
- Usar un generador de números aleatorios contrastado en el mercado
- Utilice un encapsulado/carcasa de producto resistente a la manipulación
- Hacer cumplir la confidencialidad e integridad hacia y desde el ancla de confianza

5.3 ¿Cómo Reduzco el Impacto de un Ataque contra el Ancla de Confianza?

También es importante tener en cuenta que la forma en que se fabrica y aprovisiona un dispositivo tiene un efecto drástico en la seguridad de un Dispositivo Periférico en producción. El proceso de fabricación determinará si los Dispositivos Periféricos están codificados de forma segura con claves. El proceso de gestión de la puesta en marcha y aprovisionamiento determinará cómo se asocia un Dispositivo Periférico con un usuario en particular, y si el dispositivo puede verse comprometido antes o después de que se realice una asociación.

- Aplicar procesos seguros en la cadena de suministro
- Personalizar cada Dispositivo Periférico antes de desplegarlo en campo
- Utilizar una provisión única para cada Dispositivo Periférico
- Implementar funciones para proteger la privacidad y asignar identificadores únicos para cada Dispositivo Periférico

5.4 ¿Cómo Reduzco la Probabilidad de que se Suplante un Dispositivo Periférico?

Después de la clonación de dispositivos por razones comerciales, un ataque deseable desde la perspectiva de un adversario es la suplantación de una persona o un dispositivo en particular. Esto puede o no estar directamente asociado con el ataque de un individuo en particular. Podría ser simplemente la suplantación de un dispositivo con el objetivo de eludir un control de seguridad, como el de un bloqueo digital habilitado para Bluetooth.

Independientemente de la lógica, combatir este ataque se puede lograr mediante el uso de una TCB, personalización, autenticación y también:

- Aplicando el Secreto Perfecto hacia Adelante
- Bloqueando secciones críticas de la memoria

5.5 ¿Cómo Evito la Posibilidad de que se Suplanten Servicios o Dispositivos (Pares)?

Cada red de IoT se compone no solo de Dispositivos Periféricos, sino también de servicios de red y pares en la comunicación. Los Dispositivos Periféricos deben ser autentificados por los servicios, y viceversa. Esto garantiza que los servicios críticos, como las actualizaciones de las aplicaciones, no puedan alterarse para comprometer aún más la red IoT.

- Seguridad de las comunicaciones de Dispositivos Periféricos
- Secreto Perfecto hacia Adelante
- Usar un generador de números aleatorios probado
- Actualizaciones de las aplicaciones vía OTA
- Detección de una autenticación mutua defectuosa o no existente
- Recolección de metadatos no autorizada

5.6 ¿Cómo Evito la Manipulación del Firmware y Software?

Una vez que se ha establecido una raíz de confianza, el Dispositivo Periférico puede autenticarse desde un componente fiable. Hacerlo permite que el Dispositivo Periférico establezca una base de confianza y asegure que la aplicación de la siguiente capa no haya sido alterada involuntariamente (por ejemplo, a través de una NVRAM defectuosa) o intencionalmente por un atacante. Logre esto con:

- Una plataforma de ejecución mínima viable (Aplicación de “Roll-Back”)
- La firma criptográfica de las imágenes de aplicaciones
- El arranque el sistema implementado fuera de la EEPROM interna
- El bloqueo de secciones críticas en la memoria
- El abandono de gestores de arranque inseguros
- La utilización de un encapsulado/carcasa de producto resistente a la manipulación

5.7 ¿Cómo Reduzco la Posibilidad de que se Ejecute Código Remotamente?

Si la alteración del firmware o software físico no arroja los resultados adecuados, el adversario puede pasar a ataques más complejos, como la ejecución de código en el gestor de arranque o en las aplicaciones que se comunican a través de las interfaces de red o de bus. Si todos los pares en la red se autentican, como se describió anteriormente en este capítulo, será mucho más difícil para un adversario inyectar contenido malicioso. Sin embargo, la mayoría de los dispositivos, de una manera o de otra, requieren conectarse a una red de comunicaciones públicas para interactuar con dispositivos de otras organizaciones. Por lo tanto, es posible que no puedan aplicar adecuadamente las restricciones sobre el origen de los datos.

Por lo tanto, la entrada de datos en el sistema informático desde las interfaces tanto remotas como físicas debe ser analizada con mucho cuidado. Para limitar el potencial de un ataque exitoso a una aplicación, y limitar la exposición de sus datos una vez que la aplicación esté comprometida, considere lo siguiente:

- Hacer cumplir la protección de memoria en el diseño del dispositivo
- Usar la memoria interna para guardar los secretos (claves)
- Actualizar la aplicación a través de la interfaz OTA
- Ejecutar aplicaciones con niveles de privilegios apropiados
- Imponer una separación de tareas en la arquitectura de aplicaciones
- Hacer cumplir la seguridad en los lenguajes de programación utilizados
- Hacer cumplir las mejoras de seguridad en el sistema operativo
- Reforzar la Seguridad de la interfaz de usuario
- Auditar el código de terceros

5.8 ¿Cómo Deshabilito la Depuración no Autorizada o la Instrumentación de la Arquitectura?

Un atacante con conocimientos de arquitecturas IoT y acceso a herramientas de depuración intentará utilizar medios estándar de depuración y diagnóstico para obtener acceso a los secretos del sistema, o para alterar o inyectar código en su beneficio. Restringir la capacidad de un adversario para hacer esto disminuirá el potencial de ataques rápidos e “invisibles” que un usuario puede no detectar.

- Utilizar un ancla de confianza resistente a manipulaciones
- Implementar registro y diagnóstico
- Bloquear secciones críticas de la memoria
- Detectar las anomalías
- Utilización de un encapsulado/carcasa de producto resistente a la manipulación
- Deshabilitar las tecnologías de depuración y prueba
- Implementar una interfaz de usuario segura

5.9 ¿Cómo debo Manejar los Ataques de Canal Lateral?

Cuando un atacante por alguna razón no utiliza las opciones típicas para sus ataques, buscará opciones esotéricas para obtener secretos de un dispositivo. Estos ataques evalúan cómo se comporta el hardware para determinar si un patrón en el comportamiento puede equipararse a un valor, como un “1” o un “0”, o una instrucción en particular. Esto, después de un cierto tiempo, le dará al analista la capacidad de aplicar ingeniería inversa a los datos procesados por el sistema embebido.

Además, el atacante puede usar una tecnología muy cara de análisis para extraer secretos del dispositivo, o para construir circuitos extremadamente pequeños que hagan conexiones que “se salten” a nivel de hardware las capas de seguridad utilizadas a nivel del circuito integrado (silicio). Si bien estos ataques son extremadamente difíciles de combatir, hay algunas cosas que el diseñador puede hacer para disuadir los ataques:

- Personalizar cada Dispositivo Periférico antes de la instalación y puesta en marcha
- Usar la memoria interna para guardar las claves/secretos
- Utilizar una carcasa/encapsulado de producto resistente a la manipulación
- Corrompiendo la memoria cuando se hagan ataques a sus interfaces
- Implementar umbrales de bloqueo cuando se detecten ataques en los dispositivos
- Definir umbrales para el consumo de energía en los dispositivos
- Definir los procesos de desconexión y retirada progresiva del mercado (apagado) de los dispositivos
- Desconexión de componentes que se quieran reemplazar por otros fraudulentamente y al detectar puentes hardware no fiables
- Desconexión en un ataque de arranque en frío
- Combatir un ataque FIB (haz de iones)

5.10 ¿Cómo Debo Implementar una Gestión Remota Segura?

La gestión remota es una parte fundamental del ciclo de vida de un Dispositivo Periférico IoT que debe protegerse para garantizar que no se abuse del canal utilizado para su gestión y administración. Esto no es solo un problema con atacantes externos desconocidos. También pueden ocurrir ataques internos, ya sea dentro del círculo del consumidor/usuario o dentro de la organización del Proveedor de Servicios IoT.

- Gestión de contraseñas para el Dispositivo Periférico
- Gestión remota del Dispositivo Periférico
- Implementar registro y diagnóstico de los componentes del servicio
- Utilizar el Secreto Perfecto hacia Adelante
- Utilizar un APN privado

5.11 ¿Cómo Detecto Dispositivos Periféricos Comprometidos?

Dependiendo de la arquitectura del Dispositivo Periférico, puede ser casi imposible determinar si el hardware o el firmware han sido manipulados y si el dispositivo se comporta con normalidad. Sin embargo, un dispositivo comprometido puede detectarse mediante un comportamiento anómalo, siempre que la infraestructura rastree, registre y alerte cuando se detecten acciones sospechosas (comandos, accesos,). Considere las siguientes recomendaciones:

- Implemente la detección de anomalías
- Utilice una carcasa/encapsulado de producto resistente a la manipulación
- Programar umbrales de consumo de energía para generar alertas

5.12 ¿Cómo implemento de Manera Segura un Dispositivo sin una Conexión al Back-end?

Hay ciertos momentos en los que la conexión a un entorno de back-end no está disponible ni se desea. En estos entornos, la seguridad se hace complicada debido a la obvia incapacidad de administrar claves de seguridad, identidades y mecanismos dinámicos de

autenticación. Sin embargo, se puede lograr un nivel adecuado de seguridad. Considere las siguientes recomendaciones:

- Implementar una base de computación confiable
- Definir una raíz organizacional de confianza
- Personalice cada Dispositivo Periférico antes de su puesta en marcha comercial
- Implemente el “Secreto Perfecto hacia adelante”
- Autenticando la identidad de los Dispositivos Periféricos
- Entornos sin conectividad en el back-end

5.13 ¿Cómo Aseguro la privacidad del Consumidor/Usuario?

La privacidad del consumidor es un tema complejo que requiere un análisis en profundidad no solo de la tecnología del Dispositivo Periférico, sino de todo el producto o servicio de IoT. Cada componente en el sistema completo debe analizarse para detectar posibles lagunas en la privacidad. Revise las siguientes recomendaciones para obtener más información sobre la aplicación de seguridad en la privacidad:

- Implementar el Secreto Perfecto hacia Adelante
- Implementar comunicaciones seguras para el Dispositivo Periférico
- Gestionar la privacidad
- Asegurar la privacidad e identidades exclusivas para cada Dispositivo Periférico
- Utilizar un APN privado
- No permitir la recolección de metadatos no autorizada
- Analizar los riesgos de seguridad no obvios (“ver a través de las paredes”)
- Permitir la Interceptación legal

5.14 ¿Cómo Aseguro la Protección de un Usuario Mientras Fuerzo la Privacidad y la Seguridad?

La protección de usuario es un tema que debe considerarse en un contexto concreto con la aplicación, su propósito, los entornos previstos en los que funcionará la aplicación, el tipo de consumidor/usuario y la tecnología de comunicaciones utilizada. A menudo, se debe mantener un equilibrio entre la seguridad y la protección. Esto puede no ser cierto, sin embargo. En cambio, es posible que sea necesario cambiar el modelo de la arquitectura para mantener tanto la seguridad como la protección del usuario. Donde sea posible, la seguridad no debe descartarse a favor de la protección. Ambas deben llevarse a cabo, donde sea posible. Si bien esta es una recomendación filosófica, es importante que el equipo de ingeniería revise constantemente la seguridad. Considere las siguientes recomendaciones para comenzar una discusión sobre la protección de los usuarios en IoT:

- Análisis crítico de la seguridad
- Denegación de servicio intencional e involuntario
- Permitir la Interceptación legal
- Considerar la seguridad en la cadena de suministro

5.15 ¿Qué Problemas no Podría Esperar Resolver?

En cada sistema hay riesgos que no se pueden resolver debido a las leyes de la física, el costo o simplemente la falta de soluciones tecnológicas. Algunos de estos problemas se mencionan aquí:

- Denegación de servicio intencional e involuntario
- Detección y solución a componentes puenteados y puentes cableados no confiables
- Riesgos de seguridad no obvios (ver a través de las paredes)
- Combatir haces de iones y rayos X enfocados
- Considerar la seguridad en la cadena de suministro
- Interceptación legal

6 Recomendaciones Críticas

Al desarrollar un Dispositivo Periférico seguro, siempre se deben implementar las siguientes recomendaciones. Estas definen una arquitectura segura para los Dispositivos Periféricos. Sin estas recomendaciones, el Dispositivo Periférico tendrá un perfil de seguridad incompleto que será fácilmente atacado por un hacker.

6.1 Implementar una Base de Computación Confiable para los Dispositivos Periféricos

El primer paso para proteger cualquier sistema embebido es la definición de una Base de Computación Confiable (TCB). En el contexto de un Dispositivo Periférico (o dispositivos integrados similares), una TCB es un conjunto de hardware, software y protocolos que garantiza la integridad del Dispositivo Periférico, implementa autenticación mutua entre pares en la red de comunicaciones, gestiona las comunicaciones y la seguridad de las aplicaciones.

El núcleo de la TCB es el ancla de confianza, una tecnología de hardware segura que almacena y procesa secretos criptográficos como “Pre Shared Keys” (PSK) o claves asimétricas. Las anclas de confianza, como por ejemplo un UICC, se pueden utilizar para autenticar no solo a los pares durante las comunicaciones de red, sino que se pueden mejorar con vistas a almacenar datos útiles para la seguridad de la aplicación de los Dispositivos Periféricos.

Una vez que se selecciona e integra el ancla de confianza en la solución de un Dispositivo Periférico, se pueden elegir o diseñar bibliotecas SW y HW que integren el ancla de confianza dentro del conjunto global de elementos que conforman una TCB. La TCB permitirá que el sistema operativo y las aplicaciones principales del Dispositivo Periférico administren más fácilmente la seguridad general no solo del dispositivo, sino también de la red.

Es importante que el equipo de ingeniería elija el ancla de confianza correcta para la solución, ya que cada combinación de ancla de confianza y TCB dará como resultado un nivel de seguridad diferente. Algunas combinaciones e implementaciones de ancla de confianza darán lugar a una falsa sensación de seguridad.

Las variaciones más comunes de una Base de Computación Confiable son las siguientes (clasificadas de menos a más seguras):

- No se implementa una TCB (texto sin codificar)
- Clave pre-compartida estática (PSK)
- Clave pública estática
- PSK personalizada
- Clave pública personalizada

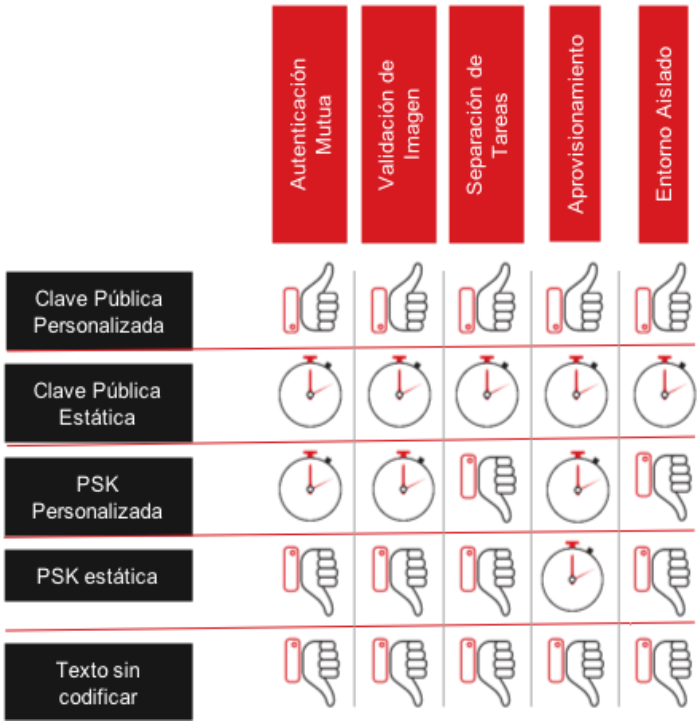

Figura 3 – Garantías de Seguridad para Cada Tipo de TCB.

Observe la figura de arriba. En este diagrama, las capacidades de cada variante de TCB se catalogan por importancia. Un ícono con el pulgar hacia abajo denota que el modelo de TCB no puede cumplir con la estrategia de seguridad representada en la fila superior. Un icono de cronómetro indica que la estrategia de seguridad se puede usar, pero estará sujeta a una ruptura en la seguridad (“hackeo”) después de un tiempo razonable. Un ícono con el pulgar hacia arriba muestra que la estrategia de seguridad se puede implementar de una manera muy fiable y robusta y que la seguridad se preservará probablemente durante mucho tiempo.

Si bien una TCB se puede usar para asegurar muchos aspectos del producto y de los servicios de IoT en general, este documento se centra en cinco conceptos básicos:

- Validación de la imagen de una aplicación ejecutable
- La autenticación mutua de pares en la red
- Separación de tareas dentro de la arquitectura de seguridad de IoT

- Aprovisionamiento y personalización
- Seguridad del entorno totalmente aislado (o seguridad del portal sin conexión)

Una TCB que implementa la *validación de una imagen ejecutable* protege al Dispositivo Periférico verificando criptográficamente cada imagen ejecutable que se deba cargar y ejecutar en el dispositivo. Este proceso comienza en el gestor de arranque, que debe validar criptográficamente la siguiente etapa de ejecución, generalmente nos referimos al núcleo del sistema operativo. El gestor de arranque también podría validar una imagen del sistema operativo o una imagen de la aplicación de firmware almacenada en la NVRAM.

Una TCB que implementa *autenticación mutua* de pares en la red proporciona una raíz de confianza para la autenticación de componentes en la red, y se autentica criptográficamente contra los pares en la red. Esto aumenta la probabilidad de que los pares en la red representen las identidades que dicen representar. Por ejemplo, si el par en la red afirma ofrecer un servicio de actualización de firmware, el TCB autenticaría al par como parte del core de la red del proveedor de servicios de IoT antes de aceptar las actualizaciones de firmware que provengan de ese par.

Una TCB que implementa una *separación de funciones* utiliza una jerarquía de claves para identificar componentes o servicios variables dentro de las ofertas del proveedor de servicios de IoT. Por ejemplo, un conjunto de claves criptográficas podría representar un servicio de actualización de firmware, mientras que un segundo conjunto de claves criptográficas podría representar un servicio tipo "push". Dado que estos servicios tienen funcionalidades completamente dispares, no deben usar las mismas claves e identidades criptográficas para la comunicación. Como tal, la TCB debe administrar y verificar cada identidad para separar un servicio o función de otra. Esto reduce la capacidad de un adversario para comprometer toda la infraestructura del servicio IoT si una de las claves criptográficas se ve comprometida. En otras palabras, si un atacante compromete la clave del "servicio push", no tendrá también la capacidad de hacerse pasar por el servicio de actualización de firmware.

Una TCB que implementa la *personalización y el aprovisionamiento*, asegura que el Dispositivo Periférico tenga una identidad que sea criptográficamente única con respecto a cualquier otro Dispositivo Periférico de su tipo. También asegura que todas las identidades en las comunicaciones estén protegidas para reducir el potencial de filtraciones o pérdidas de privacidad.

Una TCB que implementa la *seguridad en un entorno aislado* impone políticas y procedimientos que validan la autenticidad de los pares y la confidencialidad e integridad de los datos, incluso si no hay un servicio de back-end que ayude en el proceso. En otras palabras, si la comunicación con los servicios de back-end se interrumpe durante un período de tiempo prolongado, el ecosistema IoT aislado aún podrá funcionar con un alto grado de seguridad. Aunque la integridad de los entornos aislados se degrada con el tiempo, una TCB bien diseñada que implemente la seguridad del entorno aislado puede fortalecer la capacidad de recuperación de la red y alargar el tiempo en que el entorno puede considerarse seguro.

En este contexto, la *personalización* implica la existencia de un conjunto único de claves que están asociadas con un ancla de confianza específica. El proceso de *personalización* incluye la generación e instalación de las claves únicas, la asociación de las claves con el circuito integrado (chip) único y la diseminación segura de esta información y los metadatos relevantes a las autoridades correspondientes. Esto asegura que cada chip tenga una identidad criptográfica única. *Estático*, en este contexto, se refiere a un mismo conjunto de claves utilizado para cada Dispositivo Periférico.

Si bien las TCB se pueden utilizar para resolver casi cualquier problema de seguridad que pueda tener un sistema embebido, existen varios problemas fundamentales en un sistema que una TCB debe ser capaz de resolver:

- Validación de imágenes de aplicaciones relativas a Dispositivos Periféricos
- Autenticación de red y/o autenticación entre pares
- Una separación de funciones
- Aprovisionamiento y personalización
- Aprovisionamiento y comunicación en un entorno aislado (sin conexión)
- Aleatorización

Si bien es obvio que elegir no implementar una TCB tiene como consecuencia una falta de seguridad, existen sutilezas en las variadas implementaciones comunes de una TCB que deberían abordarse. Si no se analizan estas sutilezas, pueden dar lugar a importantes lagunas en la seguridad en la implementación final.

6.1.1 Modelos para las Claves de un Ancla de Confianza

6.1.1.1 Claves Estáticas

Una implementación de clave estática ya sea de tipo PSK o claves asimétricas, se define como una solución donde cada Dispositivo Periférico utiliza el mismo secreto criptográfico para resolver un problema determinado. Si bien se pueden usar diferentes claves para resolver distintos problemas de seguridad para un sistema de IoT, la clave sigue siendo la misma para cada Dispositivo Periférico.

Este modelo parece seguro porque cada problema de seguridad puede resolverse a nivel de la TCB de una manera efectiva. Sin embargo, la vida útil de la solución global puede variar mucho. Dependiendo de la seguridad del ancla de confianza, de la elección del algoritmo criptográfico y del tamaño de clave, los atacantes podrían “hackear” la solución casi de inmediato.

El problema realmente surge porque un compromiso único de la clave expone a todos los Dispositivos Periféricos a un compromiso dentro de un sistema. Esto devalúa la implementación de la TCB en concreto y de alguna manera malgasta el tiempo y el dinero utilizados para implementar la solución de seguridad para el Dispositivo Periférico y la arquitectura general del sistema IoT implementado. Por lo tanto, este tipo de modelo en una TCB pone efectivamente en peligro la seguridad de un sistema, como si fuera una bomba de relojería.

6.1.1.2 Claves Personalizadas

Independientemente de si se implementa una solución PSK o asimétrica, la personalización es imprescindible para que una TCB funcione de manera efectiva. La personalización deshabilita la capacidad de un atacante para utilizar un ancla de confianza comprometida para anular la seguridad de todo el ecosistema de IoT. Si un adversario solo puede comprometer un único Dispositivo Periférico a la vez, y requiere de un acceso físico para hacerlo, será extremadamente lento, costoso y complejo implementar un compromiso importante dentro del sistema en cuestión de IoT. Esto representa una victoria significativa para el negocio.

Debido a que los estándares en las comunicaciones celulares han evolucionado en las últimas décadas de manera importante, los operadores de red han perfeccionado el modelo PSK para personalizar las anclas de confianza, como se hace dentro de un UICC. Como resultado, un UICC si es preciso, podría aprovisionarse para servir como un ancla de confianza para una aplicación IoT dentro de un Dispositivo Periférico, esto representa y ayuda a implementar una solución rentable para las aplicaciones de IoT. En un futuro no muy lejano, cuando un eUICC esté disponible en una solución IoT, se podría habilitar como un ancla de confianza, incluso si estuviera ya desplegado comercialmente.

Hoy, la tecnología de claves personalizadas es la solución de seguridad más efectiva para un ancla de confianza. Las TCB implementadas hoy en día en IoT deberían basarse en una solución de TCB personalizada. Los proveedores de servicios de IoT deberían hablar con su operador de red para determinar si una UICC o una SIM podrían implementarse como un ancla de confianza en la capa de aplicación.

6.1.2 Protocolos y Tecnologías de una TCB

Junto con un ancla de confianza, la TCB debe incorporar protocolos, políticas y bibliotecas de software para brindar seguridad al producto o servicio de IoT globalmente. Una ventaja en la utilización de anclas de confianza estándar utilizadas por las tecnologías de red celulares, es la posibilidad de introducir en las anclas, software de personalización y aprovisionamiento que ya existe dentro de las librerías de los Operadores de Red. Las tecnologías, los protocolos y las “suites”, como los que se muestran a continuación, pueden utilizarse en una TCB para autenticar un Dispositivo Periférico en la red:

- La aplicación oneM2M SM UICC, especificada en oneM2M TS-0003
- La “Generic Bootstrapping Architecture” (GBA) 3GPP TS 33.220 (Ver Anexo A)

El uso de estas tecnologías ayudará a acelerar la implementación de aprovisionamiento y personalización, ya que las bibliotecas y protocolos han sido investigados por ingenieros experimentados y analistas de seguridad durante muchos años. Sin embargo, es posible que estos protocolos no permitan que la TCB valide por completo la aplicación del Dispositivo Periférico, o que el Dispositivo Periférico pueda autenticar correctamente los mensajes o autorizar comandos. La TCB debe incorporar otros protocolos para realizar estas tareas, como la validación del firmware, la validación de mensajes de actualización inalámbrica y más.

En un futuro cercano, una tecnología como la de los eUICC mejorará las capacidades desde la perspectiva de la aplicación, y las UICC proactivas permitirán el uso de dos funcionalidades a la vez, una para arrancar el Dispositivo Periférico en sí, al tiempo que administra la seguridad en la red donde se conecta dicho dispositivo. Este es una mejora importante ya que los operadores de red pueden administrar de manera remota y segura el dispositivo eUICC en nombre del proveedor de servicios IoT. Además, la funcionalidad del manejo confidencial de la información de la tarjeta ("Confidential Card Content Management") especificada en la especificación de la tarjeta GlobalPlatform[15] permite que varios actores en los ecosistemas de servicios IoT administren su propia aplicación de forma independiente, si el operador de red lo permite.

6.1.3 Riesgo

No implementar una TCB, es una elección crítica que puede perjudicar a toda la arquitectura de un sistema de IoT. Sin una TCB bien definida, la interacción entre el ancla de confianza y la aplicación en el back-end se definirá de forma vaga, y puede tener lagunas que pueden ser "hackeadas" por atacantes. La TCB asegura que las comunicaciones entre el ancla de confianza, la aplicación principal y los pares en la red sean seguras, fiables y estén bien actualizadas. Sin una TCB, no hay un componente central que pueda administrar el ciclo de vida de seguridad en un Dispositivo Periférico.

6.2 Usar un Ancla de Confianza

Para que un Dispositivo Periférico pueda participar en un ecosistema, debe poder verificar la integridad de su propia plataforma y debe poder autenticar la identidad de sus pares. Para hacer esto, los Dispositivos Periféricos requieren de un ancla de confianza integrada dentro de una Base de Computación Confiable.

Un ancla de confianza es un elemento de hardware seguro, ya sea un chip físico independiente o un núcleo seguro dentro de una CPU, que es capaz de almacenar y procesar secretos criptográficos de forma segura. Un dispositivo UICC o eUICC es un ejemplo de una tecnología segura que se puede usar como un elemento de confianza para almacenar secretos de autenticación.

Usar un elemento de confianza efectivamente implica almacenar, verificar, actualizar y procesar datos. Los datos pueden ser información secreta o pública que debe verificarse criptográficamente. En cualquier caso, el ancla de confianza debe poder determinar con seguridad si los mensajes y las identidades pueden ser autenticados, y debe poder decirle a la TCB de forma segura el resultado de todas las operaciones de autenticación o criptografía. Esto permite que la aplicación y la TCB tomen decisiones importantes que afectarán a la seguridad en todo el Dispositivo Periférico. Por ejemplo, un ancla de confianza puede ayudar a un Dispositivo Periférico a determinar si un par en la red se está haciendo pasar por un recurso crítico, como un servidor de implementación de parches. Si el ancla de confianza no puede validar un par en la red, la TCB y la aplicación en el Dispositivo Periférico deben optar por no interactuar con dicho par, y deben alertar al usuario, cuando sea posible, sobre el recurso de red fraudulento.

Gracias a la disminución en el costo de los componentes y un fuerte aumento en la demanda, las anclas de confianza se pueden encontrar con relativa facilidad

comercialmente. Esto no solo incluye la tecnología HW en sí del ancla de confianza, sino también las bibliotecas SW e interfaces aprobadas para su uso con dicha solución para el ancla. Esto permite que el equipo de ingeniería genere una solución de ancla de confianza en muy poco tiempo, y ayudará a asegurar que la longevidad de la tecnología no se vea debilitada por un software personalizado o estándares mal implementados. Siempre que sea posible, los estándares existentes deben usarse para disminuir el potencial de lagunas en la seguridad.

Otro desafío en la implementación de un ancla de confianza en Dispositivos Periféricos ligeros es el tamaño del componente. Si se utiliza un ancla de confianza externa, será necesario utilizar un encapsulado los más pequeño posible para dicho componente. Lograr esto es difícil cuando el formato físico incorpora una tecnología como la de un UICC. Sin embargo, la norma ETSI TS 102 671 resuelve este problema al introducir un formato muy pequeño para el encapsulado de aproximadamente 6 milímetros por 5 milímetros. Los encapsulados con los formatos "MFF1" y "MFF2" para una tarjeta inteligente de tipo UICC permiten poder utilizar todas las funcionalidades incluidas en una UICC a la vez que garantizan que los requisitos para su integración sean mínimos con respecto al tamaño. La seguridad se fortalece mediante la utilización de un formato físico (chip) que se puede aprovisionar en campo y que se suelda en el dispositivo (PCI), lo que hace que sea más difícil para los atacantes transferir la identidad de un dispositivo a otro dispositivo.

Los gastos incurridos en el desarrollo y despliegue de un ancla de confianza pueden incluir:

- El costo de la tecnología de base, ya sea integrada en la CPU o en un chip separado
- El costo de integrar la tecnología en el circuito, donde sea necesario
- El costo de diseñar o integrar el controlador en el sistema operativo y en la TCB
- El costo de diseñar la Aplicación para utilizar el ancla de confianza
- La gestión del ancla de confianza, para:
    - El mantenimiento de claves de seguridad, revocación de claves y desactivación de identidades
    - El mantenimiento de la infraestructura requerida para asegurar y administrar las claves y los metadatos
- Monitoreo de la identidad del ancla de confianza en el lado del servicio
    - Implementación de una “lista negra” de dispositivos, cuando sea necesario
- Integración de los servicios de operador, cuando sea posible, para supervisar y administrar las anclas de confianza como en un UICC

6.2.1 Riesgo

Los riesgos de no utilizar un ancla de confianza son muchos, pero todos parten del mismo problema de base: la capacidad de un atacante de robar claves relevantes para todo el ecosistema de IoT. El resultado de esto es que un adversario puede:

- Clonar identidades de Dispositivos Periféricos
- Suplantar los servicios de IoT
- Implementar parches o actualizaciones no autorizados
- Realizar cambios no autorizados en el software del Dispositivo Periférico

Estas lagunas en la seguridad pueden dar como resultado problemas costosos para la empresa a lo largo del tiempo, y pueden permitir que no solo los atacantes, sino también los competidores, abusen de la infraestructura IoT en su beneficio.

6.3 Usar un Ancla de Seguridad contra Manipulaciones Físicas

Algunas Anclas de Seguridad tienen seguridad física adicional para protegerse contra cierta clase de ataques, como a través de FIB, análisis de canales laterales y “glitching” (comportamiento inesperado). Mientras que algunos ataques, como la utilización de FIB, son casi imposibles de evitar desde una perspectiva de costos, la fabricación del ancla de confianza puede utilizar tecnologías modernas para hacer que los ataques sean más costosos. Cuanto más costoso es un ataque, es menos probable que se use aleatoriamente contra Dispositivos Periféricos. En cambio, los ataques se centrarán en objetivos en los que el gasto valga la pena.

En un futuro próximo, algunos fabricantes de anclas de confianza están planeando implementar variaciones de su tecnología conforme a los estándares federales de procesamiento de la información (FIPS) [10], EMVCo [11] y Criterios Comunes ya aprobados. Los ingenieros que desarrollen una nueva tecnología deberían determinar si sus diseños actuales respaldarán el cambio a módulos o circuitos compatibles con las normas al uso en un futuro cercano.

Para obtener más información, revise la última versión de cada estándar para evaluar qué capacidades ofrece su fabricante. Tenga en cuenta que algunos niveles de seguridad son intencionalmente casi imposibles de implementar para los dispositivos de consumo debido al costo y la complejidad de las implementaciones.

6.3.1 Riesgo

El riesgo de no utilizar un ancla de confianza resistente a la manipulación es extremadamente alto. Por ejemplo, si un ancla de confianza se construye simplemente con claves criptográficas almacenadas en una NVRAM, cualquier atacante con las herramientas y la destreza para extraer esas claves, puede potencialmente alterar toda la infraestructura. Sin embargo, si los secretos se almacenan en un ancla de confianza a prueba de manipulaciones, el gasto para extraer los secretos es alto, lo que hará menos probable que se extraigan los secretos, haciendo que el ancla de confianza en cuestión no sea considerada como un posible blanco para el atacante.

Es importante darse cuenta de que, si la implementación del ancla de confianza es deficiente desde el punto de vista de seguridad, la probabilidad de que la extracción de secretos que resulte en un compromiso puede ser suficientemente alta. Cualquier compromiso hará inútiles los gastos incurridos durante el proceso de diseño, de definición de la arquitectura, de la producción y puesta en marcha comercial. Esto puede resultar en gastos importantes irrecuperables. Por lo tanto, es imperativo garantizar que la empresa haya diseñado una implementación correcta.

6.4 Utilizar una API para Acceder a la TCB

Una vez que se ha establecido una raíz de confianza dentro de la TCB, se debe utilizar un protocolo que incorpore las capacidades de la TCB y de la raíz de confianza de manera efectiva. La API debe garantizar que:

- Toda verificación de firma la realiza la TCB
- No hay claves privadas expuestas desde la TCB
- La TCB puede realizar el intercambio de claves en nombre de la aplicación
- El descifrado puede ser realizado por la TCB
- El cifrado se puede realizar en la TCB
- La firma de mensajes se puede realizar en la TCB
- El relleno seguro de mensajes se puede hacer con la TCB
- Se aplique la confidencialidad e integridad entre la TCB y la aplicación

Este conjunto de capacidades ayudará a garantizar que la TCB nunca exponga activos de seguridad críticos a una aplicación insegura o entorno hardware. Esto se puede lograr utilizando una especificación existente que aplique estos requisitos de manera uniforme. Considere evaluar las siguientes especificaciones:

- SIM Alliance Open Mobile API [12]
- GlobalPlatform Secure Element Access Control [13]
- Especificación de la API del Entorno de ejecución de confianza (TEE) de GlobalPlatform [14]
- Trusted Computing Group (TCG)
- oneM2M TS-0003 [20]

Muchas anclas de confianza vendrán con bibliotecas de software que se pueden implementar como una TCB. Estas bibliotecas tendrán unas APIs que los ingenieros pueden usar para interactuar con la TCB. Las bibliotecas proporcionadas con el ancla de confianza deben ser las preferidas, cuando estén disponibles, ya que es probable que hayan sido analizadas por expertos en el campo del desarrollo de anclas de confianza. Sin embargo, el equipo de ingeniería debe evaluar la lista de requisitos establecidos en esta recomendación, y debe asegurarse de que la biblioteca tenga en cuenta adecuadamente estas cuestiones.

Además, las TCB solo deberían ser accesibles desde aplicaciones con los privilegios adecuados que se ejecutan en el Dispositivo Periférico. Nunca se debe poder acceder a la interfaz de la TCB desde una aplicación sin privilegios o no fiable (de terceros) que se ejecute en un Dispositivo Periférico. Todo el acceso se debe realizar a través de un servicio fiable que evalúa las solicitudes y, opcionalmente, alerta al usuario cuando las aplicaciones no fiables realizan solicitudes sospechosas o centradas en la privacidad exclusivamente.

El desafío en la implementación de este protocolo reside en garantizar que no se puedan alterar todos los mensajes entre el punto de origen de los datos y la TCB, y viceversa. Es más efectivo si un segmento de la EEPROM se puede invocar desde la aplicación, para realizar estas funciones en nombre de la aplicación. Al aislar la parte medular del código de la API en la EEPROM interna y al usar la RAM interna para procesar los mensajes, se expondrán menos datos críticos en los buses externos.

6.4.1 Riesgo

Si una interfaz del protocolo de la aplicación no está bien definida, el uso de una TCB puede tener resultados no deseados o efectos secundarios. Al definir el protocolo de manera preliminar y examinarlo en busca de problemas de lógica y de seguridad, el equipo de ingeniería puede identificar fallos de manera más rápida y efectiva que pueden ocasionar problemas de seguridad más adelante. Por lo tanto, la definición del protocolo debe incorporar la evaluación de las APIs existentes que incluyen las necesidades del proveedor de servicios IoT. Si se puede identificar una tecnología existente y bien establecida, siempre será mejor que una solución personalizada.

6.5 Definir una Raíz de Confianza Organizativa

Una raíz de confianza organizativa es un conjunto de políticas y procedimientos criptográficos que rigen cómo las identidades, aplicaciones y comunicaciones pueden y deben protegerse criptográficamente. Se debe usar una criptografía robusta, ya sea en forma de claves simétricas únicas, certificados o claves públicas. Esto depende del modelo disponible para su uso en la TCB, las capacidades del ancla de confianza y lo que tiene sentido para el equipo de ingeniería.

Se debe usar una clave raíz privada, ya sea simétrica o asimétrica, para firmar digitalmente otras claves utilizadas en la jerarquía. Por ejemplo, si nuestra organización ejemplo, “IoT Company LLC”, quiere crear una raíz de confianza organizativa, generarían una clave raíz en una máquina fiable. Esta clave representará la raíz de la organización. Luego generarían nuevas claves que representarían a cada sub-organización que debería tener jerarquías de seguridad independientes. Los ejemplos pueden ser:

- Clave de firma de código
- Clave de comunicaciones del servidor
- Clave de comunicaciones punto a punto
- Clave de identidad de dispositivo Periférico
- Clave de revocación maestra

Cada una de estas claves debe estar firmada por la clave raíz de la organización. Todas estas claves, su firma correspondiente y la clave raíz deben almacenarse en el ancla de confianza utilizada por la TCB. Luego, cada vez que se utiliza la aplicación vinculada a una clave en particular, la aplicación puede usar las claves específicas para validar los mensajes enviados a través de los canales de comunicación.

Este modelo ayuda a garantizar que todos los mensajes estén protegidos a través de la jerarquía criptográfica. Al separar los deberes entre los tipos de clave específicos, las claves comprometidas se pueden revocar a través del mismo proceso de comunicación.

Algunos protocolos existentes que ayudan a implementar este método son:

- Transport Layer Security (TLS); La última especificación válida
- Secure Shell (SSH2)
- Protocolo de estado de certificado en línea (OCSP) IETF RFC 2560
- Arquitectura Genérica de Arranque (GBA) (Ver Anexo A) 3GPP TS 33.220

La dificultad surge cuando los servicios que necesitan las claves criptográficas deben ser desplegados. En lugar de colocar un activo crítico para la seguridad, como la clave de comunicaciones del servidor, en un servidor web accesible a través de Internet, se debe generar un certificado o par de claves específico para esa capa en el servidor. Entonces, este certificado puede estar firmado por la clave de comunicaciones del servidor. De esta forma, cualquier Dispositivo Periférico puede verificar que el servicio haya sido autenticado por la raíz de confianza, pero la clave crítica de la organización no estará expuesta a atacantes.

Si alguna vez se compromete una clave, se puede revocar mediante el uso de la clave maestra de revocación para autenticar la revocación.

No hace falta decir que todas las claves principales en la raíz de confianza de la organización son fundamentales para la seguridad de la infraestructura. Estas claves deben estar muy protegidas, y solo deben ser utilizadas por miembros internos de confianza del equipo principal responsable. Se recomienda encarecidamente utilizar un Módulo de seguridad de hardware (HSM) aprobado para almacenar, acceder y usar las claves.

Mientras que un HSM a menudo puede representar un gasto significativo al comienzo de la implementación de una tecnología, los efectos financieros a largo plazo son muy positivos. En lugar de incurrir en un gasto significativo más adelante para el análisis e ingeniería forense para diagnosticar y combatir un riesgo particular que podría haber sido resuelto por una TCB y un HSM, que al implementarlos de inicio representan un gasto inicial relativamente pequeño comparativamente.

6.5.1 Riesgo

El riesgo de no usar una raíz de confianza organizativa es que cualquier compromiso con una sola clave puede resultar en un compromiso para todo el ecosistema. Al separar la organización en una jerarquía y desplegar claves separadas para cada capa en la jerarquía, las claves se pueden reciclar a intervalos regulares y de acuerdo con la prioridad de la aplicación o sub-organización con la que se relaciona la clave. Esto crea una separación de funciones entre las capas de la organización y disminuye la capacidad de una clave comprometida para dañar la seguridad de toda la infraestructura.

6.6 Personalizar cada Dispositivo Periférico antes de Ponerlo en Marcha Comercialmente

Los Dispositivos Periféricos deben estar habilitados con identidades criptográficas únicas para garantizar que los adversarios, competidores y aficionados no puedan hacerse pasar por otros usuarios o dispositivos en entornos de producción. Para lograr esto de manera adecuada, el proceso de personalización debe realizarse en la fabricación. Esto puede hacerse a través del fabricante de la solución TCB particular o durante el proceso del ensamblaje de la PCI.

Para resolver el proceso de personalización, considere lo siguiente:

- Generar una clave criptográfica única

- Firmar esta clave utilizando la clave organizativa para firma del Dispositivo Periférico (o una derivada de esta última)
- Almacenar la clave en el ancla de confianza de la TCB
- Generar (usar) un identificador interno único para ese Dispositivo Periférico específico
- Almacenar el identificador único en el ancla de confianza de la TCB
- Guardar el identificador único, la clave y la firma en el sistema de autenticación del back-end del servicio IoT

Tenga en cuenta que la personalización de la plataforma del Dispositivo Periférico es independiente de la personalización de la identidad de la red. El uso de un UICC para la autenticación de red es beneficioso por muchas razones, y cuando sea posible, el UICC podría usarse también como un ancla de confianza. Sin embargo, si el ancla de confianza de la red solo puede utilizarse para la autenticación de la red, la personalización del ancla de confianza de la aplicación debe realizarse por separado. Se requiere la unicidad criptográfica del ancla de confianza de la aplicación para garantizar que la plataforma de la aplicación se verifique antes de la ejecución de la aplicación en el Dispositivo Periférico.

Llegando a un acuerdo específico con el operador de red u otra parte emisora, los UICC a veces pueden aprovisionarse antes de la entrega para servir como un ancla de confianza centrada en la aplicación. En un futuro cercano, los desarrolladores de Dispositivos Periféricos deben evaluar si la tecnología eUICC es adecuada para su utilización en productos y servicios de IoT. Estas tecnologías permitirán el aprovisionamiento “en el terreno” de secretos criptográficos de forma similar a un ancla de confianza centrada en la aplicación. Dado que la industria móvil celular es líder en el proceso de personalización y aprovisionamiento, puede haber una ventaja significativa al usar un eUICC como un ancla de confianza.

Además, estas tecnologías incorporarán capacidades de aprovisionamiento remoto y canales seguros para la comunicación segura entre la aplicación y el ancla de confianza implementada en un eUICC. Estas capacidades proporcionarán personalización sobre el terreno, lo que reducirá el costo general de Personalización y Aprovisionamiento para cada Dispositivo Periférico de manera individual.

En el Anexo B se incluye un breve tutorial sobre el uso de tarjetas UICC en un ecosistema de servicios de IoT.

El desafío consiste en administrar las identidades de Dispositivos Periféricos y el proceso de firma. Cada identidad debe estar catalogada, junto con identificadores únicos que coincidan con la identidad, en un sistema que no se pueda alterar. Si bien el proceso generalmente se realiza durante el montaje de las PCI en fabrica, se debe establecer una conexión desde esa instalación a la empresa para transferir de manera segura los datos de identidad.

Desarrollar esta solución puede ser sencillo con algunas instalaciones/fábricas que están más familiarizadas con la personalización criptográfica. Es posible que otras instalaciones de fabricación no cuenten con un proceso para lograr esto. La industria de tecnologías celulares ha tenido éxito de esta manera debido a su capacidad para controlar la fabricación y la puesta en marcha comercial de tecnologías de sistemas embebidos y circuitos

integrados como un UICC. Si bien la industria de la telefonía celular ha sido un líder en este proceso durante algún tiempo, el proceso de personalización y provisión de una aplicación IoT en un Dispositivo Periférico aún está “en pañales”.

Esté preparado para determinar si la identidad del Dispositivo Periférico debe (o podría) ser administrada por una pasarela o un enlace ascendente. La evaluación de la arquitectura del producto o servicio de IoT ayudará a determinar si este atributo de administración de identidad afectará al proceso de personalización. Si bien la confianza se puede distribuir a las pasarelas, la organización debe determinar si la confianza se puede delegar adecuadamente sin disminuir la seguridad general del sistema de comunicaciones y autenticación.

Los gastos relacionados con la personalización suelen incluir, entre otros:

- La implementación del proceso de personalización en el fabricante del circuito integrado
- La coordinación o entrega de los valores personalizados únicos tanto al fabricante como al proveedor de servicios IoT
- Implementación y gestión de las identidades personalizadas

6.6.1 Riesgo

Si la organización decide no implementar la personalización de los Dispositivos Periféricos, corre el riesgo de no poder diferenciar un Dispositivo Periférico de otro. Si todas las claves se conforman en los sistemas de Dispositivos Periféricos, no importa si los números de serie son únicos. La razón de esto es que, si las claves se pueden extraer de cualquier Dispositivo Periférico individualmente, el atacante sería capaz de hacerse pasar por cualquier Dispositivo Periférico del ecosistema.

La personalización combate esto forzando al atacante a extraer los secretos criptográficos de cada Dispositivo Periférico que quiere clonar o suplantar. Debido a que el costo de este proceso puede ser muy alto, la personalización utilizando un ancla de confianza es el método más robusto para combatir la clonación y la suplantación.

6.7 Implementar una Plataforma de Ejecución Mínima Viable (Recuperación de la Aplicación)

Una plataforma mínima de ejecución viable (MVeP) representa la cantidad mínima de trabajo que se requiere para crear un entorno de ejecución fiable para comunicarse con el ancla de confianza. Típicamente, esto significa:

- Configurar el reloj interno o el oscilador
- Configurar los periféricos del core (memoria, almacenamiento)
- Habilitación de varios puentes a nivel de hardware o dispositivos periféricos
- Autenticación del siguiente fragmento de código que va a ser ejecutado por la CPU
- Ejecución de la siguiente etapa de código
- Gestionar la recuperación de la imagen de la aplicación

Una vez que se ha definido este MVeP, el gestor de arranque mínimo puede usar el ancla de confianza para verificar un gestor de arranque más robusto, o puede ejecutar el resto del gestor de arranque después de verificar las aplicaciones externas. Esto permite definir un entorno coherente con un esfuerzo mínimo que autentica cadenas de código en serie que definirán la plataforma de aplicaciones.

Otro beneficio es que con el uso del modelo MVeP, incluso los procesadores con una cantidad mínima de NVRAM interna o EEPROM pueden arrancar una arquitectura confiable usando un ancla de confianza interno o externo.

Por último, un MVeP es importante para retroceder a las versiones estables de una plataforma en particular. Si se puede definir un MVeP que tenga la funcionalidad mínima requerida para verificar la integridad de las imágenes de firmware de la aplicación y configurar el entorno de ejecución, su funcionalidad puede separarse de la funcionalidad de la aplicación principal. Por lo tanto, si falla una actualización de firmware por alguna razón, un MVeP aún se puede usar para volver a conectarse a la red del back-end y descargar otra imagen de firmware (ya sea la misma imagen o una imagen anterior). Esto también permite que los Dispositivos Periféricos con chips NVRAM dañados todavía se comuniquen con los servicios de back-end y envíen información de diagnóstico.

6.7.1 Riesgo

Si bien puede parecer beneficioso, definir un MVeP asegura que la arquitectura del Dispositivo Periférico en general verifique criptográficamente cada paso del proceso de arranque. Este paso es fundamental para garantizar que un Dispositivo Periférico pueda autenticarse en la red y autenticar a sus pares. Si el MVeP tiene una arquitectura deficiente, puede dar lugar a lagunas de seguridad en el proceso de arranque que pueden ser utilizadas para el beneficio de los atacantes, lo que invalida la arquitectura de seguridad.

6.8 Provisionar cada Dispositivo Periférico de Manera Única

Si bien la personalización garantiza que cada dispositivo sea único una vez que se haya fabricado, el aprovisionamiento asegura que un dispositivo único se active, actualice y asocie con una identidad de cliente en particular. El proceso de aprovisionamiento ayuda a distinguir los dispositivos que se han fabricado de los dispositivos que se han comprado y/o han desplegado en un entorno IoT. Esto ayuda al proveedor de servicios IoT a:

- Distinguir entre dispositivos activos y desactivados
- Asociar Dispositivos Periféricos con redes u otros recursos vinculados con un cliente en particular
- Personalizar un Dispositivo Periférico de acuerdo a las necesidades del cliente
- Determinar más fácilmente si un cliente en particular o Dispositivo Periférico se ha visto comprometido

El proceso de aprovisionamiento no ocurre durante la fabricación, sino que se fundamenta en el proceso de personalización implementado en la fabricación. El aprovisionamiento generalmente ocurre sobre el terreno, dependiendo del cliente que inicializa el proceso de activación. Sin embargo, para que el proceso sea seguro, el aprovisionamiento depende de los tokens de seguridad únicos establecidos durante el proceso de personalización para garantizar que un Dispositivo Periférico en concreto esté vinculado a un cliente único. De

esta forma, un adversario no podrá registrar (o anular el) registro de Dispositivos Periféricos simplemente adivinando algunos parámetros del Dispositivo Periférico. En cambio, requerirían de cada token criptográfico único generado y establecido durante el proceso de personalización, esto es computacionalmente inviable.

De esta manera, el proveedor de servicios de IoT puede garantizar matemáticamente que es improbable que los adversarios puedan burlar o registrar arbitrariamente Dispositivos Periféricos cuando quieran. Esto conduce a un entorno de IoT más seguro y estable, donde la relación entre los clientes y los dispositivos puede ser más fiable.

6.8.1 Riesgo

No implementar el proceso de aprovisionamiento puede provocar una desincronización entre los servicios de IoT de una empresa y sus Dispositivos Periféricos. Será más difícil para la empresa monitorizar los Dispositivos Periféricos y establecer qué dispositivos se han habilitado en el ecosistema y cuales se han deshabilitado. Además, puede ser difícil establecer qué dispositivos están asociados con clientes particulares, lo que aumentará la dificultad de supervisar un dispositivo problemático o potencialmente comprometido “en el terreno”.

6.9 Gestionar las Contraseñas en los Dispositivos Periféricos

Los dispositivos que incorporan interfaces de usuario deben ser capaces de administrar las contraseñas de manera eficaz. Esto requiere de varias cosas:

- Reducir del riesgo de un ataque de fuerza bruta
- Deshabilitar el uso de contraseñas predeterminadas o precodificadas/definidas
- Aplicación de las mejores prácticas para las contraseñas
- No permitir la visualización de las credenciales de usuario en las interfaces de inicio de sesión
- Imposición de umbrales y retrasos incrementales para intentos de inserción de contraseña no válidos

Los usuarios deberán estar protegidos contra uno de los ataques más simples: otro usuario intentando adivinar su contraseña. Esto se puede solventar simplemente no permitiendo un ataque de fuerza bruta. Esto se puede hacer aumentando el intervalo de tiempo entre intentos de inserción de contraseñas. Con cada intento fallido de inicio de sesión, se debe incrementar el tiempo de espera antes de que se pueda ingresar de nuevo una contraseña. Debería implementarse un umbral máximo de modo que se pueda intentar sólo hasta N veces como máximo. De lo contrario, se debe aplicar un período de bloqueo razonable en la interfaz. El usuario debe ser notificado que ha habido intentos de adivinar su contraseña utilizando fuerza bruta muy probablemente, una vez que se ingresen las credenciales reales.

Las contraseñas grabadas previamente (en algún tipo de memoria) o predeterminadas nunca se deben usar en los sistemas de IoT. Nunca debe haber una contraseña para el administrador de sistema como "puerta trasera" para entrar en el sistema. Nunca debe haber una cuenta privilegiada con credenciales predeterminadas. Esto es esencial para

proteger los dispositivos de aquellos “hackers” que de manera aleatoria intentan detectar dispositivos con una seguridad deficiente.

Las contraseñas deben cumplir con los requisitos mínimos de calidad como parte de las mejores prácticas actuales de seguridad de la información. Esto asegura que una contraseña que se intente adivinar con “fuerza bruta” sea una tarea casi imposible y ayuda al usuario a evitar el robo de información. Considere revisar las pautas OWASP o SANS para la seguridad de contraseñas con el fin de asegurar que la aplicación cumple con las mejores prácticas recientemente aprobadas.

Las contraseñas nunca se deben mostrar en la pantalla de un usuario. Oculte siempre la contraseña con el carácter “asterisco” u otro tipo de glifo.

Además, todas las interfaces que aceptan contraseñas deben utilizar las técnicas para evitar el uso de “fuerza bruta”. También es importante que el proceso que valida la contraseña haga cumplir las políticas establecidas. Por ejemplo, un código JavaScript incrustado en una página web que se muestra en un navegador web no debe implementar este proceso para evitar el uso de “fuerza bruta”. Cualquier atacante experto en Internet puede omitir estos controles interactuando con el servidor de autenticación en el back-end. La tecnología de lucha contra la “fuerza bruta” debe implementarse del lado del servidor en este modelo. En las aplicaciones móviles, donde un pin o contraseña local se guarda en la región de almacenamiento seguro de la aplicación, el dispositivo móvil debe evitar los ataques de fuerza bruta en esta interfaz directamente.

Además, después de cada intento de contraseña no válido, el sistema de control de ataques de “fuerza bruta” debería aumentar el tiempo de espera requerido para volverlo a intentar. También debe haber un límite máximo para intentos de contraseña no válidos. Una vez que se alcanza este umbral, la interfaz debe bloquearse esperando una autenticación de dos factores u otro modelo más complicado de identificación.

Este proceso es extremadamente simple de implementar y requiere muy poco esfuerzo por parte del equipo de ingeniería.

6.9.1 Riesgo

El riesgo de no implementar esta recomendación es que:

- La posibilidad que aquellos dispositivos que se roben puedan ser hackeados utilizando la “fuerza bruta” para adivinar la contraseña
- Los ataques lanzados desde la navegación libre por Internet pueden llegar a hackear los sistemas de IoT simplemente usando contraseñas previamente codificadas en los dispositivos conectados
- Los usuarios pueden verse comprometidos a través de la técnica de "shoulder surfing" (Mirar por encima del hombre) si la interfaz de usuario muestra la contraseña descodificada que se utiliza para entrar en el sistema

6.10 Utilizar un Generador de Números Aleatorios Comercial

Determine si su TCB es capaz de generar números realmente aleatorios. Esto es importante ya que sin esta característica, el proceso de verificación criptográfica puede verse afectado,

haciendo que los datos encriptados sean más fáciles de adivinar y debilitando la integridad de los datos.

Esto también es extremadamente importante para la generación de claves criptográficas únicas. Dado un conjunto de condiciones del entorno, un atacante no debe poder influir en el entorno para hacer que una TCB genere números predecibles durante la generación de claves, la firma o el relleno de mensajes criptográficos.

Este proceso es tan simple como identificar si la TCB es capaz de implementar FIPS [10], EMVCo [11] o la generación de números aleatorios aprobados por “Common Criteria”.

6.10.1 Riesgo

Utilizar criptografía sin un generador de números aleatorios contrastado es peligroso por muchas razones. Si bien las razones son demasiadas para enumerarlas aquí, hay algunas debilidades clave que deben tenerse en cuenta:

- La generación de claves criptográficas puede verse comprometida, provocando la generación de claves débiles o predecibles
- Contraseñas, claves o los paneles de relleno que utilicen un proceso de verificación único son por lo general poco fiables y predecibles
- El relleno de mensajes utilizado para prohibir la posibilidad de reproducción de mensajes puede verse comprometido

Estos problemas pueden provocar fallos significativos en la integridad general de la seguridad criptográfica de todo el ecosistema de IoT. Este riesgo no solo afecta al Dispositivo Periférico, sino que afecta a toda la red.

6.11 Firmar Criptográficamente las Imágenes de las Aplicaciones

Todas las aplicaciones almacenadas fuera de la memoria EEPROM en el core de la CPU deben estar autenticadas criptográficamente. Para hacer esto, simplemente siga el siguiente procedimiento:

- Identificar los metadatos que representan la versión de la imagen de la aplicación
- Generar un hash criptográfico de la imagen de la aplicación, incluidos los metadatos
- Validar que los metadatos de la aplicación coincidan con los metadatos internos
- Validar que el valor de hash coincida con el valor interno del ancla de confianza
- Validar criptográficamente la firma con la clave de firma de la aplicación
- Validar criptográficamente que la clave de firma de la aplicación fue firmada por la Raíz de Confianza de la organización

Este proceso se debe ordenar implementando las actividades menos seguras de tener éxito primero, y las operaciones mas seguras después. De esta forma, se realizará menos trabajo para detectar los riesgos más probables.

Este proceso es excepcionalmente fácil de implementar, especialmente cuando la TCB es capaz de realizar la mayor parte del procesamiento en nombre de la aplicación. El verdadero desafío es más sutil: y es identificar qué aplicación está realizando la operación.

Una aplicación que no ha sido verificada criptográficamente no puede realizar la operación, ya que no tiene forma de saber si su código ha sido “hackeado” por un atacante. Los atacantes normalmente modifican el código en la NVRAM para manipular los sistemas embebidos, si estos no verifican la aplicación.

En su lugar, una aplicación en la EEPROM debe realizar este procedimiento primero sobre cualquier imagen de la aplicación dentro de una memoria persistente externa. Entonces, esa aplicación puede realizar la operación sobre sí misma o puede solicitar una aplicación codificada en la EEPROM de la CPU para realizar este tipo de pruebas en su nombre.

6.11.1 Riesgo

Si la imagen de la aplicación almacenada en el firmware del Dispositivo Periférico (NVRAM) no está firmada criptográficamente, el sistema no podrá diferenciar entre el código autorizado y el código previamente inyectado por un atacante. Esto podría permitir que no solo un atacante manipule el código ejecutable para hackear un Dispositivo Periférico comprometido físicamente, sino que también podría permitir que un competidor instale su propio software en un Dispositivo Periférico.

6.12 Gestionar de Manera Remota el Dispositivo Periférico

Si bien no todos los Dispositivos Periféricos requieren de gestión remota, los que sí lo hacen deben diseñarse de manera que se garantice que atacantes no puedan robar las credenciales administrativas para comprometer algunos (o todos) los Dispositivos Periféricos desplegados sobre el terreno. La solución mas adecuada dependerá de las capacidades del Dispositivo Periférico. Sin embargo, se deben seguir los siguientes lineamientos:

- No almacene los elementos criptográficos privados en un almacenamiento inseguro en los Dispositivos Periféricos, como las claves privadas SSH, las claves privadas TLS o las contraseñas
- Cuando sea posible, genere tokens administrativos (claves criptográficas o contraseñas) para cada Dispositivo Periférico
- Cuando se usen contraseñas para el acceso, se deben aplicar las mejores prácticas con respecto a la complejidad y longitud de las contraseñas al ingresarlas.
- Cuando sea posible, aplique una autenticación de doble factor para los administradores del sistema
- Asegúrese de alertar al usuario final cuando un administrador acceda remotamente al Dispositivo Periférico
- Considere restringir el acceso administrativo a una red privada virtual (VPN)
- No incluya el acceso a funciones administrativas remotas en una aplicación externa o en una API de acceso público, utilice un canal de comunicaciones separado y distinto
- Haga cumplir la confidencialidad y la integridad en el canal de las comunicaciones para la gestión de los dispositivos
- Disminuya el potencial de reproducción de comandos administrativos garantizando que el protocolo de comunicaciones tenga la entropía adecuada mediante el uso de un protocolo de comunicaciones estándar de la industria

6.12.1 Riesgo

Si no se define, implementa y aplica una política para la gestión remota, los Dispositivos Periféricos podrían ser manipulados remotamente. Si no existe un modelo de seguridad claro para el acceso de “super-usuarios” a los Dispositivos Periféricos, los atacantes podrán hacer ingeniería inversa sobre la implementación tecnológica o extraer claves de seguridad de los Dispositivos Periféricos que darán como resultado el acceso a todos los dispositivos Periféricos del ecosistema. El acceso administrativo es a menudo una de las primeras “puertas” tecnológicas utilizadas por los atacantes en los sistemas embebidos, ya que a menudo están mal configurados o no son muy avanzados desde el punto de vista tecnológico.

6.13 Implementar Funciones de Registro y Diagnóstico

Para evaluar los problemas de seguridad en los Dispositivos Periféricos, el Proveedor de servicios de IoT debe evaluar constantemente el comportamiento de los dispositivos y determinar si funciona conforme a lo esperado. Para lograr esto, se deben de seguir tres estrategias:

- La detección de anomalías
- El registro de los datos
- La elaboración de diagnósticos de funcionamiento

Un Dispositivo Periférico debe registrar los datos de su comportamiento interno y enviarlo de forma intermitente a los servicios de back-end para su procesamiento y análisis. Estos registros deben estar compuestos por la actividad estándar, la actividad del kernel y de las aplicaciones y otros metadatos.

La información de diagnóstico también debe ser analizada a intervalos regulares y transmitida al servicio de back-end a través de los registros estándar de actividad o separados específicamente del resto. Los mensajes de diagnóstico deben incluir la mayor cantidad posible de datos en torno al Dispositivo Periférico, incluida la temperatura, duración de la batería, uso de memoria, tiempo de ejecución, lista de procesos (cuando corresponda) y más. Esta información ayudará a identificar cuándo y qué servicio(s) están relacionados con un evento problemático o anómalo.

La detección de anomalías en la red debería ayudar a detectar un problema que no puede identificarse mediante el registro o el análisis de la información de diagnóstico. También ayudará a clasificar los problemas que se pueden observar en los registros o diagnósticos, o atribuir los problemas a un componente específico que puede estar reaccionando mal en el mundo físico (hardware). Por ejemplo, un módem de tecnología celular que se reconecta continuamente a la red o un sensor que genera datos incorrectos.

En conjunto, esta información no solo ayudará a identificar si se observa un defecto en la tecnología sobre el terreno. También ayudará a identificar si el comportamiento errático es indicativo de un evento relacionado con la seguridad.

6.13.1 Riesgo

Si no se implementan los registros de actividad y los diagnósticos, la empresa puede estar perdiendo información crítica. Esta información puede no solo afectar a la seguridad del ecosistema, sino que puede ayudar a diagnosticar fallos críticos de diseño del producto IoT.

6.14 Forzar la Protección de Memoria

Los sistemas embebidos a menudo se diseñan con microcontroladores que no son muy avanzados, sin unidades de administración de memoria (MMUs) y sin unidades de protección de memoria (MPUs). Sin embargo, este tipo de tecnologías deben emplearse en cualquier plataforma que quiera:

- Ejecutar aplicaciones sin privilegios específicos
- Ejecutar aplicaciones o aplicaciones no fiables (de terceros)
- Ejecutar un emulador o máquina virtual en un proceso sin privilegios específicos

Cualquier entorno que requiera la ejecución de una aplicación no privilegiada debe poder protegerse contra aplicaciones deshonestas o comprometidas. Esto garantiza que estas últimas no puedan acceder a las áreas de memoria que controlan los recursos vitales para la seguridad, como una TCB, el controlador del ancla de confianza o los registros hardware de los periféricos.

El desafío en esta área a menudo consiste en la migración de una plataforma con un microcontrolador de 8 bits a una plataforma más robusta, utilizando un microcontrolador de 32 bits o una arquitectura de procesador más completa. Sin embargo, hay muchos sistemas operativos disponibles gratuitamente o con una licencia de uso nominal barata para los sistemas embebidos que implementan correctamente la protección de memoria con una MPU o una MMU.

6.14.1 Riesgo

Si no se utilizan estas tecnologías, las aplicaciones deshonestas o posiblemente comprometidas no tendrán restricciones para alterar recursos básicos como los controladores, registros periféricos o incluso servicios privilegiados como el kernel y otras aplicaciones. La falta de protección de la memoria permite que cualquier aplicación tenga acceso completo a todos los registros de la memoria integrada en el microcontrolador o procesador. Las aplicaciones sin privilegios concretos no deben de poder utilizar estos recursos.

6.15 Forzar el Arranque Fuera de la EEPROM Interna

Normalmente el código de un gestor de arranque se instala dentro de la memoria tipo EEPROM de sólo lectura, dentro de la CPU. Esto no es siempre así, sin embargo, determine si su CPU carga el gestor de arranque desde una fuente externa. Si la CPU no tiene EEPROM que le permita verificar el código del gestor de arranque, este puede ser manipulado por un atacante local para configurar la CPU a su gusto.

Dependiendo del nivel de protección existente en el circuito integrado o región de memoria que guarde el gestor de arranque, un atacante podría usar un bus local (como el bus serie de la SPI) o una API remota (como la ofrecida por el firmware para la gestión OTA) para

manipular el código embebido en el microprocesador. Esto dará como resultado que un atacante sea capaz de modificar la plataforma de computación colocando código personalizado en el punto de ejecución más fiable: la primera parte del código ejecutable. Otro ataque podría simplemente intercambiar el circuito físicamente que integra el gestor de arranque por un circuito propio del atacante que contiene instrucciones personalizadas, desoldando y soldando el nuevo chip en la PCI. Sin una forma de verificar la integridad del código externo, el usuario no podrá distinguir entre un software certificado y uno que no lo es.

Para personalizar un gestor de arranque, un atacante tendría que desarrollar o externalizar el desarrollo del gestor de arranque. Dependiendo de los recursos disponibles, y del procesador objetivo del hack, la dificultad de este procedimiento es variable, desde muy sencillo hasta extremadamente complicado.

Considere usar una CPU o MCU/MPU con una EEPROM interna o una NVRAM con capacidad de bloqueo para almacenar el gestor de arranque. Esto garantizará que la plataforma pueda al menos verificar el primer ejecutable cargado y ejecutado en la CPU, esto implica que el dispositivo será mas seguro.

6.15.1 Riesgo

No evaluar la cadena de confianza y no verificar la integridad del código de arranque en la CPU puede resultar en un compromiso total del sistema. Estos pasos son fundamentales para garantizar la seguridad de un Dispositivo Periférico de IoT y, por lo tanto, de todo el ecosistema.

6.16 Bloquear las Secciones de Memoria Críticas

Las aplicaciones críticas almacenadas en regiones de memoria ejecutables, como los gestores de arranque de la fase inicial o las Bases de Computación Confiable, deben almacenarse en una sección de memoria de sólo lectura. Esto asegura que el dispositivo se puede iniciar con una configuración válida sin dejar intervenir a ningún factor externo. Sin esta garantía, el código ejecutable cargado durante la primera fase de ejecución, no podrá ser considerado cien por ciento fiable desde el punto de vista de la configuración inicial del sistema y del estado al que se llega después del arranque.

Si bien es cierto que los atacantes aún pueden modificar el sistema reemplazando estas secciones críticas de la memoria con su propio código, les exige implementar su propia versión personalizada del software, lo que puede ser un proceso muy complejo y difícil. Esto aumenta enormemente el costo total del ataque y la habilidad requerida para poder ejecutarlo. Además, si se aplican las estrategias de personalización y de aprovisionamiento, estos pasos obligarán al atacante a recrear el proceso para cada uno de los Dispositivos Periféricos, personalizando su solución a las características criptográficas únicas del sistema local. Esto hace que el ataque general sea excepcionalmente costoso y disminuye su viabilidad.

Para evitar este riesgo, simplemente identifique si los pasos necesarios para almacenar secciones críticas en la memoria pueden deshabilitarse. Alternativamente, comience por utilizar una tecnología EEPROM que se pueda bloquear.

Asegúrese de que, si se usa un proceso de bloqueo, este último no se configure por software. Los bloqueos implementados por software solo se habilitan después de que el software haya ejecutado la funcionalidad necesaria para activar dicho bloqueo. Habrá una ventana de unos milisegundos en la que un atacante puede modificar el estado de la memoria a su favor. Por lo tanto, los bloqueos por hardware, como el uso de fusibles o bits programables para el bloqueo, siempre se deben emplear cuando sea posible.

6.16.1 Riesgo

Sin estados de bloqueo o de solo lectura, un atacante puede alterar fácilmente las secciones críticas de la memoria. Esto puede darles el suficiente privilegio para comprometer toda la plataforma del Dispositivo Periférico sin necesidad de otras acciones, burlando así todos los controles de seguridad posteriores utilizados en el sistema.

6.17 Evitar los Gestores de Arranque Inseguros

El trabajo de un gestor de arranque no es solo configurar la CPU para la ejecución de una aplicación principal, sino también cargar y transferir el control de ejecución a la aplicación. Para lograr esto, el gestor de arranque normalmente encuentra y carga la aplicación primaria en la memoria principal de la CPU. El problema surge cuando gestores de arranque predeterminados por defecto se usan en ciertos tipos de sistemas.

Muchos gestores de arranque utilizados comercialmente en los microcontroladores, por ejemplo, permiten que el firmware externo se cargue en la memoria de la CPU para su ejecución, o permiten actualizaciones del firmware a través de interfaces serie. Otros gestores de arranque pueden “pedirle” al usuario las ubicaciones en memoria que contengan imágenes de la aplicación que se quieran ejecutar, lo que permite que el usuario ejecute cualquier aplicación que elija.

Si bien esta funcionalidad es deseable en un entorno de computación estándar, de escritorio, computadora portátil o incluso en un servidor, esto es inaceptable en los sistemas embebidos. Esto se debe a que si un gestor de arranque carga y ejecuta una aplicación no verificada y no fiable, no hay garantía en cuanto a la fiabilidad o seguridad de la aplicación ejecutada, lo que deja al dispositivo embebido en un estado incierto.

Por lo tanto, para remediar este problema:

- El gestor de arranque debe ser capaz de verificar criptográficamente la imagen de la aplicación que se ejecutará
- El gestor de arranque predeterminado/estándar no se debe usar si permite imágenes alternativas o copias del firmware ejecutable a través de memoria externa
- El gestor de arranque no debe permitir imágenes de aplicaciones cargadas desde ubicaciones de almacenamiento arbitrarias
- La imagen ejecutable del gestor de arranque de la primera fase debe estar bloqueada en la EEPROM y solo debe actualizarse a través de un proceso seguro

Además, el diseño de un gestor de arranque debería estar sujeto al análisis de un analista de seguridad externo. Comprometer un gestor de arranque a través de la manipulación de errores en el software puede conducir a la ejecución de código personalizado, o a omitir las

comprobaciones de verificación de integridad. Esto puede llevar a que se pueda hacer un ”jail-break” para el sistema operativo del dispositivo, esto puede afectar muy negativamente al negocio del proveedor de servicios IoT en concreto. Asegúrese de que todos los gestores de arranque utilizados en el sistema se auditen a fondo para detectar fallos en el software que podrían generar riesgos de seguridad.

6.17.1 Riesgo

Un gestor de arranque inseguro puede ser tan dañino como un proceso de arranque mal estructurado en un procesador. Asegurar la integridad del gestor de arranque es un paso fundamental para garantizar la integridad al fin y al cabo del Dispositivo Periférico IoT.

6.18 Implementar la Transmisión de Claves con Secreto Perfecto hacia Adelante (PFS)

El Secreto Perfecto hacia Adelante (PFS) se ocupa del intercambio de las claves criptográficas durante la configuración de las comunicaciones entre dos Dispositivos Periféricos. En general, los Dispositivos Periféricos tendrán certificados asimétricos que se utilizarán para autenticar sus identidades. Una vez completada la fase de autenticación, se genera una clave simétrica y se acuerda mutuamente mediante el uso de cifrado asimétrico para proteger la negociación de claves. Una vez que se genere y acuerde esta clave, se utilizará para asegurar el resto de la sesión entre las dos entidades. Esto se hace para reducir la necesidad de recursos de computación involucrados en la criptografía asimétrica. La criptografía simétrica es computacionalmente más económica, lo que significa que es más rápida y consume menos energía en los dispositivos embebidos o de baja potencia.

Sin embargo, hay una trampa. Este modelo de acuerdo de clave común presupone que las claves asimétricas siempre se mantienen en secreto. Este puede no ser el caso. En el futuro, una entidad con suficientes recursos podrá ser capaz de calcular la clave privada para cualquier clave pública asimétrica. Si el atacante guarda la información de cada sesión en las comunicaciones entre una entidad objetivo y sus pares, la entidad podrá descifrar cada mensaje de las comunicaciones del pasado generando la clave privada en algún momento posterior.

Además, la clave criptográfica de un servidor puede verse comprometida por terceros anónimos o incluso por personas directamente involucradas con el negocio. Si esto ocurre, cualquiera que haya estado almacenando mensajes de las comunicaciones protegidos por una clave asimétrica robada, podrá descifrar esos mensajes.

Una solución a este problema es generar un par de claves asimétricas efímeras durante el proceso de negociación de claves. Solo la clave pública para este par de claves efímeras se transmite a cada lado del enlace de comunicación, y se puede usar para las operaciones con una clave simétrica.

Esta clave efímera debe generarse con suficiente entropía y con un tamaño de clave lo suficientemente grande como para imposibilitar un ataque de agotamiento de los recursos computacionales dentro de un período de tiempo razonable. Esto asegurará que el proceso de negociación de claves sea sostenible y menos susceptible de ser atacado en el futuro.

Además, esta metodología garantiza que los pares utilicen su clave asimétrica persistente solo para la autenticación, no para la confidencialidad y la integridad. Si esta clave asimétrica es robada o expuesta al público, solo afectará el proceso de autenticación, no a la confidencialidad e integridad del canal de comunicaciones.

Para que este proceso sea aún más resiliente frente a ataques, la clave asimétrica utilizada para la autenticación debe estar sujeta a un proceso de revocación seguro que garantice que un Dispositivo Periférico pueda verificar si una clave se ha visto comprometida. El Dispositivo Periférico ya no debería confiar en esa clave para la autenticación si se le ha notificado que se ha producido tal compromiso.

6.18.1 Riesgo

No implementar PFS puede exponer todas las comunicaciones de red a un atacante, si este alguna vez obtiene acceso a una clave privada utilizada para proteger el canal de comunicaciones. En cualquier momento posterior, si el adversario captura la clave privada, todas las comunicaciones capturadas por el adversario en el pasado podrán ser descifradas. Esto conllevará graves consecuencias.

6.19 Implementar Comunicaciones Seguras entre los Dispositivos Periféricos

Si bien este aspecto está cubierto por otras recomendaciones y riesgos a lo largo de esta guía, es importante tener en cuenta que la seguridad de las comunicaciones de un Dispositivo Periférico es la mayor amenaza que puedan tener dentro de un ecosistema de IoT. La capacidad de un adversario para manipular el canal de comunicaciones es la forma más sencilla de que un Dispositivo Periférico se vea comprometido.

Como resultado, los diseñadores de Dispositivos Periféricos deben implementar la seguridad de las comunicaciones desde las siguientes perspectivas:

- Autenticación de pares de red
- Confidencialidad de los datos
- Integridad de los mensajes

Aunque los mensajes de texto sin encriptar pueden enviarse y recibirse para interactuar con Dispositivos Periféricos diseñados por otras organizaciones, los datos transferidos a través de cualquier canal que incorpore comandos, datos de privacidad del usuario o mensajes críticos del sistema deben estar protegidos. El primer paso es autenticar el dispositivo que desea comunicarse para garantizar que sea el que dice ser. Esto es especialmente importante para verificar si este par en las comunicaciones representa un servicio del sistema.

A continuación, se requiere la confidencialidad de los datos para garantizar que terceras partes no puedan leer los datos críticos que se transmiten a través del canal de comunicaciones.

Finalmente, la integridad del mensaje es vital para garantizar que los mensajes secretos no hayan sido manipulados por un atacante.

Estos tres atributos, combinados, darán como resultado un modelo de comunicaciones que puede sobrevivir durante años con pocos cambios en su concepción.

Este proceso se simplifica mucho mediante el uso de protocolos de seguridad existentes y probados, como, entre otros, los siguientes:

- El último estándar TLS aprobado
- El último estándar DTLS aprobado
- SSH2 para autenticación e intercambio de claves
- GBA para generación e intercambio de claves
- OAuth2 para autorización
- "BEST" (Battery Efficient Security"), seguridad de batería eficiente para dispositivos de comunicación que sean ligeros con pocos recursos computacionales (MTC) [21]

Si bien el equipo de ingeniería puede usar cualquier "suite" de computación que cumpla con los requisitos antes mencionados, la utilización de un conjunto de protocolos de comunicaciones estándar reducirá el número de errores que se puedan observar sobre el terreno. Esto se debe a que los expertos en seguridad de la información y criptografía están involucrados en el desarrollo de protocolos estandarizados.

Las propiedades de la seguridad en las tecnologías de comunicaciones celulares basadas en 3GPP, incluidas las tecnologías de red LPWA estandarizadas como NB-IoT y LTE-M, se pueden encontrar en el documento GSMA PRD CLP.14 [4].

6.19.1 Riesgo

Aunque no hace falta decir que la seguridad de las comunicaciones es un requisito fundamental, pero muchas veces falta aclarar el "*porqué*". La seguridad de las comunicaciones no solo garantiza que un adversario no pueda leer los datos. También asegura que:

- Un dispositivo Periférico no pueda ser suplantado
- Un servicio crítico no pueda ser suplantado
- Se puedan detectar mensajes modificados
- Los cambios en el software o en las configuraciones de seguridad se pueden realizar de forma segura

Sin comunicaciones seguras, no hay garantías en cuanto a la calidad, fiabilidad o privacidad de un producto o servicio de IoT.

6.20 Autenticar la Identidad de los Dispositivos Periféricos

Si cada Dispositivo Periférico lleva una identidad criptográficamente única, como un número de serie único, el dispositivo debe poder demostrar que realmente representa ese número de serie. Para hacer esto, la TCB debe firmar criptográficamente un mensaje con una clave conocida solo por la TCB y el servicio de back-end IoT, la solución tecnológica implementada por GBA es una posible solución para resolver este problema. El mensaje debe contener la identidad única (número de serie u otro token) y los metadatos correspondientes al Dispositivo Periférico.

El mensaje que debe firmar la TCB también debe contener un reto criptográfico emitido por el sistema de back-end. Esto evita que un adversario pueda reproducir un mensaje de autenticación ya enviado desde la TCB al back-end. Si una entropía suficiente se incluye en el desafío, la posibilidad de repetir el mensaje es casi inexistente.

Para generar un reto criptográfico con relación a la identidad de un Dispositivo Periférico se debe:

- Recibir una solicitud del Dispositivo Periférico que contiene el token de identidad único
- Generar un reto único y enviarlo al Dispositivo Periférico
- Recibir la respuesta al reto del Dispositivo Periférico que contiene la firma y el mensaje
- Verificar que la firma sea correcta usando la clave compartida
- Asegurar que el mensaje firmado contenga el token de identidad correcto y cualquier otro metadato relevante
- Confirmar la firma verificada

Para procesar un reto uno debe:

- Conectarse al sistema de back-end
- Recibir la identidad criptográfica del sistema de back-end
- Autenticar criptográficamente la identidad del sistema de back-end utilizando la TCB
- Enviar un mensaje que contenga la identidad del Dispositivo Periférico y otros metadatos al back-end
- Recibir un reto del back-end
- Generar un mensaje que contiene el token de identidad único, los metadatos y el desafío
- Firmar el mensaje
- Enviar el mensaje y su firma al back-end
- Verificar que el sistema de back-end haya aprobado el mensaje firmado

6.20.1 Riesgo

El riesgo de no implementar esta recomendación es que los Dispositivos Periféricos podrán ser clonados o vulnerables a los ataques de suplantación. Esto puede abrir la infraestructura de la empresa a ataques de competidores y adversarios. Los competidores pueden utilizar la falta de autenticación de la identidad del Dispositivo Periférico para construir una plataforma que compita a partir de la misma lista de materiales para su implementación, pero a un costo menor.

Alternativamente, un competidor puede usar la falta de autenticación para vender hardware que se puede conectar a la infraestructura que la empresa utiliza para su servicio de IoT. Estos problemas pueden generar una pérdida de ingresos para la empresa y un mayor gasto operativo, ya que el competidor puede beneficiarse del uso de la infraestructura de red de la empresa, aunque no paguen por usarla. Debido a que el ancho de banda de la red, el uso de los servidores en la nube, de la CPU, del disco y otros recursos tienen un

costo cuantificable, este tipo de competencia desleal puede tener un impacto grave en el negocio de una organización vulnerable.

7 Recomendaciones de Alta Prioridad

Las recomendaciones de alta prioridad representan el conjunto de recomendaciones que deben implementarse, pero solo si la arquitectura del Dispositivo Periférico lo requiere. Por ejemplo, no todas las arquitecturas de Dispositivo Periférico requieren una carcasa de producto a prueba de manipulaciones. Estas recomendaciones deben evaluarse para determinar si el caso de negocio las considera como un requisito fundamental.

7.1 Uso de la Memoria Interna para los Secretos

Cuando sea posible, los procesadores deben usar la memoria interna de la CPU para el procesamiento de los secretos mas importantes y las claves criptográficas que no se encuentren dentro de un ancla de confianza. Esto asegurará que, si un atacante está monitoreando o es capaz de manipular el bus de memoria, no obtendrá secretos fundamentales, pero solo verá los efectos del uso de estos secretos en una aplicación en ejecución.

Este modelo asegurará el mantenimiento en el tiempo de los secretos criptográficos, lo que obligará al atacante a desistir de tratar de descubrir esos secretos. En cambio, el atacante tendrá que confiar en la manipulación de bits en la memoria RAM que equivale a los efectos del uso de dichos secretos. Esto requerirá que el atacante cambie los bits en la memoria cada vez que los secretos se usen internamente, incrementando enormemente la complejidad del ataque.

No todos los sistemas operativos definen modelos para utilizar la RAM interna para el procesamiento de secretos. Por lo tanto, podría ser necesario que el equipo de ingeniería lo implemente. Si bien este proceso no es difícil, tampoco es trivial. El código ejecutable debe garantizar que todas las rutinas de uso de memoria usen regiones específicas garantizadas y que estén localizadas en la memoria interna del procesador. Esto puede requerir trabajo adicional, según el sistema operativo y el conjunto de herramientas del compilador utilizado.

7.1.1 Riesgo

La mayoría de los microprocesadores y algunas CPUs tienen una pequeña cantidad de SRAM interna dedicada al código que se ejecuta desde una EEPROM interna o una NVRAM interna. Normalmente, esta SRAM es inaccesible para los periféricos externos, a menos que esté expuesta a propósito mediante el uso de métodos tipo DMA (Acceso Directo a Memoria). Si se mantienen en privado, los secretos criptográficos procesados por el código tienen una probabilidad mucho menor de estar expuestos a adversarios capaces de interceptar las comunicaciones con la RAM.

Si bien el riesgo no es alto, los secretos criptográficos no deberían transmitirse por los buses de acceso público, a fin de disminuir la posibilidad de un ataque. Los atacantes bien equipados capaces de interceptar comunicaciones con las RAM a velocidades potencialmente altas pueden capturar datos y entre ellos los secretos criptográficos. Sin embargo, se necesitaría un ingeniero capacitado para realizar ingeniería inversa y capturar mensajes en las RAM que podrían atribuirse a operaciones criptográficas.

Como resultado, si bien esta es una recomendación importante, puede no ser crítico para garantizar la seguridad física. Si las claves criptográficas principales se almacenan dentro de un ancla de confianza, y la aplicación solo procesa las claves de sesión, no es probable que la captura y procesamiento de las claves en la RAM externa produzca un compromiso inmediato. Sin embargo, esto supone que la arquitectura criptográfica limita las claves expuestas a aquellas que no son críticas para las operaciones de IoT básicas, como la rotación de claves, la generación de claves de sesión y la revocación de certificados.

7.2 Detección de Anomalías

El modelado de comportamiento de Dispositivos Periféricos es una parte imprescindible para la seguridad en sistemas IoT. Esto se debe a que un Dispositivo Periférico ya comprometido no se puede distinguir de un Dispositivo Periférico que se comporta normalmente si solo se registran y analizan las interacciones correctas con el dispositivo. Para obtener una perspectiva más completa de un entorno IoT, la huella digital del comportamiento de un dispositivo en su totalidad debe catalogarse para identificar anomalías que pueden ser indicativas de un comportamiento anómalo.

Este tipo de comportamiento se puede detectar a través de:

- Reinicios erráticos o “resets” del dispositivo
- Conexiones intermitentes a una red de comunicaciones a intervalos irregulares
- Conexión a servicios finales no aprobados, o conexión a servicios finales en momentos inapropiados
- Una huella digital del tráfico de red significativamente diferente de lo normal
- Múltiples mensajes mal formados enviados desde el Dispositivo Periférico a servidores de dispositivos

Si el comportamiento normal de un Dispositivo Periférico es catalogado por el Proveedor de Servicios de IoT, la organización podrá identificar patrones de comportamiento que deberían ser indicativos de un comportamiento anómalo. Al establecer una base de referencia para el comportamiento y luego supervisar continuamente los posibles valores atípicos, la empresa puede diagnosticar más rápidamente los problemas de seguridad y rendimiento en los entornos de producción.

Catalogar las huellas digitales del comportamiento de los dispositivos también puede ayudar a la empresa a vincular más rápidamente un conjunto de funcionamientos defectuosos a una característica en particular o a condiciones determinadas del entorno de ejecución. Esto puede conducir a soluciones de ingeniería a un ritmo más rápido que si no se recopilan datos de comportamiento.

7.2.1 Riesgo

Sin la detección de anomalías, se podría tardar muchísimo tiempo en detectar que un Dispositivo Periférico ha sido comprometido dentro de un ecosistema de IoT. Si el comportamiento anómalo del Dispositivo Periférico solo es visible fuera de las fases de operación normales del dispositivo, el equipo de gestión puede no tener motivos para desconfiar del Dispositivo Periférico. Sin embargo, si la detección de anomalías se

implementa en todo el ecosistema, el comportamiento malicioso puede detectarse -y, por lo tanto, contenerse- mucho antes.

7.3 Usar un Encapsulado o Carcasa de Producto a Prueba de Manipulaciones

El dispositivo físico no solo debe ser resistente a las alteraciones a nivel de los circuitos integrados, sino que también debe ser resistente a las alteraciones a nivel de producto. El encapsulado o carcasa utilizada en el producto debe proporcionar protección frente a adversarios o usuarios "curiosos". Esto puede implementarse de varias maneras:

- Circuitos que invalidan la NVRAM cuando se abre el encapsulado
- Sensores que funden los fusibles de seguridad cuando se detecta luz
- Sensores que activan una alerta cuando se mueve un dispositivo físicamente de su posición inicial
- Epoxi que cubre los componentes/circuitos del core del sistema
- Las alertas generadas cuando componentes internos o extraíbles se mueven a quitan del dispositivo

El uso de estas metodologías puede disminuir la posibilidad de que un equipo (Dispositivo Periférico) sea manipulado. Sin embargo, puede ser más rentable mejorar el diseño del circuito. Si bien estas metodologías reducirán en gran medida el potencial de compromiso de los hackers aficionados amateurs o atacantes, no reducirán las capacidades para comprometer un equipo de los analistas de seguridad bien equipados y con experiencia.

Por lo tanto, estos métodos mejoran la capacidad de la empresa para garantizar que el producto en sí no pueda ser manipulado físicamente mientras no esté en posesión del usuario o propietario. En otras palabras, si un usuario deja su dispositivo en casa o "en el terreno", un atacante no solo debe obtener acceso físico para comprometer el dispositivo, sino que también debe evitar los controles de seguridad a prueba de manipulaciones para modificar y luego reemplazar el dispositivo con su propio "clon" personalizado. Esto evita la posibilidad de que los dispositivos se vean comprometidos y reemplazados rápidamente, lo cual mejora considerablemente la seguridad física del dispositivo.

Sin embargo, si el modelo de amenaza ignora este aspecto y se centra en remediar un ataque físico genérico, incluidos los hechos por atacantes avanzados y con equipamiento ad hoc, no soluciona por completo esa amenaza. En ese caso, estas características de seguridad adicionales que tienden a evitar la manipulación ralentizarán a un atacante, pero no detendrán por completo a un atacante con tiempo y experiencia.

Por lo tanto, se debe alcanzar un equilibrio entre lo que es rentable y el modelo de amenaza de un dispositivo en concreto. Sin embargo, los atacantes inteligentes han ideado componentes similares físicamente a los cajeros, "skimmers", para ser adaptados físicamente al cajero automático y poder robar las credenciales de los usuarios (p.ej. a través de una cámara integrada). Por lo tanto, la protección contra la manipulación física solo puede lograr parte del resultado deseado. El diseño de la aplicación y el hardware debe ser un paso adicional para disminuir los efectos de los ataques físicos.

Los ingenieros y los directivos empresariales deben evaluar el modelo de amenaza posible de un producto o servicio determinado y equilibrar el riesgo de ataque con las medidas de resistencia a la manipulación implementadas en el dispositivo. Cada tipo de resistencia a la manipulación tendrá un costo, dependiendo del proceso, la ingeniería y los materiales involucrados. Y, sin embargo, este esfuerzo y solución puede no dar como resultado el nivel de seguridad deseado.

Un ejemplo de este problema son los circuitos integrados con recubrimiento de epoxi. Si bien este proceso es válido, hay dos cosas que un atacante puede hacer fácilmente para eludir el uso de epoxi:

- Utilización de circuitos de derivación que se conectan al componente recubierto de epoxi
- Eliminar físicamente el epoxi

Si bien el epoxi oculta el componente a la vista, no obstaculiza (ni puede hacerlo) los electrones que se trasmiten a través de los circuitos que se conectan al chip revestido con epoxi. Por lo tanto, si los secretos críticos se envían a través de un bus hardware, la resina de epoxi no detendrá la posibilidad de que un atacante intercepte estos datos.

Además, el epoxi en sí mismo simplemente puede eliminarse. Varias técnicas caseras han sido descubiertas en los últimos años por aficionados que describen claramente un método práctico para eliminar epoxi de un circuito utilizando productos químicos y procesos disponibles para cualquier consumidor. Si bien el proceso puede involucrar sustancias cáusticas y potencialmente peligrosas, los procedimientos descritos por ingenieros con experiencia en ingeniería inversa son muy detallados y pueden ser implementados por cualquier persona en un laboratorio u oficina debidamente ventilados.

Por lo tanto, se debe realizar una evaluación de riesgos que pondere claramente los beneficios de la tecnología resistente a la manipulación con la facilidad de compromiso. Si cada dispositivo aleatoriamente debe protegerse simplemente para que un adversario no pueda manipularlo o modificarlo de alguna forma, se deben emplear las técnicas de resistencia a la manipulación. Si el requisito es que los atacantes avanzados no puedan interceptar mensajes en los buses hardware, se debe considerar una arquitectura de seguridad que proteja la aplicación y al sistema operativo por encima de la implementación de una tecnología resistente a la manipulación.

7.3.1 Riesgo

Como se señaló en la sección anterior, el riesgo de no desplegar soluciones resistentes a la manipulación varía enormemente según los requisitos del dispositivo. Si el requisito es que el dispositivo debe alertar al usuario si un dispositivo se ha manipulado físicamente, se ha roto o se ha modificado de alguna forma, la resistencia a la manipulación es importante. Si el requisito es que el dispositivo debe estar protegido contra el análisis por parte de un experto en seguridad o atacante sean aficionados o no, la implementación de la seguridad en la arquitectura es probablemente la solución mas apropiada para este tipo de riesgo.

En cualquier caso, el riesgo de no desplegar soluciones resistentes a la manipulación es que no se va a poder detectar jamás si alguien ha manipulado físicamente el dispositivo. Si

bien esto puede no ser muy importante para las aplicaciones con una arquitectura robusta y con una seguridad reforzada, si que es vital para productos que ofrecen servicios críticos para sus usuarios, como dispositivos médicos, sistemas telemáticos y de seguridad doméstica o de automatización.

7.4 Forzar la Confidencialidad y la Integridad, desde y hacia el Ancla de Confianza

Todas las comunicaciones hacia y desde el ancla de confianza deben ser autenticadas y deben hacer cumplir la confidencialidad e integridad de los datos. La única excepción a este modelo es cuando el ancla de confianza se integra en el núcleo del procesador. Sólo se puede confiar en cualquier ancla de confianza externo, como el implementado en un UICC, cuando los mensajes transmitidos hacia o desde el ancla están protegidos.

Para ello, elija anclas de confianza que sean capaces de autenticar y cifrar las comunicaciones, y valide que todos los mensajes que contengan respuestas a desafíos se envíen de manera confidencial y, de ser posible, con integridad verificable.

Los UICC que se pueden gestionar utilizando “Secure Channel” (Canal Seguro) son capaces de mantener la confidencialidad y la integridad. El proveedor de servicios de IoT debe analizar con el operador de red si la tecnología “UICC Secure Channel” puede utilizarse para proteger sus aplicaciones IoT. En un futuro cercano, un eUICC será capaz de proteger las aplicaciones. Entonces, se podrá usar “Secure Channel” para volver mas seguras las aplicaciones de los Dispositivos Periféricos desde el proceso iniciado por el gestor de arranque hasta el proceso de autenticación en la red.

Si bien esto debería ser un ejercicio relativamente sencillo, hay sutilezas en este proceso. Es necesario probar cada aspecto de la capa de comunicaciones. Algunos mensajes provenientes de varios tipos de anclas de confianza pueden no ser confidenciales o estar garantizados desde el punto de vista de su integridad. Por ejemplo, un mensaje que indica si una operación tuvo éxito o no, puede no parecer ser malicioso, pero debe protegerse de todas formas para garantizar que un atacante no envíe una respuesta personalizada, engañando a la aplicación.

Algunas anclas de confianza pueden no ser capaces de garantizar la integridad en el canal de comunicaciones. Se prefiere la integridad, y debe emplearse para garantizar que un mensaje no haya sido manipulado. Pero hacer esto requiere una base de confianza tanto en el procesador principal (“anfitrión”) como en el ancla de confianza, lo que puede no ser razonable para la aplicación.

Dado que todos los sistemas embebidos pueden sufrir un ataque directamente sobre el hardware por parte de un adversario suficientemente equipado, puede ser demasiado requerir una raíz de confianza en ambos procesadores simplemente para proteger las comunicaciones de un bus local. Sin embargo, en aplicaciones donde la seguridad física es crítica, la integridad debe de implementarse.

7.4.1 Riesgo

El riesgo de no hacer cumplir la confidencialidad e integridad puede llegar a ser muy “interesante”. Este riesgo puede variar desde un compromiso completo del sistema hasta la

obtención de información que no dañe directamente al ecosistema de IoT. Esto se debe a que ciertos mensajes pueden entrar en juego en este caso. Por ejemplo, si una TCB solicita que el ancla de confianza verifique la integridad de un mensaje, transmitirá el mensaje a través de un bus hardware al ancla de confianza.

Si el ancla de confianza se encuentra dentro de la CPU, es poco probable que un atacante pueda alterar este mensaje sin un equipo sofisticado y costoso. Sin embargo, si el ancla de confianza se implementa en un chip separado en la PCI, puede haber una oportunidad para que el atacante altere el mensaje “puenteando” físicamente el circuito e insertando su propio hardware. Si, por ejemplo, el ancla de confianza recibe el mensaje y simplemente responde a la consulta diciendo "Sí, este mensaje es válido" sin implementar alguna técnica para forzar la comprobación de la integridad del mensaje, la TCB no podrá verificar si el mensaje ha sido manipulado por un atacante con acceso físico al bus.

Además, incluso si la integridad de la respuesta se verificase, un adversario con acceso físico al bus puede simplemente comprometer el circuito, capturar la solicitud de mensaje a la TCB, emitir su propio mensaje fiable al ancla de confianza y esta responda normalmente a la TCB. Si el bus de comunicaciones hardware no está protegido adecuadamente, este ataque también es posible, anulando la capacidad del ancla de confianza de realizar su trabajo.

Sin embargo, el hecho de esperar que tanto la CPU como el ancla de confianza tengan anclas de confianza internas individuales crea una paradoja. ¡Cómo puede una CPU que tenga un proceso de arranque, confiar en sí misma si un adversario puede cambiar la CPU, además la CPU tiene que usar su propia EEPROM para verificar la integridad del ancla de confianza! Esto crea un “rompecabezas”, pero que puede ser resuelto.

Una solución es insertar una clave pública en la ROM de la CPU. Esta clave se puede usar para verificar la integridad de los mensajes enviados por el ancla de confianza. Si se transmite un mensaje arbitrario (por verificar) a través del bus de hardware al ancla de confianza, el ancla de confianza puede responder con un mensaje firmado que incluya el mensaje original como parte de la respuesta. Esto verifica que el mensaje en realidad se originó a partir del ancla de confianza, y que el mensaje que se procesó es de hecho el mensaje que se esperaba procesar. La única preocupación que queda, es asegurarse de que los “nonces” (números aleatorios) utilizados en el relleno de mensajes aseguren que los mensajes criptográficos no se puedan volver a reproducir.

Con lo anterior en mente, es fácil identificar que la criptografía puede fallar debido a problemas muy sutiles, no sólo en la criptografía, sino también en los algoritmos que se utilizan en las comunicaciones criptográficas. Es por lo que la implementación de la confidencialidad y la integridad (correctamente) es tan importante.

7.5 Actualizaciones OTA de las Aplicaciones

La actualización remota de la imagen de una aplicación de un Dispositivo Periférico puede ser un proceso simple y directo. La complejidad proviene de una solución ingenieril demasiado compleja que en realidad no aborda fallos en la seguridad de manera realista. Desde una perspectiva de almacenamiento persistente, el proceso de diseño es muy simple:

- Definir una ubicación para la imagen de la aplicación activa
- Definir una ubicación para la copia de seguridad de la imagen de la aplicación (si corresponde)
- Definir una ubicación para la imagen de la aplicación de emergencia
- Si existe un espacio de backup para la imagen de la aplicación, actualice este espacio con la imagen activa
- Verifique criptográficamente la imagen activa usando la firma almacenada en la TCB
    - Esto asegura que el medio de almacenamiento no está dañado, y que un adversario no modificó los bits durante el proceso de escritura
- Descargue la nueva imagen en su totalidad o por partes y sus metadatos y firma
- Parchear la imagen activa con las partes transmitidas
- Verificar la firma criptográfica utilizando la TCB
- Reiniciar con la nueva imagen de la aplicación

Si el proceso falla en cualquier punto, el sistema debe volver a una imagen de respaldo para asegurar que la aplicación funciona según lo estipulado, o el sistema de emergencia puede usarse para llamar al centro de gestión y notificar al Ecosistema de Servicios IoT que se ha producido un fallo.

La dificultad proviene de la creación de un modelo de almacenamiento que resuelve dos problemas:

- Un atacante que intenta manipular el proceso de actualización
- Una anomalía hardware

Sin un sistema de respaldo o partición de emergencia, el dispositivo no tendrá más remedio que fallar. Debido a que los sistemas embebidos generalmente no tienen interfaces de usuario robustas, esto puede presentar un punto significativo que genere estrés y problemas entre la empresa y sus clientes. Si se registra un fallo, se tiene que hacer de la manera mas elocuente posible no solo para la tranquilidad del usuario, sino también para la fiabilidad del sistema.

Es notable que algunos atacantes pueden querer corromper el proceso de actualización a propósito, para forzar un estado en el sistema persistentemente vulnerable. Por ejemplo, si se encuentra una vulnerabilidad que se pueda aprovechar en la versión actual y activa de la aplicación, y hay un parche disponible para la versión más reciente de la aplicación.

El beneficio de este modelo es que incluso si el atacante inhabilita el proceso de negociación de la red, el sistema de back-end tiene la oportunidad de tomar nota de este evento. Si la red de back-end identifica que un nodo se está comunicando normalmente excepto durante las actualizaciones, se debe generar una alerta para que el administrador del sistema determine si ese Dispositivo Periférico en concreto se está utilizando de forma anormal.

7.5.1 Riesgo

Si el proceso de actualización de la aplicación OTA no está correctamente estructurado, puede provocar que los adversarios inyecten de forma remota código ejecutable en los

Dispositivos Periféricos. Si el atacante tiene una posición privilegiada en la red, podría afectar a miles de Dispositivos Periféricos a la vez. El resultado del ataque puede variar desde una simple ejecución de código hasta la denegación de servicio (bloqueando los Dispositivos Periféricos) o la alteración total de la función asignada al Dispositivo Periférico.

7.6 Autenticación Mutua Mal Diseñada o Sin Implementar

En los entornos de comunicación, los pares se hablan entre sí a través de protocolos similares de identidad. Esto significa cosas diferentes en diferentes contextos, pero en cada entorno una dirección de algún tipo identifica al destino de un mensaje. Cualquier módulo de comunicaciones que implemente un protocolo determinado puede indicar que es el propietario de una dirección particular. Incluso si una implementación particular de un protocolo está diseñada, u obligada, a usar la dirección de hardware de un módulo de radio local, no existe una regla que establezca que un usuario puede alterar físicamente la EEPROM de ese módulo y cambiar la dirección de hardware concreta. Incluso si la implementación se niega a permitir que un usuario cambie la dirección hardware de forma dinámica, aún puede ser manipulada para cambiar la dirección. El resultado de esta funcionalidad es, esencialmente, la falsificación: o el acto de tomar la identidad de otra computadora con el propósito de interceptar los mensajes destinados a esa computadora.

7.6.1 Autenticación de Clientes

Todos los entornos son vulnerables a la suplantación. Por ejemplo, cualquier modem celular puede indicar que es el propietario de cualquier identidad internacional de suscriptor móvil (IMSI), ya sea verdadera o no. Cualquier computadora portátil puede cambiar su dirección Ethernet, haciéndose pasar por otras computadoras en la red de área local (LAN). Independientemente de la tecnología de red empleada, sea cableada, RF o celular, la identidad de un Dispositivo Periférico utilizada para la comunicación puede ser suplantada.

La protección contra esto es la autenticación. Por ejemplo, en la red celular, cualquier persona con el equipo adecuado puede pretender poseer cualquier IMSI que elija. Sin embargo, los operadores celulares imponen la autenticación codificando una clave criptográfica en la SIM que es única para ese suscriptor (IMSI). Cuando un dispositivo celular se comunica con una estación base que indica que está representando un IMSI particular, la estación base emitirá un desafío criptográfico que solo puede ser resuelto por alguien con la clave criptográfica única almacenada en la tarjeta SIM aprovisionada para esa identidad en particular. Si el atacante no puede resolver el desafío criptográfico, la estación base puede verificar que el atacante no representa el IMSI en cuestión y puede impedir que el usuario se conecte a la red.

El modelo arriba descrito representa la autenticación basada en el cliente. Este es el modelo donde el subsistema del servidor (incluidas las estaciones base) permite a los clientes (Dispositivos Periféricos) unirse y abandonar la red siempre que los clientes puedan autentificar criptográficamente su identidad. Sin embargo, existe un problema a la inversa que expone a los clientes a la manipulación: la autenticación del servidor.

7.6.2 Autenticación de Servidores

En el modelo 3GPP, solo se autentican los Dispositivos Periféricos (denominados Equipos de usuario en 3GPP). Los Dispositivos Periféricos no autentican las estaciones base a las

que se conectan. Por lo tanto, cualquier estación base puede reclamar dar servicio en nombre de cualquier proveedor de telefonía celular. Las personas capaces de manipular o construir una estación base celular pueden suplantar a cualquier proveedor de telefonía móvil. Una estación base celular personalizada cuesta actualmente menos de 1,000 USD para su implementación, pero el resultado de dicho procedimiento solo permite la interceptación de mensajes en un área reducida, local. Una vez que se construye la estación base falsa, se puede hacer pasar por un proveedor local de telefonía celular e interceptar llamadas telefónicas, mensajes de texto e incluso datos provenientes de Dispositivos Periféricos localizados en el entorno de la base.

Los protocolos de red 3GPP más nuevos, como UMTS y LTE, imponen la autenticación mutua de ambas entidades. Esto permite que los Dispositivos Periféricos verifiquen criptográficamente que la estación base a la que se conectan está efectivamente dando servicio en nombre del operador Celular que dice ser. Un adversario debe ahora hackear la criptografía del operador celular para hacerse pasar por una estación base, lo que aumenta significativamente la complejidad, la dificultad y el costo de un ataque.

7.6.3 Interceptores Celulares o Estaciones Base Falsas

Sin embargo, existen excepciones a esta regla, como los interceptores celulares. Estos dispositivos, generalmente utilizados por los contratistas del gobierno, los gobiernos y los servicios de inteligencia, están codificados con claves criptográficas proporcionadas a estas entidades por ciertos operadores de telefonía móvil, con el fin de utilizarse para la seguridad pública nacional. Estos sistemas utilizan estas claves para interceptar de forma pasiva las comunicaciones bidireccionales o para realizar activamente ataques de “hombre en el medio” (MITM) contra objetivos específicos.

Sin embargo, en el modelo de amenaza de las comunicaciones modernas, el acceso a esta tecnología no se limita a los gobiernos y agencias de inteligencia. Hoy en día, estos sistemas pueden construirse a partir de piezas que cuestan solo varios cientos de dólares, lo que resulta en una estación base falsa rentable capaz de interceptar o suplantar las comunicaciones celulares.

7.6.4 La Seguridad en las Comunicaciones es una Seguridad Puerta a Puerta

La aparición en el mercado celular de los interceptores ayuda a resumir esta sección de forma bastante adecuada, al manejar la idea de que la seguridad de las comunicaciones no es absoluta. Solo protege el canal de comunicación entre dos entidades. Estas entidades, sin embargo, actúan como puertas que permiten que los datos entren y salgan de los ecosistemas a los que están conectadas.

Por ejemplo, una tarjeta SIM particular puede aprovisionarse para su uso en un sistema de control industrial tal como un dispositivo de control de un pozo de petróleo. Una tarjeta SIM, por diseño, es un componente extraíble. Cualquier persona con acceso físico al dispositivo de supervisión del pozo de petróleo puede extraer la tarjeta SIM y colocarla en una computadora portátil. Si la computadora portátil tiene un software que puede simular la funcionalidad del dispositivo de control específico, el servidor del back-end no podrá diferenciar entre el dispositivo real en el pozo y la computadora portátil. Sin embargo, la

computadora portátil se autenticará en la red celular gracias a la tarjeta SIM. Por lo tanto, la red celular ha autenticado la tarjeta SIM, pero no la computadora portátil.

7.6.5 Resolviendo la Autenticación Mutua

Cada par en un ecosistema de IoT debe autenticar a todos los demás pares que participan en ese ecosistema. Para lograr esto, se debe utilizar una TCB para garantizar que la arquitectura criptográfica adecuada se esté empleando para garantizar las comunicaciones y la tecnología inherente. La autenticación mutua no puede ocurrir si las claves se exponen fácilmente a los atacantes. Revise la sección de las TCB en este documento para más información.

Una vez autenticado, cada par debe encriptar y firmar los mensajes enviados a otros pares en la red. Cada par que recibe un mensaje debe validar criptográficamente los datos antes de actuar sobre él. Dado que no todos los protocolos de comunicación son capaces de autenticación mutua, o poseen una fuerte criptografía, es imperativo que el ingeniero de aplicaciones diseñe un protocolo con las características deseadas para que imponga confidencialidad e integridad, en lugar de depender del protocolo de comunicaciones.

Incluso los protocolos más robustos que incorporan la autenticación mutua, como LTE, no abordan la seguridad de la infraestructura más allá de la red de comunicaciones celular. Solo la seguridad del protocolo en una capa superior puede abordar el riesgo de las debilidades en la infraestructura más allá del control del proveedor de telefonía móvil.

7.6.6 Riesgo

El riesgo de no implementar una solución de seguridad para la aplicación robusta es que el Dispositivo Periférico deberá confiar la seguridad a la capa de comunicaciones. Como se describe en esta recomendación, puede que no sea adecuado confiar únicamente en la red para resolver los problemas de seguridad en la aplicación. Incluso si se puede confiar en el operador, los mensajes pueden pasar a través de varios elementos de la infraestructura de red que no pertenecen ni están controladas por el operador antes de que los datos lleguen a los servidores propiedad del proveedor de servicios de IoT. Por lo tanto, el proveedor de servicios de IoT se pone en riesgo con cualquiera que tenga el control de los sistemas que interceptan, alteran o componen mensajes hacia o desde los Dispositivos Periféricos.

7.7 Gestión de la Privacidad

El aspecto más característico en la tecnología IoT es su capacidad para conectar el mundo físico con el mundo digital. El resultado de esto es una brecha en la privacidad, ya que el entorno físico del usuario está directamente asociado con las cosas que le gustan y con lo que ve al navegar por Internet. Esto puede causar efectos indeseables en el tiempo.

Como resultado, es importante que los proveedores de servicios de IoT consideren la privacidad de sus usuarios y desarrollen interfaces de gestión de la privacidad que estén integradas tanto en el Dispositivo Periférico, cuando sea posible, como en la interfaz web del producto o servicio.

Esta tecnología debería permitir al usuario determinar qué atributos de su privacidad están siendo utilizados por el sistema, cuáles son los términos del servicio y la posibilidad de desactivar la exposición de esta información a la empresa o a sus socios. Este sistema con

la granularidad adecuada y la exclusión voluntaria ayudará a garantizar que los usuarios tengan el derecho y la capacidad de controlar la información que comparten sobre sí mismos y sobre su mundo físico.

7.7.1 Riesgo

Los riesgos potenciales de no proteger la privacidad del consumidor son muchos. Los problemas de persecución, acoso, caracterización del perfil, amenazas y más, son posibles efectos realistas y prácticos de no proteger los datos del usuario.

7.8 Privacidad e Identidades Únicas para los Dispositivos Periféricos

Cada Dispositivo Periférico es conocido digitalmente por su “huella digital”. Esta huella digital se compone de direcciones, números de serie e identidades criptográficas que son exclusivas del Dispositivo Periférico específico. Sin embargo, estos tokens también pueden asociar directamente un dispositivo a un cliente, ubicación o servicio en particular. En muchas situaciones, esto no es lo que se pretende. Por ejemplo, los smartphones se pueden rastrear porque la dirección wifi incorporada y asignada al teléfono se utilizó cuando se buscaban activamente puntos de acceso 802.11. Estas direcciones se pueden rastrear a medida que viajan de un lugar a otro. Esto permitiría a cualquier persona capaz de asociar una determinada dirección de wifi con un usuario en particular, ver sus movimientos en todo el mundo. Para combatir esto, los fabricantes de software para smartphones están generando ahora direcciones aleatorias para los clientes wifi de los teléfonos cuando escanean puntos de acceso, por lo que es casi imposible rastrear los teléfonos de esta manera.

Los Dispositivos Periféricos IoT se pueden rastrear de una manera similar a través de las direcciones Bluetooth Low Energy (BLE), direcciones 802.15.4, wifi o incluso IMSI celular. Siempre que sea posible, el Proveedor de Servicios IoT debe desarrollar su Dispositivo Periférico de tal manera que use una dirección de radio aleatoria para conectarse a entornos nuevos, permitiendo que la privacidad del usuario permanezca intacta.

Esto también es cierto para las claves criptográficas, como las claves públicas SSH. Si bien los usuarios generalmente desean que sus claves públicas sean conocidas por el público en general, las claves públicas criptográficas en Dispositivos Periféricos expondrán la identidad del usuario de un Dispositivo Periférico particular, lo cual no es deseable. En cambio, el usuario debería poder seleccionar la opción de difundir su identidad cuando se conecta a un nuevo entorno.

7.8.1 Riesgo

Si no se evita adecuadamente este riesgo, se podrá rastrear a los usuarios con Dispositivos Periféricos móviles a medida que estos se unan y dejen las redes. Esto abre brechas importantes sobre la privacidad que los equipos legales, los legisladores e incluso las compañías de seguros están analizando actualmente. No implementar adecuadamente el mantenimiento de la privacidad para disminuir el potencial de localización puede causar que un nuevo proveedor de servicios de IoT se enfrente a consecuencias legales en un futuro cercano.

7.9 Ejecutar las Aplicaciones con Niveles de Privilegio Apropiados

Las aplicaciones que se ejecutan en un Dispositivo Periférico generalmente no requieren privilegios de super-usuario. Muy a menudo, las aplicaciones requieren acceso a controladores de dispositivos o a un puerto de red. Si bien algunos de estos dispositivos, puertos u otros objetos pueden requerir privilegios de super-usuario para acceder inicialmente a ellos, los privilegios de super-usuario no son necesarios para realizar operaciones posteriores. Por lo tanto, es una buena práctica utilizar privilegios de super-usuario al inicio de la aplicación para obtener acceso a estos recursos. A partir de ahí, los privilegios de super-usuario deberían ser eliminados.

Quitar privilegios de super-usuario es un proceso común que está bien documentado y se ha implementado excepcionalmente bien en aplicaciones como Secure Shell (SSH), Apache2 y otros servidores bien diseñados. El proceso generalmente abarca:

- Iniciar la aplicación con privilegios altos
- Acceder a todos los recursos que requieren privilegios determinados
- Identificación de una identidad de usuario (por ejemplo, ID de usuario de UNIX e ID de grupo) que la aplicación debe ejecutar como si lo fuera
- Reemplazar la identidad del proceso con la ID de usuario/grupo objetivo, eliminando así los privilegios de super-usuario de la aplicación en ejecución

Se puede ver un modelo más complejo en la implementación de SSH de *privsep*, que ejecuta un servicio privilegiado cuyo único propósito es iniciar la aplicación principal bajo una identidad de usuario/grupo objetivo. De esta forma, si el servicio se cierra, se puede reiniciar fácilmente sin el compromiso de recursos privilegiados.

Para obtener más información, consulte: “SSH Privilege Separation”: http://www.citi.umich.edu/u/provos/ssh/privsep.html

7.9.1 Riesgo

La ejecución de aplicaciones con niveles de privilegios elevados puede resultar en un compromiso total del sistema si una sola aplicación se ve comprometida. Como los privilegios de super-usuario otorgan a una aplicación acceso total a todo el sistema en ejecución, no hay manera de contener a un hacker una vez que accede fraudulentamente a dicha aplicación. Disminuir o anular los privilegios ayuda a contener al adversario y limita su capacidad de incrementar sus privilegios dentro del sistema embebido. Esta puede ser la diferencia entre un compromiso total del sistema y una pequeña “molestia”.

7.10 Hacer Cumplir la Separación de Funciones en la Arquitectura de las Aplicaciones

Las aplicaciones que se ejecutan en un Dispositivo Periférico deben tener diferentes identidades de usuario asociadas con cada proceso único. Esto garantiza que, si una aplicación se ve comprometida, una aplicación separada en el mismo Dispositivo Periférico no se verá comprometida a menos que un segundo ataque orientado a esta segunda aplicación tenga éxito. Este paso adicional requerido en nombre de un atacante suele ser un obstáculo crítico para el proceso general de desarrollo de ataques y aumenta el costo y la complejidad de estos contra un Dispositivo Periférico.

Por ejemplo, un servicio de red que permite a un usuario recuperar información sobre el estado de un Dispositivo Periférico no debe poder también manipular la TCB a través de este mismo proceso. Esa capacidad estaría fuera de lugar en relación con el propósito del servicio original. Estas dos operaciones distintas deben manejarse en aplicaciones separadas y ejecutarse bajo identificadores de usuario separados en el sistema operativo local, lo que ayuda a separar las funciones de la aplicación y reduce el riesgo de abuso si un componente se ve comprometido.

Para implementar esto correctamente, la protección de memoria debe estar habilitada en la arquitectura de hardware subyacente, y el sistema operativo debe adoptar el concepto de niveles de privilegios. Un software/aplicación sin privilegios debe tener acceso restringido a recursos privilegiados, como controladores, archivos de configuración u otros objetos.

Los servicios deben realizar solicitudes para acceder a los recursos con privilegios, pero a través de una API restringida, como una llamada al sistema, para garantizar que todos los mensajes estén bien formados y que se ajusten a los requisitos de la arquitectura de seguridad.

El concepto de varios niveles de privilegios es un concepto muy antiguo. Sin embargo, en los sistemas embebidos, a menudo se pasa por alto esta funcionalidad ya que los usuarios no pueden iniciar sesión en la consola y ejecutar sus propias aplicaciones. Como resultado, todos los servicios a menudo se despliegan para usuarios con privilegios determinados. Sin embargo, esto no es aconsejable.

Cada aplicación o servicio debe implementarse utilizando un privilegio determinado y personalizado. En la mayoría de los entornos, esta se traduce en una identidad de usuario separada. Esta separación de tareas mediante la definición de diferentes identidades de usuario garantiza que si un servicio está en peligro no puede afectar directamente los recursos utilizados por otro servicio en el mismo sistema. Para comprometer otros servicios y usuarios, los ataques secundarios se deben enfocar al sistema operativo local para alterar los privilegios.

Esto requiere una planificación y una arquitectura de aplicaciones adecuada que utilice la separación de privilegios correctamente.

7.10.1 Riesgo

Si no se aplica una separación de funciones, cualquier servicio que se llegue a comprometer en el Dispositivo Periférico puede comprometer a todos los demás servicios en el dispositivo, ya que estos al ejecutarse compartirán la misma identidad de usuario y/o grupo. Si se implementa la recomendación, un servicio con pocos privilegios que se vea comprometido a través de la red no resultará en un compromiso inmediato de todo el sistema.

Debido a que esta recomendación es fácil de implementar, es fundamental para la seguridad de los Dispositivos Periféricos IoT. Cabe señalar que a menudo se necesita una experiencia contrastada para comprometer de forma remota un servicio de red. Si también se requiere que el atacante eleve los privilegios implementando un ataque a nivel del kernel,

u otro ataque secundario, para obtener el control del sistema completo, el atacante puede no tener el tiempo, las habilidades o el equipo para poder ejecutar el ataque.

Aumentar la dificultad de un ataque con un simple cambio de configuración como este, contribuirá en gran medida a garantizar la longevidad del dispositivo.

Además, como los servicios comprometidos se pueden detectar a través de la supervisión de procesos y otros análisis, cualquier compromiso del servicio puede alertar al Ecosistema de Servicios IoT de que se ha detectado un compromiso. Esto permite a los administradores del sistema actuar para asegurarlo antes de que se haya logrado un compromiso global. Esto también permite a los administradores diagnosticar y parchear el software vulnerable antes que se ataque al sistema globalmente aprovechando una vulnerabilidad en particular. Esto le da al negocio una ventaja significativa incluso contra hackers expertos.

7.11 Hacer Cumplir la Seguridad de los Lenguajes de Programación

Los lenguajes de programación tienen diversos grados de seguridad, dependiendo del propósito del lenguaje y de su nivel (p. ej. ensamblador o Java). Algunos lenguajes proporcionan construcciones para limitar el acceso a los registros de la memoria directamente e imponen restricciones sobre cómo se debe de usar. El equipo de ingeniería debe identificar un lenguaje que sea capaz de proporcionar seguridad durante la ejecución de la aplicación o sobre el equivalente binario resultante.

Se deberían proporcionar funciones de seguridad durante el tiempo de ejecución o dentro del mismo compilador, cuando sea posible, para restringir el potencial de abuso de una vulnerabilidad por parte del atacante. En un entorno de ejecución bien definido, incluso si un fallo de programación es fácil de desencadenar, puede ser extremadamente difícil de lograr un compromiso global de sistema. Esto supone que las mejoras de seguridad se utilizan para proteger la forma en que se ejecuta la aplicación, se accede a la memoria y es compatible con las mejoras de seguridad del sistema operativo.

7.11.1 Riesgo

El riesgo de no reforzar la seguridad en los lenguajes de programación y en la aplicación resultante puede resultar en un compromiso le la aplicación fácil de implementar. Algunos sistemas de programación como PHP son especialmente poco fiables y nunca deben ser utilizados por un equipo de ingeniería profesional. Otros lenguajes, como Python, son adecuados para entornos de producción, pero tienen sutilezas para implementar entornos seguros que pueden poner en riesgo un sistema y tienen que evaluarse cuidadosamente. Por lo tanto, el riesgo resultante puede variar considerablemente. El equipo de ingeniería debe usar el proceso de evaluación de riesgos y modelado de amenazas para evaluar de manera adecuada qué lenguaje es el mejor para su entorno de producción.

7.12 Implementar Auditorias de Seguridad Tipo “Pentesting” (Pruebas de Penetración) Persistente

Realizar una auditoría de seguridad solo en el momento del despliegue no es suficiente para la mayoría de las implementaciones de sistemas IoT donde los Dispositivos Periféricos nuevos se pueden desplegar y configurar en cualquier momento. Se recomienda utilizar un

enfoque de “pentesting” persistente para lograr la detección temprana de software vulnerable en los Dispositivos Periféricos y en sus configuraciones no seguras.

Implementar una estrategia como esta puede proporcionar una detección rápida y una administración temprana de las amenazas identificadas, lo que aumenta la velocidad de resolución de problemas y reduce el período de exposición a amenazas.

Una estrategia completa de pruebas de penetración persistente debe proporcionar una forma automática y programada de realizar: descubrimiento de activos para crear un inventario, identificación y análisis de estos activos, verificación y explotación de vulnerabilidades conocidas, verificación de configuraciones inseguras e informes y alertas adecuados que deberían ayudar con la corrección de posibles problemas de seguridad.

7.12.1 Riesgo

El riesgo de no implementar una estrategia de Pruebas de Penetración persistente es que las auditorías de seguridad solo se puedan ejecutar una vez durante el despliegue, pero nunca se evaluarían en este caso nuevos Dispositivos Periféricos y configuraciones que se agreguen al sistema IoT. Esta situación puede conducir a un conjunto de Dispositivos Periféricos vulnerables que nunca se identifican como expuestos hasta que sean atacados por un hacker.

8 Recomendaciones de Prioridad Media

El conjunto de recomendaciones de prioridad media abarca el conjunto de recomendaciones que son relevantes dependiendo de las opciones de diseño de los Dispositivos Periféricos. Por ejemplo, la utilización de mejoras de seguridad a nivel del sistema operativo solo es válida si hay un sistema operativo ejecutándose en el Dispositivo Periférico. Si el Dispositivo Periférico está compuesto por una aplicación de kernel monolítico o un Sistema Operativo en Tiempo Real (RTOS) integrado con una sola aplicación embebida, la recomendación puede no ser aplicable. Cuando las recomendaciones sean relevantes para el diseño de los Dispositivos Periféricos, se deben implementar.

8.1 Imponer las Mejoras sobre el Nivel de Seguridad en los Sistemas Operativos

Las aplicaciones que se ejecutan en un sistema operativo deben diseñarse para utilizar (ya sea de forma transparente o deliberada) las mejoras de seguridad del sistema operativo subyacente y kernel. Esto incluye tecnologías tales como:

- ASLR ("Address Space Layout Randomization”)
- Memoria no ejecutable (“Stack”, estructuras en árbol, BSS, ROData, etc.)
- Protección contra la perdida de referencias del puntero del usuario (UDEREF)
- Protección contra la fuga de estructura (divulgación de la información)

Cada sistema operativo utilizado en un sistema embebido proporcionará diferentes variaciones y combinaciones de estas tecnologías, a veces bajo diferentes nombres. Determine qué es lo que el sistema operativo y el núcleo pueden proporcionar, y habilite estas tecnologías, cuando sea posible, para mejorar la seguridad de las aplicaciones.

El problema consiste en identificar de qué es capaz cada sistema operativo. Por ejemplo, las aplicaciones que se ejecutan en plataformas que no tienen una unidad de administración de memoria (MMU) pueden no ser capaces de ASLR. Sin embargo, el equivalente de UDEREF se puede aplicar incluso en entornos con solo una unidad de protección de memoria (MPU). Evalúe qué tecnología se utiliza y sus capacidades, y determine qué nivel de seguridad se puede lograr mediante la combinación de arquitectura, kernel, sistema operativo y protección de aplicaciones.

8.1.1 Riesgo

No aplicar esta recomendación dará como resultado un entorno de tiempo de ejecución de la aplicación que es sustancialmente más fácil de explotar. Estas mejoras limitarán significativamente el número de atacantes que sean capaces (si es que lo hacen) de desarrollar un ataque fiable para un servicio vulnerable.

Por lo tanto, si una aplicación desarrollada por la empresa tiene fallos de seguridad que podrían explotarse para obtener capacidades de ejecución remota de código, esto puede evitarse al aplicar ASLR, NX, UDEREF y otras tecnologías. Esto limitará la capacidad de un atacante para desarrollar un ataque en un período de tiempo razonable, ya que se requerirá que el atacante utilice técnicas avanzadas y complicadas que deben personalizarse para cada objetivo concreto. Esto aumenta no solo la dificultad, sino también el tiempo y los gastos necesarios para lograr un ataque totalmente funcional.

Sin estas mejoras, se puede desarrollar un ataque con éxito utilizando software disponible en la web, gratis y disponible en pocas horas.

8.2 Deshabilitar las Tecnologías de Depuración y Pruebas

Cuando se desarrolla un producto, a menudo se habilita con tecnologías de depuración y prueba para facilitar el proceso de diseño e implementación. Esto es completamente normal. Sin embargo, cuando un dispositivo está listo para el despliegue en producción, estas tecnologías deben eliminarse del entorno de producción antes que se defina una configuración aprobada para su comercialización.

La configuración aprobada con la que se despliega un producto nunca debe contener interfaces de depuración, diagnóstico o prueba que puedan ser utilizadas por un atacante. Tales interfaces son:

- Interfaces de consola de línea de comandos
- Consolas con depuración detallada, diagnóstico o mensajes de error
- Puertos de depuración de hardware como JTAG o SWD
- Servicios de red utilizados para la depuración, el diagnóstico o las pruebas
- Interfaces administrativas, como SSH o Telnet

Todas esas tecnologías deberían estar deshabilitadas en la configuración aprobada del sistema.

Los puertos serie que el sistema puede eliminar también se deben eliminar físicamente de la PCI. Sin embargo, muchas veces los puertos serie de las UART / USART se habilitan a través de pines físicos del microcontrolador o procesador. Si estos pines todavía están

habilitados como una consola de acceso, un atacante simplemente puede puentear (conectarse a) los pines para interactuar con la consola. Eliminar el puerto serie sobre la PCI, como una interfaz DB9, no deshabilita la consola en sí.

Además, los puertos de depuración como JTAG y SWD no deberían desactivarse solamente a través del software. Estos dispositivos se deben inhabilitar mediante la alteración de los fusibles o seguros hardware. Deshabilitar estas características por software ofrece una ventana de oportunidad para que un atacante se conecte a los puertos JTAG, SWD o una interfaz de depuración de hardware similar antes de que el software desactive la interfaz. Esta ventana de oportunidad a menudo es suficiente para que un atacante logre su cometido.

8.2.1 Riesgo

Sin implementar esta recomendación, las empresas se exponen a la extracción de secretos críticos de la unidad de procesamiento central (CPU). Esto puede permitir que los atacantes carguen su propio firmware en la NVRAM o la EEPROM, y les permita extraer o alterar secretos críticos que comprometan además la red del sistema IoT o sus dispositivos.

La desactivación de los puertos de depuración es un paso fundamental para garantizar la integridad del producto o servicio de IoT. Sin embargo, es importante que la empresa evalúe el riesgo de inhabilitar estas tecnologías y sopesarlas con el beneficio de poder diagnosticar y depurar los problemas identificados sobre el terreno. Puede ser significativamente más difícil reparar fallos a nivel de producción de un producto si no hay forma de depurar su sistema de ejecución.

8.3 Corrupción del Contenido de la Memoria a través de Ataques a los Periféricos

Los sistemas de ejecución de programas se basan en la coherencia para garantizar que el resultado de los algoritmos sea predecible con respecto a un conjunto de datos de entrada determinados. Los sistemas de ejecución también esperan que los componentes funcionen de manera fiable, y que para cada bit escrito, ese bit es estable y no se modifica hasta que el procesador lo cambie. Dentro de los sistemas cerrados, esta teoría es aplicable. Cuando se producen anomalías en este modelo, pueden comprometer, o simplemente dañar, un entorno de ejecución.

La seguridad de la información presenta una clase de anomalías intencionalmente provocadas para obtener acceso a objetos que de otro modo serían inaccesibles. Una ventana de oportunidad que daría la posibilidad a un atacante de inducir un comportamiento anómalo en su beneficio es el acceso directo a la memoria (DMA). En pocas palabras, DMA es una tecnología que los procesadores pueden usar para permitir que los componentes externos (periféricos) tengan acceso a la memoria del procesador principal sin interferencia de la CPU. Por lo tanto, la CPU puede otorgar un acceso directo a un dispositivo periférico a una región de la memoria. Este periférico puede leer o escribir en esa región de la memoria.

Si el procesador no restringe adecuadamente la región de memoria que puede utilizar el periférico, este puede tener acceso a más memoria principal de la requerida para la funcionalidad prevista. De esta manera, si el periférico (por ejemplo, un controlador Ethernet) tiene asignada una región DMA destinada a ser utilizada como un búfer circular

para las tramas Ethernet recibidas, y la región DMA asignada comprende toda la extensión de la memoria principal, el firmware en el controlador Ethernet ahora puede leer y escribir arbitrariamente en toda la memoria del sistema. La CPU no tendrá forma de bloquear el firmware del controlador Ethernet para que no escriba en la memoria.

El resultado de este ataque es doble. Los datos se pueden extraer de la memoria principal y codificarse en paquetes de red o información de aplicación para su aprovechamiento encubierto o inmediato. Alternativamente, un atacante podría insertar secretamente una puerta trasera (malware) en la memoria principal sobrescribiendo el código ejecutable de una aplicación.

Desde la perspectiva del procesador, es poco lo que puede hacer para identificar si una ventana o zona de memoria excesivamente permisiva ha sido corrompida por un dispositivo periférico malicioso. Para combatir este ataque, identifique si el procesador utilizado en el Dispositivo Periférico es capaz de restringir la funcionalidad de la DMA a regiones de memoria pequeñas y predecibles. De ser así, asegúrese de que cada región de memoria esté definida para cada Dispositivo Periférico que lo requiera. No habilite los espacios de memoria de manera arbitraria, cuando sea posible, para ser usados por Dispositivos Periféricos.

Algunos procesadores no pueden permitir la restricción pormenorizada en el tamaño o la ubicación en la memoria lineal o virtual de una ventana de DMA. Como los ataques a través de DMA se deben considerar una amenaza realista en los Dispositivos Periféricos de IoT para aplicaciones críticas, evalúe si tiene sentido considerar un procesador alternativo que permita estas restricciones sobre DMA.

Para las plataformas que exponen puertos como en el estándar IEEE1394, Thunderbolt, Express Card u otros puertos que permiten el acceso directo a DMA para la Interconexión a Componentes Periféricos (Peripheral Component Interconnect, PCI), existen ataques “enlatados” y rentables accesibles en la web.

Para las plataformas donde un ataque basado en DMA requiere el ataque a un componente de hardware local, la dificultad ciertamente aumentará, pero no está fuera del alcance de un compromiso de seguridad basado en la posibilidad de regrabar el firmware de este periférico para modificar el comportamiento de la DMA para comprometer un Dispositivo Periférico local. Sin embargo, el costo, el tiempo y la experiencia serán un factor que hará que el actor en este caso sea un hacker patrocinado (pagado).

8.3.1 Riesgo

No restringir la posibilidad de que los componentes externos alteren la DMA puede someter a la plataforma a un compromiso general o, al menos, a la extracción de secretos clave, datos centrados en la privacidad o propiedad intelectual del Dispositivo Periférico.

8.4 Seguridad de la Interfaz de Usuario

Los Dispositivos Periféricos IoT que tienen interfaces de usuario, como pantallas táctiles, pantallas ricas en información o tecnologías de interfaz alternativas, deben ser capaces de presentar información al usuario y capturar información de un usuario de manera segura.

Si bien los atributos de la interfaz de usuario, como las contraseñas, ya se han cubierto en este documento, hay algunos problemas más sutiles que deben discutirse:

- Sistemas de alerta
- Confirmación de las acciones

Cuando se produce una anomalía, como una manipulación física o una aplicación que se comporta de forma rara, el usuario debe recibir una alerta visible. Alternativamente, el usuario debería poder revisar las alertas del sistema desde la interfaz de usuario.

Además, todas las funciones realizadas por el dispositivo que son causadas por código embebido en la interfaz o transiciones automáticas de una interfaz a otra deben ser confirmadas por el usuario. Un ejemplo de esto es si la cámara del dispositivo lee un código QR, o una interacción con dispositivos NFC o RFID solicitará que el sistema se conecte a una URL. En estos casos, se debe solicitar al usuario que confirme la acción y valide que la acción que se tiene que realizar es deseable. El usuario debe tener la opción de cancelar la acción. El usuario debería poder ver todos los detalles sobre la acción determinada, incluida la URL completa a la que se conectaría.

8.4.1 Riesgo

Si no se implementa esta recomendación, los usuarios serán vulnerables a ataques que no se pueden detectar. Mientras que algunos diseñadores de sistemas aprecian la fluidez de la transición de la lectura de un dispositivo RFID a, por ejemplo, el sitio web del producto correspondiente. Ciertamente puede haber efectos indeseables en este comportamiento. Los usuarios podrían verse obligados a ver contenidos no deseados sin su consentimiento, o los usuarios podrían ser engañados para visitar sitios web o realizar acciones que debiliten su situación con respecto a la seguridad o privacidad.

Además, los usuarios que tienen dificultades para comprender los mensajes de alerta pueden no entender los riesgos de usar un dispositivo potencialmente manipulado. Esto puede disminuir incluso la seguridad física del usuario y podría ponerlos en peligro.

8.5 Auditorías de Código Externas

Cada vez que una sección de código importante se implemente, como un gestor de arranque, al ser un componente crítico en la construcción de una plataforma segura en tiempo real, debe ser auditada para detectar los riesgos. Si un atacante puede manipular un gestor de arranque para que ejecute código fraudulento o para eludir la secuencia de autenticación, este código al final se vuelve inútil. Esto perjudicaría el presupuesto, el tiempo y la experiencia utilizados por la empresa en el despliegue de esta tecnología, malgastando los gastos en ingeniería incurridos hasta la fecha.

Una brecha en la seguridad en esta área también puede resultar en una ventaja para un competidor de la empresa a través de la suplantación de identidad, abusos de API, interceptación de datos, clonación de dispositivos e incluso cambio de marca del dispositivo. Por lo tanto, es imperativo que las secciones críticas de código sean auditadas por alguien o una empresa reconocida en este ámbito, para garantizar que la tecnología no corra riesgo de ser hackeada. Por lo tanto, para encontrar un equipo experto en la seguridad de la información adecuado para realizar la auditoría, evalúe qué tipo de código y lenguaje será

auditado. Típicamente, en este modelo nos referimos a: C, ensamblador y posiblemente C++ o Java.

Identifique un equipo que tenga experiencia en estos lenguajes, así como en la arquitectura utilizada. Mientras que muchos equipos de seguridad de la información realizan auditorías de código fuente, no muchos de ellos pueden realizar auditorías en la plataforma particular utilizada por el negocio de IoT. Cada plataforma tiene diferencias sutiles, y lo mejor es utilizar un equipo familiarizado con la plataforma que se utiliza.

8.5.1 Riesgo

Si bien la contratación de consultores externos para evaluar la tecnología desarrollada internamente puede ser difícil, es un requisito de seguridad. Esto se debe a que los ingenieros que desarrollan tecnología deben ser capaces de demostrar que su arquitectura es testeable. Esto es difícil de hacer si los ingenieros que desarrollan la arquitectura son los únicos que lo revisan. Los ingenieros tienden a visualizar su código base a partir de la arquitectura que han intentado diseñar e implementar, no desde la perspectiva de la implementación real deseada. Por lo tanto, a menudo se necesitan “ojos” externos para encontrar detalles en la arquitectura y la implementación que podrían causar lagunas en la seguridad.

8.6 Utilizar un APN Privado

En las redes celulares 3GPP, un APN actúa como una red privada configurada específicamente para un conjunto de dispositivos autenticados. Por lo general, un APN privado (también llamado "APN seguro") es una red privada a la que solo pueden acceder los dispositivos autenticados asociados con un negocio específico. Al utilizar un APN, las empresas pueden restringir qué Dispositivos Periféricos pueden conectarse a su infraestructura de servicio a través de la red celular. Esto ayuda a reducir la cantidad de usuarios que tienen acceso directo a los servicios de IoT en la infraestructura del back-end.

Otros atributos de una APN privada pueden ayudar a disminuir el potencial de que dispositivos periféricos deshonestos puedan abusar del ecosistema de IoT. Los cortafuegos pueden limitar a qué servicios o computadoras se pueden conectar desde la APN. Un APN bien configurado impedirá que Dispositivos Periféricos establezca conexiones directas entre sí, lo que no permite que un Dispositivo Periférico afectado migre (e infecte) horizontalmente a través de la infraestructura de red a otros Dispositivos.

Analice con el operador de telefonía celular o el operador de red móvil virtual (MVNO) con el que la empresa está trabajando para determinar qué tecnologías están disponibles en la utilización de un APN seguro. Otros servicios como la supervisión, la inclusión de dispositivos anómalos en listas negras y la vinculación de las identidades del usuario a distintas funciones, pueden estar disponibles.

8.6.1 Riesgo

Utilizar un APN privado puede restringir muchos tipos de ataques. Por ejemplo, los APN privados permiten que la empresa reduzca la cantidad de conexiones que se pueden realizar desde el Dispositivo Periférico directamente a Internet. Los Dispositivos Periféricos nunca deberían tener permitido conectarse directamente a recursos no fiables en Internet.

Solo las organizaciones asociadas deben ser admitidas, y esos servicios deben ser autenticados.

Sin el uso de un APN privado, los Dispositivos Periféricos comprometidos pueden comunicarse sin restricciones con cualquier servicio de Internet o protocolo. Esto puede permitir que un adversario abuse del Dispositivo Periférico para lanzar un ataque secundario en una infraestructura separada. Esto podría implicar un ataque de denegación de servicio (DoS) o podría ayudar a facilitar un ataque más peligroso contra otra empresa, gobierno o persona física.

Es notable, sin embargo, que un APN privado no reduce el riesgo de que un atacante comprometa el enlace de comunicaciones entre el Dispositivo Periférico y el APN privado. Además, el APN privado solo actúa como una puerta de entrada a los servicios de back-end, y no impone ninguna seguridad entre el APN y los servicios de back-end en la red privada del proveedor de servicios de IoT. Estas posibles vulnerabilidades en la seguridad deben abordarse por separado, independientemente de las mejoras que se otorgan mediante el uso de una APN privada.

8.7 Implementar Umbrales de Bloqueo Relativos a las Condiciones del Entorno (Ambientales)

Los componentes dentro de un sistema embebido están diseñados para ser utilizados dentro de ciertos umbrales ambientales de funcionamiento. Esto incluye niveles de voltaje, consumo de corriente, temperatura ambiente o de operación y humedad. Cada componente generalmente se clasifica para poder funcionar en un rango de valores determinado. Si el dispositivo está sujeto a condiciones cuyos parámetros están por encima o por debajo de ese rango, el componente puede funcionar de manera errónea o comportarse de una manera que sea útil para un atacante.

Por lo tanto, es importante detectar cambios en estas variables en los componentes para determinar si el dispositivo debe continuar funcionando o si debe apagarse. Sin embargo, debe tenerse en cuenta que la desconexión puede ser un efecto deseado y que el atacante puede aprovecharse de esta situación para provocar una denegación de servicio. El equipo de ingeniería debe evaluar este modelo para determinar si es mejor desconectar el dispositivo o mantenerlo en línea.

De todos modos, el modelo generalmente incorpora:

- Detección de interrupciones momentáneas de funcionamiento (“Brown-out”) y apagado, cuando el voltaje cae demasiado bajo
- Protección al circuito con un “techo de tensión” para garantizar que los niveles de voltaje no excedan un umbral
- Circuitos de limitación de corriente para garantizar que el consumo de corriente no pueda bajar o superar ciertos niveles
- Supervisión interna de la temperatura en la CPU, MCU y otros componentes que miden los niveles internos
- Opcionalmente, se pueden evaluar los niveles de humedad para determinar si el medio ambiente en el que opera el dispositivo está volviéndose demasiado húmedo o demasiado seco

La temperatura es extremadamente importante ya que las altas temperaturas pueden indicar un problema de circuito provocado por el usuario, el entorno o incluso un problema de hardware o software. El monitoreo de la temperatura permitirá que el sistema operativo o la aplicación desconecten los recursos (o todo el dispositivo) para garantizar que el Dispositivo Periférico no provoque un incendio u otro problema.

Los bajos niveles de temperatura también cambian el comportamiento de un dispositivo. Esto puede ralentizar un circuito o hacer que sus componentes reaccionen de forma inesperada. Esto puede ser útil para un atacante si la temperatura puede causar una anomalía predecible que afecte la aplicación o los circuitos de una manera que le pueda venir bien para sus propósitos.

La dificultad en los umbrales de bloqueo se manifiesta al analizar la temperatura y la humedad. Los niveles de voltaje y corriente deben ser controlados por circuitos de “Brown-out” y “Black-out” (intermitencia o falta de alimentación) en la PCI o en el procesador directamente. Dado que los ingenieros podrán buscar los niveles adecuados para el voltaje y los umbrales de corriente de un circuito integrado, pueden implementar fácilmente protecciones para estos problemas.

Para la temperatura y la humedad, la decisión de actuar es un poco más problemática ya que estos niveles pueden ser causados por acciones de un adversario sin tocar el dispositivo físicamente. En el caso de la temperatura, los niveles que pueden causar un problema de seguridad en el equipo con un alta probabilidad, deben hacer que el dispositivo tome las medidas adecuadas para bajar la temperatura. Sin embargo, en entornos críticos como sistemas de control industrial o dispositivos médicos, el dispositivo debe intentar continuar con las operaciones críticas, siempre que sea posible. Si los niveles superan un determinado umbral acordado por los ingenieros y directivos de la empresa, solo entonces debe apagarse el dispositivo.

8.7.1 Riesgo

Con respecto al consumo de voltaje y corriente, el riesgo de un ataque está relacionado con la inyección de fallos (“glitching”) y otros ataques de canal lateral que pueden beneficiarse de los cambios en estos niveles de consumo. Si se implementa la detección de cambios bruscos en la alimentación (“Brown-out” y “Black-out”) en el procesador, se reduce el riesgo de un ataque. De lo contrario, el riesgo está relacionado con los picos de voltaje o corriente que podrían causar problemas de seguridad con el dispositivo físico, o permitir que un atacante integre ataques tipo “glitching” (y similares) para burlar la seguridad de los componentes.

Estos problemas deben resolverse mediante el uso de circuitos en la PCI que protegen los componentes contra picos anómalos o caídas de tensión o corriente.

Para los cambios dramáticos en los parámetros ambientales, el riesgo está relacionado con la seguridad del usuario. Las altas temperaturas causadas por el uso excesivo de la CPU u otras anomalías pueden causar que se quemen los circuitos, que se derramen productos químicos o incluso incendios.

8.8 Forzar Umbrales del Consumo de Energía

Los Dispositivos Periféricos que proporcionan servicios críticos al usuario deben estar habilitados con un umbral de advertencia que indique eventos relacionados con el consumo de energía. Estos eventos pueden incluir:

- Estado de batería baja
- Batería extremadamente baja
- Eventos de apagado
- Eventos de interrupción momentánea del funcionamiento
- Cambios a la utilización para la alimentación de una batería de respaldo

El usuario debe ser notificado con un cierto tiempo para permitirle reaccionar para compensar la pérdida de energía. Esto podría lograrse con un LED que indica un estado de energía particular según el color, como verde para OK, naranja para Bajo y rojo para Crítico.

Los sistemas que están conectados a la corriente de red alterna deben configurarse para advertir al usuario cuando se producen eventos de interrupción de suministro momentáneo o apagado. Además, el Dispositivo Periférico debe registrar estos eventos en la memoria persistente para garantizar que el usuario y los administradores del equipo puedan recuperar la información en un momento posterior. La información debe tener un “sello temporal”.

El problema en este proceso es identificar a qué velocidad se está agotando la energía de la batería y la energía adicional requerida para notificar al usuario de un cambio en el estado de la alimentación del equipo. Todo esto se puede lograr a través de los conceptos empleados en ingeniería eléctrica, y no debería ser un gran desafío para las empresas de ingeniería experimentadas.

8.8.1 Riesgo

Sin un sistema bien diseñado de advertencia con respecto al consumo de energía, los usuarios no podrán prepararse adecuadamente para eventos potencialmente críticos que afecten a la alimentación de sus equipos. Si bien esto puede ser beneficioso en el caso de dispositivos simples como medidores de frecuencia, temporizadores y otros dispositivos portátiles, los dispositivos más críticos como los localizadores personales, los sistemas telemáticos y los sistemas de seguridad para el hogar pueden verse seriamente afectados por la pérdida de alimentación.

8.9 Entornos sin Conectividad al Back-end

8.9.1 Método

Los Dispositivos Periféricos, especialmente las pasarelas, deben ser capaces de aplicar la seguridad de las comunicaciones incluso en entornos donde la conectividad al back-end no está disponible. Independientemente de si la falta de conectividad es temporal o no, la pasarela o Dispositivo Periférico deben de ser capaces de continuar con las tareas de seguridad como si el sistema de back-end estuviera disponible.

Para lograr esto, la TCB se debe usar para autenticar a todos los pares a los que el Dispositivo Periférico se debe conectar, para manejar datos relativos a la privacidad del

sistema y de configuración o enviar comandos. La TCB se puede usar para garantizar que los mensajes enviados y recibidos con pares determinados provienen de una entidad que ha sido aprovisionada por la misma organización. Esto reduce la probabilidad de que se comunique, por ejemplo, con un dispositivo que haya sido suplantado.

La interoperabilidad aún puede continuar al poder comunicarse con otros dispositivos que no pueden ser autenticados. Sin embargo, el tipo de información que se maneje con estos dispositivos debe restringirse a clases de datos con respecto a la interoperabilidad y no sensibles para el servicio o usuarios.

El reto proviene de decidir qué Dispositivos Periféricos autenticar y con que otros dispositivos comunicarse con texto sin codificar. La empresa debe decidir qué tipos de datos se clasifican y se deben proteger de sus pares no autenticados. Una vez que se logre esta clasificación de datos, la organización podrá determinar qué dispositivos son razonablemente fiables incluso sin la participación en las comunicaciones de los servicios centrales de IoT.

8.9.2 Riesgo

El riesgo de implementar soluciones en entornos sin comunicaciones entre sus elementos es que abre una oportunidad para que la competencia hackée la infraestructura. Los competidores pueden afectar de manera considerable al negocio ofreciendo interoperabilidad y utilizando sitios sin conexión como entornos de prueba.

En cambio, la empresa puede optar por permitir la interoperabilidad, pero hasta cierto punto. Ciertos servicios básicos y la propiedad intelectual fundamental para su negocio pueden protegerse permitiendo las comunicaciones sólo para pares autenticados que se validen mediante el uso de una TCB. Esto ayuda a reducir la exposición del negocio a problemas de propiedad intelectual y a competidores agresivos.

8.10 Desactivación y Retirada del Mercado de Dispositivos

Todos los Dispositivos Periféricos tienen un ciclo de vida, como se discute en otra parte de este documento. Algunos dispositivos deben ser retirados del mercado debido a que un usuario cancela su suscripción, mientras que otros dispositivos deben ser retirados debido a un comportamiento anómalo o sospechoso. Independientemente de la causa, la empresa debe estar preparada para desconectar del servicio IoT al dispositivo de forma segura utilizando su TCB y modelo de comunicaciones.

La retirada gradual, como se analiza en otra parte de este documento, es el proceso de desconexión de un dispositivo de toda una red de equipos y los servicios que los respaldan. Un producto o servicio que se ha vuelto obsoleto para una empresa, o una empresa que decida cerrar, debe de desactivar sus dispositivos y la red que utiliza para disminuir el riesgo de abuso por parte de los atacantes que quisieran hackear estos elementos en proceso de desconexión.

Para lograr esto, deben usarse la TCB y los protocolos que la soportan. En general, el proceso consta de los siguientes pasos:

- Crear un mensaje de desactivación por parte del Ecosistema de Servicios

- Confeccionar un mensaje único dirigido al Dispositivo Periférico que recibe el mensaje
- Firmar el mensaje usando la clave PSK o la clave asimétrica
- Enviar el mensaje hacia el Dispositivo Periférico
- Recibir un mensaje del Dispositivo Periférico que reconoce criptográficamente la desactivación
- Eliminar el Dispositivo Periférico de la lista de dispositivos autenticados
- No permitir más comunicaciones desde este Dispositivo Periférico

En el lado del dispositivo, la aplicación que se ejecuta en el software debería:

- Conectarse a servicios de back-end críticos a través del Ecosistema de Servicios
- Pedir al servicio los mensajes críticos pendientes par él
- Recibir el mensaje de desactivación
- Verificar la firma del mensaje usando la TCB y el ancla de confianza
- Generar el mensaje de acuse de recibo y firmarlo criptográficamente con la clave PSK personalizada o la clave asimétrica
- Ejecutar el comando de desactivación
- Responder al mensaje del servicio crítico

Es importante que el mensaje se firme y prepare para su transmisión antes de la desconexión del dispositivo, ya que el proceso de desactivación incluye la invalidación y eliminación de claves de seguridad del ancla de confianza. Debido a este proceso, las claves utilizadas para firmar el mensaje de desactivación no estarán disponibles. El servicio necesita recibir el mensaje cuya integridad sea verificable para garantizar que el Dispositivo Periférico efectivamente recibió y procesó el mensaje.

La dificultad de este proceso radica principalmente en que la desactivación de un dispositivo potencialmente comprometido presupone que el dispositivo no se ha visto comprometido hasta el punto en que rechazará el comando de desactivación. Si se ha comprometido de manera importante, puede que no ejecute este comando.

Como resultado, es imperativo que el sistema de back-end que se ejecuta en el Ecosistema de servicios no permita que el Dispositivo Periférico se pueda comunicar con servicios críticos. Si el dispositivo intenta interactuar con sus pares en la red o servicios críticos, el sistema de back-end debe generar una alerta y avisar al administrador de sistema que se ha producido el evento anómalo.

8.10.1 Riesgo

Los riesgos de no implementar la desactivación y la retirada del dispositivo del servicio son muchos, desde el compromiso total de una red por parte de atacantes hasta permitir que los dispositivos comprometidos continúen usando servicios conectados a la red. El riesgo más común está asociado con usuarios que han cancelado su suscripción con un proveedor de servicios IoT. Si estos usuarios no se desconectan de la red, pueden continuar comunicándose con otros pares en la red de Dispositivos IoT, o pueden tener acceso a servicios que ya no deberían ser accesibles. Esto genera un costo para el proveedor de

servicios IoT, que debe pagar las comunicaciones, el tiempo de CPU y el almacenamiento en el Ecosistema de Servicios.

8.11 Captura de Metadatos no Autorizada

El IoT en la actualidad está diseñado para unir el mundo físico al mundo digital. En este modelo, los efectos de la tecnología son potencialmente mucho más invasivos que en el pasado. Mediante el uso de metadatos, las empresas o particulares pueden rastrear y vigilar intencionalmente el comportamiento de usuarios de manera aleatoria o intencionada (usuario en concreto).

El análisis de metadatos se utiliza cuando la comunicación entre dos entidades de red está encriptada, pero las estructuras en los protocolos que identifican el tipo de mensaje o la identidad del emisor y/o receptor están expuestas. Estos metadatos se pueden usar para inferir información relevante.

Considere el escenario en el que los automóviles transmiten mensajes que contienen metadatos atribuibles a un usuario específico. Cualquier persona con la capacidad de rastrear (local o remotamente) estos fragmentos de metadatos, puede ser capaz de inferir los movimientos del usuario y por ende el comportamiento o la intención del usuario al moverse. Si existen fallos en la seguridad que puedan explotarse en el sistema telemático del vehículo, es posible rastrear y poner en el ojo de mira al sistema telemático de un usuario específico, poniéndolo en riesgo incluso físicamente.

Las entidades legales y las compañías de seguros están preocupadas acerca de cómo estos riesgos afectarán el futuro de las finanzas en los servicios para los automóviles, y están comenzando a involucrarse en la legislación y los estándares que determinarán cómo los ingenieros deben diseñar equipos telemáticos para este sector. Este cambio eventualmente llegará a las verticales de IoT menos activos, a medida que se desarrolle más este tipo de tecnologías.

Para combatir la captura de metadatos, encripte tantos datos como sea posible y use identificadores binarios únicos para los módulos de comunicación. Haga cumplir una política que impida que los usuarios externos puedan usar la API del sistema de IoT para inferir los números de serie del hardware y otras identidades identificables en los perfiles de los usuarios. Cuando sea posible, no permita que la estructura de un mensaje quede expuesta a terceros. No permita que los comandos, actividades o comportamientos se expongan a terceros. Exigir confidencialidad e integridad en todos los datos relacionados con la privacidad del usuario.

8.11.1 Riesgo

La aplicación de pocas medidas de seguridad en las comunicaciones puede permitir la recolección de datos o metadatos que pone en peligro a un usuario final o expone sus datos privados. A medida que las compañías de seguros logren imponer en el mercado los requisitos de privacidad para el usuario final que use tecnologías determinadas, la empresa puede correr riesgos si no se responsabiliza por los datos que generan sus dispositivos.

9 Recomendaciones de Baja Prioridad

Las recomendaciones de baja prioridad abarcan el conjunto de recomendaciones que se aplican a riesgos que son extremadamente costosos de combatir, o es poco probable que afecten al diseño de Dispositivos Periféricos. Si bien estas recomendaciones son valiosas y la información detallada en las recomendaciones es importante, las estrategias para reducir y reparar las deficiencias de seguridad discutidas pueden estar fuera del alcance de las empresas al considerar la parte económica de rentabilidad del negocio en particular. Evalúe cada recomendación y determine si los riesgos descritos son relevantes o importantes para la empresa y sus clientes. Si los clientes requieren que se aborden estos riesgos, aplique las recomendaciones.

9.1 Denegación de Servicio Intencional e Involuntaria

Para las comunicaciones por radio, existe una amenaza constante de interferencia ("*Jamming*"), o la transmisión intencional de ruido o patrones que pueden utilizarse para codificar señales legítimas. Como las señales de radio se componen simplemente de electrones que se transmiten a través del "aire" con un patrón específico, es bastante fácil inventar una serie de señales que interrumpen o destruyen el patrón que forma los datos para la comunicación entre dispositivos.

Normalmente, el objetivo de dicho ataque es la interrupción simple, para no permitir o denegar el servicio a usuarios legítimos. En otros casos, el abuso puede ser más útil. Por ejemplo, los protocolos de comunicaciones que no tienen mecanismo de autenticación pueden ser falsificados. Para lograr esto, la señal original debe de interferirse de modo que la señal falsificada del adversario tenga más probabilidades de alcanzar el objetivo deseado.

Un ejemplo de esto es la suplantación de los Sistemas de Posicionamiento Global (GPS). Las señales de GPS civiles carecen de encriptación y autenticación, ya que es, en esencia, una señal de transmisión de texto plano que cualquiera puede recibir. También es una señal de radio relativamente débil y se atenúa fácilmente por anomalías ambientales como preamplificadores de frecuencia ultra alta (UHF) para receptores de televisión y microondas.

Para los dispositivos que requieren que la información de ubicación funcione correctamente, una señal de GPS alterada puede dar como resultado un riesgo de fiabilidad que puede provocar una serie de fallos para comprometer la seguridad de la información, especialmente si a posteriori se utiliza el redireccionamiento del dispositivo IoT.

Para combatir las interferencias provocadas y otras formas de ataques intencionales de denegación de servicio (DoS), desarrolle un protocolo de comunicaciones robusto que se centre en métodos para devaluar las interrupciones en el servicio. La red debería detectar si los dispositivos abandonaron repentinamente o de manera anormal la red. Cada Dispositivo Periférico debería "despedirse" con un mensaje cuando tenga la intención de abandonar la red. Si no es así, la anomalía debe tenerse en cuenta para el análisis estadístico.

Además, las claves de seguridad de comunicación deben renegociarse cada vez que un dispositivo se vuelva a unir a la red. La misma clave de seguridad de comunicaciones no debe ser utilizada. Las comunicaciones se deben de arrancar con la misma clave

criptográfica asimétrica, pero cualquier clave simétrica derivada de la negociación de claves debe ser renovada para cada sesión de comunicaciones.

El bloqueo involuntario puede ocurrir en una red radio por muchas razones: condiciones ambientales que no permiten la propagación de la señal, equipos que funcionan mal o incluso equipos adyacentes que operan en la misma frecuencia. Independientemente de la razón subyacente, los ingenieros que confían en las comunicaciones de radio esperan que haya condiciones ambientales que causen la degradación o la pérdida de la señal. Estas pérdidas deben compensarse mediante el diseño de la aplicación y el protocolo de comunicaciones de red.

Se recomienda a los desarrolladores que lean los “Lineamientos de Eficiencia en la Conectividad de la GSMA” [9], que contiene consejos sobre cómo protegerse contra los ataques no intencionales de denegación de servicio y orientan al lector con respecto a los “Informes sobre la Identidad del Host del Dispositivo” (DHIR).

9.1.1 Riesgo

Si no se combate el riesgo de un ataque tipo DoS intencionado, esto tendrá como resultado que el Dispositivo Periférico tenga un comportamiento anormal o inseguro. Si el Dispositivo Periférico siempre usa la misma clave de sesión, esta podría ser una forma en que los atacantes podrían hackear la arquitectura de red para recopilar información sobre la clave simétrica utilizada para proteger las comunicaciones. Crear una sesión segura correctamente después de que cada sesión de comunicaciones termine, es imprescindible para la seguridad de las comunicaciones de un Dispositivo Periférico.

9.2 Análisis sobre la Protección Crítica de un Dispositivo

La mayoría de los productos en IoT incorporarán algún aspecto del mundo físico con tecnología digital. Como resultado, es probable que un ser humano tome una decisión en el mundo físico en función de la información provista por un Dispositivo Periférico IoT. Alternativamente, un dispositivo IoT podría tomar una decisión que afecta al mundo físico con información obtenida a través del mundo digital.

Por lo tanto, es imperativo que los proveedores de servicios de IoT evalúen sus productos desde una perspectiva de protección a los usuarios para determinar si, cómo y cuándo incluso la vida humana puede verse afectada por la tecnología. Si no se ponen en práctica protecciones adecuadas para garantizar que no se abuse de la tecnología para causar daños físicos, sus clientes pueden correr riesgos.

Para ayudar a resolver el problema de la protección física, tenga una conversación con los equipos directivos, legal y de seguros del Proveedor de Servicios de IoT. Asegúrese de que estos equipos entiendan las capacidades y limitaciones de la tecnología utilizada en el producto o servicio. Determine si estas tecnologías pueden satisfacer las necesidades de la empresa y ofrecer a los clientes el nivel de protección necesario para la aplicación prevista.

9.2.1 Riesgo

Claramente, el resultado de no tomarse el tiempo para evaluar el impacto del producto o servicio en la seguridad y protección de los clientes podría resultar en la pérdida de ingresos, accidentes inesperados o incluso la muerte en algunos casos.

9.3 Como Frustrar los Ataques de Imitación de Componentes y Pasarelas no Fiables

Los componentes en el circuito físico (PCI) generalmente no usan algo para proteger la confidencialidad e integridad cuando se comunican entre sí o con la unidad de procesamiento central (CPU) a través de los buses físicos. Como resultado, cualquier adversario puede leer o escribir datos transmitidos en estos buses hardware. El efecto de esta vulnerabilidad de las comunicaciones es la capacidad del adversario para hacerse pasar por dispositivos legítimos en el circuito físico. Si el adversario quisiera, podría suplantar a un componente crítico como una NVRAM, RAM o incluso un ancla de confianza.

El objetivo de este ataque sería pasar por alto la seguridad empleada entre dos componentes en un bus. Un ejemplo típico de este escenario es utilizar esta vulnerabilidad para eludir el proceso de validación de integridad al analizar una imagen de aplicación almacenada en la NVRAM. Cuando la CPU recupera los datos de la memoria almacenados en la NVRAM, el atacante puede usar un sistema de transferencia para robar los contenidos de la memoria real que se transmiten a la CPU. Cuando la aplicación que se ejecuta en la CPU ha verificado la integridad de la imagen de la aplicación, el atacante puede alterar las comunicaciones en el bus físico para intercambiar selectivamente los contenidos de la NVRAM en su beneficio. En otras palabras, la CPU verifica una imagen de la aplicación (la imagen original) pero luego carga la imagen del atacante en la RAM y la ejecuta.

Una forma de protegerse contra este ataque es:

- Cargando los contenidos de la NVRAM en la RAM
- Validando la imagen de la aplicación cargada en la RAM
- Ejecutando el código directamente en la memoria RAM o guardando los contenidos de la RAM a modo de una caché

También se debería considerar en este punto que un atacante podría hackear la RAM, haciendo inútil este proceso. Sin embargo, realizar un ataque tipo “man-in-the-middle” contra la RAM es mucho más complejo y costoso que un ataque contra la NVRAM porque la velocidad del bus y los patrones de acceso son mucho más rápidos y difíciles de predecir que con una NVRAM, que se accede principalmente en bloques.

Alternativamente, el atacante puede generar sumas de comprobación para regiones más pequeñas de contenido NVRAM validado y comprobar periódicamente las firmas en la NVRAM. Si las sumas de comprobación son diferentes, entonces el contenido está siendo manipulado. Esto puede tener éxito, pero tiene un potencial de éxito menor porque el adversario solo puede manipular una pequeña cantidad de datos que la aplicación al azar, durante la ejecución, no verifica.

Cabe señalar que, si bien la mejor manera de protegerse contra este ataque es validar el contenido de la NVRAM y luego cargarlo en la RAM ejecutable, no hay una solución tipo para este problema. El costo de asegurar componentes físicos es tan alto que no es práctico resolver este ataque de una manera más elaborada, a menos que el cliente requiera tales garantías de seguridad.

Este ataque es aún más sencillo cuando se usa un protocolo de comunicaciones físicas más básico, como I2C. Los buses I2C son esencialmente sistemas de transmisión de bits. Por lo tanto, cualquier componente que se encuentre en el bus I2C puede pretender ser cualquier otro componente. Esto permitirá que un adversario se haga pasar por otros dispositivos en el bus que no comprueban la confidencialidad e integridad en el canal de comunicaciones. Cuando esto sea una preocupación, imponga confidencialidad e integridad en el protocolo de seguridad de la aplicación utilizado por encima del protocolo de bus físico.

9.3.1 Riesgo

El riesgo de no implementar ninguna solución al respecto dará como resultado la capacidad de un atacante para eludir las verificaciones de integridad en la aplicación. Esto permitirá que el atacante comprometa la aplicación que está ejecutando un código con más privilegios, como por ejemplo el de los gestores de arranque o las TCBs.

Cabe señalar, sin embargo, que este ataque es mucho menos probable que los ataques más sencillos contra el gestor de arranque. Realizar un ataque de hardware tipo “man-in-the-middle” contra componentes como una NVRAM, o componentes de alta velocidad como una RAM, es muy difícil, complejo y actualmente costoso. Si bien siempre será posible para un atacante hackear un sistema embebido de esta manera, puede ser demasiado costoso y por lo tanto prohibitivo.

Por lo cual, cargar el código en la memoria RAM y verificar su integridad puede ser una solución razonable que evitará la mayoría de los ataques, si los hubiera.

Además, por las razones descritas anteriormente y más, las claves criptográficas no deben gestionarse con privilegios inseguros. Deben almacenarse en un ancla de confianza y ser manipulados por la TCB, y nunca almacenarse en medios como la NVRAM que puedan ser suplantados o comprometidos.

9.4 Frustrar un Ataque de Arranque en Frío

Un ataque de arranque en frío es una estrategia de ataque físico contra los sistemas informáticos que extrae los secretos de una computadora en funcionamiento al eliminar la memoria física de la computadora y colocar la memoria en un sistema secundario controlado por el atacante. La ventaja de este ataque es que el hacker puede ejecutar un sistema operativo personalizado que vuelca los contenidos de la RAM a un almacenamiento permanente. Esto permitirá que el atacante revise los datos recuperados y determine si hay tokens relacionados con la seguridad que puedan usarse. Esto puede incluir:

- Secretos criptográficos o claves privadas
- Credenciales de inicio de sesión (nombres de usuario y contraseñas)
- Información de identificación personal (PII)
- Tokens de acceso para servicios web

El objetivo del ataque es poner en peligro los secretos que permiten obtener acceso por largo tiempo a un recurso que de otro modo estaría fuera de su alcance. Por ejemplo, romper los algoritmos criptográficos utilizados en el estándar más reciente de TLS sería imposible para un atacante promedio. Sin embargo, comprometer el certificado de cliente

privado utilizado en un servicio de autenticación mutua TLS permitiría al atacante simular al cliente desde un sistema controlado por él.

Para tener éxito en este ataque, el atacante debe ser capaz de eliminar la RAM del sistema informático de destino sin que cambien los bits almacenados en el chip. Como se detalla en el paper referenciado [22], esto se puede lograr enfriando los chips de memoria. Sin embargo, la memoria RAM debe ser fácilmente extraíble. Si la RAM está soldada a la placa de circuito, esto complicaría enormemente el ataque y requeriría que el atacante use una pistola de soldadura para extraer la memoria, lo que podría dañar su contenido.

Es importante tener en cuenta que el borrado o depuración (limpieza) del contenido de la memoria en el momento del apagado del dispositivo siempre es útil, y se recomienda, para mejorar la privacidad de un Dispositivo Periférico. Sin embargo, un ataque de arranque en frío puede ocurrir en cualquier momento, incluso mientras el sistema está funcionando. Por lo tanto, limpiar el contenido la memoria puede ser útil, pero puede no ser suficiente para evitar los ataques en el mundo real.

Un método más efectivo para anular este tipo de ataques, es procesar los comandos relativos a la seguridad utilizando sólo la RAM interna de la CPU. Muchas CPUs, MCUs y MPUs tienen una pequeña cantidad de SRAM interna que puede ser utilizada por una aplicación durante la ejecución de un programa. Si la aplicación limita el procesamiento de tokens de seguridad críticos (como las claves privadas) a esta RAM interna, los contenidos de la RAM extraíble (o externa) tendrán menos valor para un atacante.

9.4.1 Riesgo

No considerar la posibilidad de un ataque de arranque en frío y sus riesgos, puede hacer que se extraigan claves de seguridad críticas usando un modelo de ataque simple. Si las claves de seguridad son universales para todos los Dispositivos Periféricos en el ecosistema del proveedor de servicios de IoT, es posible que se produzca un compromiso muy importante para todo el ecosistema.

Para obtener más información, consulte: https://citp.princeton.edu/research/memory/

9.5 Riesgos de seguridad no Obvios (“Ver a Través de las Paredes”)

A pesar de habilitar y hacer cumplir la autenticación mutua, la confidencialidad y la integridad en la red de comunicaciones, los patrones de tráfico pueden correlacionarse directamente con eventos. Cuando cierto tipo de datos se transmiten en respuesta a ciertos eventos físicos, finalmente se puede establecer una correlación entre los eventos físicos y los datos. Esto puede permitir que un atacante monitoree los patrones en las señales, luego podría descifrar el significado de los patrones, ya sea que el adversario tenga o no acceso directo a los datos de texto sin codificar.

Un ejemplo de esto es la tecnología domótica que reacciona en función de la presencia física de un usuario en una habitación en particular. Un atacante capaz de monitorear remotamente el sistema de comunicaciones puede ser capaz de observar cuántos usuarios se encuentran en una casa particular, dónde se encuentran los usuarios en el hogar y quienes son los usuarios en particular, únicamente observando los patrones de

comunicación entre los Dispositivos Periféricos de IoT, pasarelas y los sistemas de back-end.

El atacante puede diferenciar fácilmente entre un hogar “muy poblado” y un hogar donde un solo individuo se encuentra en cualquier momento y el lugar preciso donde está. Las compañías de seguros y las entidades jurídicas deberán comprender cómo esto aumenta potencialmente el riesgo para los propietarios y otros inquilinos en el hogar.

Combatir este riesgo puede ser difícil. El modelo más común y simple para hacerlo es enviar muestras a una velocidad predefinida, independientemente de si hay un usuario presente para tomar muestras. Si se aplica la confidencialidad y la integridad, impidiendo que los atacantes remotos evalúen el texto plano en los datos, un observador no podrá diferenciar entre una muestra que contenga la actividad del usuario y una muestra vacía.

Sin embargo, hay consideraciones para tener en cuenta con este modelo, como el aumento de la saturación del espectro, el aumento del consumo de energía para dispositivos con tecnología de baja potencia o alimentados con batería, y el mayor nivel de procesamiento requerido para des-encriptar, verificar e interpretar los paquetes de muestra vacíos.

Una alternativa es enviar muestras a intervalos aleatorios, con ráfagas variables. Este tipo de patrón es menos costoso, consume menos energía y requiere menos potencia de procesamiento. Sin embargo, aún es posible observar cambios sutiles que indican la presencia del usuario. Por ejemplo, cualquier sistema verdaderamente entrópico es completamente aleatorio e impredecible. El comportamiento del usuario, sin embargo, es completamente predecible. Si un usuario entra a una habitación y los sensores en esa habitación se activan y comienzan a enviar datos a Dispositivos Periféricos de IoT en la red, la introducción de un comportamiento consistente en las comunicaciones puede indicar la presencia de un usuario.

Cualquier equipo que desarrolle tecnología sujeta a este tipo de riesgo debe investigar los efectos potenciales de la exposición de la privacidad y consultar con su equipo legal para determinar si la tecnología tendría un efecto sobre la postura legal o modelo del seguro a considerar por parte de la empresa.

9.5.1 Riesgo

Si el proveedor de servicios de IoT no evalúa su tecnología desde la perspectiva de posibles vulnerabilidades de la privacidad de los datos y riesgos de seguridad, la arquitectura puede necesitar una revisión sustancial para compensar los riesgos que deben abordarse. En lugar de intentar realizar ajustes costosos en la arquitectura más adelante, diseñe estas soluciones en el producto al inicio de la fase de desarrollo, o tan pronto como sea posible.

9.6 Lucha contra Haces de Iones Focalizados (FIB) y Rayos X

Un Haz de Iones Focalizado (FIB) es un instrumento de fabricación comúnmente utilizado en la evaluación y test de semiconductores (Chips). La tecnología es capaz de inspeccionar y alterar circuitos a nivel de nanómetros, lo que permite a los analistas identificar fallos en la fabricación y probar parches del circuito antes de alterar el proceso de fabricación.

En el entorno de la seguridad de la información, un FIB se puede utilizar para acceder a buses internos con el fin de interceptar datos transmitidos a componentes internos. Además, un FIB se puede utilizar para alterar los circuitos internos, lo que cambia la forma en que funcionará el componente en cuestión, lo que permite al adversario evitar una restricción de seguridad.

Casi todos los dispositivos están sujetos a un ataque por FIB. Sin embargo, solo ciertos dispositivos serán sujetos al uso de un FIB. Esto se debe a que un FIB en sí mismo es una tecnología extremadamente cara, de aproximadamente 1,000,000 USD por unidad. Debido al alto costo de la tecnología, pocas empresas tienen dicho dispositivo en su poder como herramienta de test. Además, el dispositivo no está automatizado. Se requiere un alto grado de conocimientos para manejarlo, así como un alto grado de experiencia en el análisis de semiconductores, para que se pueda sacar provecho de su utilización. Por lo tanto, el costo real de un FIB es mucho mayor a lo mencionado mas arriba ya que abarca cuestiones como el entrenamiento, salario y la experiencia del usuario requerida.

Hay empresas disponibles para su subcontratación expertas en FIB, sin embargo, como la ingeniería inversa es hasta cierto punto legal, las empresas en cuestión pueden proporcionar servicios de ataque a semiconductores para los clientes que estén interesados en la ingeniería inversa de un dispositivo. Estos contratos pueden costar entre 10,000 USD y 1,000,000 USD dependiendo del nivel de personalización y experiencia requerida para atacar un componente en particular. Por ejemplo, una compañía de este tipo tendría un catálogo de técnicas para eludir protecciones en un chip común. Pero una solución de FPGA personalizada con una nueva tecnología de bloqueo de seguridad costaría mucho más, ya que no se pueden aplicar las técnicas al uso. Se requeriría un nuevo proceso para usar el FIB con éxito, lo que implica un costo considerable de tiempo y dinero.

Algunas tecnologías nuevas, como las variantes modernas de anclas de confianza, afirman que son resistentes a las sondas FIB. Si bien existe cierta validez para estas afirmaciones, cualquier protección de hardware que no sea dinámica (y la mayoría no lo son) dará como resultado un catálogo de técnicas después de que se haya dedicado suficiente tiempo al análisis de las técnicas de derivación. Por lo tanto, las afirmaciones pueden ser válidas, pero solo por un período de tiempo.

Por lo tanto, para compensar las técnicas invasivas de ataque como estas que casi siempre tienen éxito, es imperativo que la empresa de ingeniería diseñe una estrategia de seguridad que no delimite su éxito únicamente alrededor del ancla de confianza. En su lugar, se debe diseñar un protocolo adecuado que utilice esta tecnología como un ancla de confianza base, pero personalizando las claves criptográficas de cada Dispositivo Periférico de tal forma que ningún compromiso de un solo dispositivo resulte en un compromiso de toda la red de Dispositivos Periféricos.

Considere el escenario en el que un adversario debe usar un FIB para extraer información criptográfica de cada Dispositivo Periférico que desea atacar. Esto se convertiría rápidamente en una propuesta extremadamente costosa, y estaría fuera del alcance con respecto al presupuesto de casi cualquier organización. Dado que estas metodologías de ataque no se pueden evitar por competo, se deben “devaluar”, para disminuir el riesgo a

través de una arquitectura robusta de seguridad, no a través de la ocultación de información.

9.6.1 Riesgo

El riesgo de una FIB es que los secretos criptográficos y otros datos relacionados con la propiedad intelectual pueden extraerse de un componente, incluso de uno que esté reforzado con técnicas avanzadas de seguridad. Debido a que no es práctico derrotar a un FIB de manera rentable para un sistema de IoT de consumo, la empresa debe modificar su estrategia para proteger los Sistemas de Dispositivos Periféricos o arriesgarse a un compromiso global de todo el sistema.

9.7 Asegurar la Cadena de Suministro

La seguridad de cualquier sistema informático comienza con los componentes en bruto que componen la PCI. El silicio, los tokens criptográficos, la memoria de solo lectura (ROM), el firmware y otras características generales de un sistema embebido contribuyen a la seguridad de dicho sistema. Si alguno de estos componentes es manipulado, todo el sistema podría estar sujeto a un compromiso de seguridad.

Como resultado, los proveedores de servicios de IoT que son conscientes de la seguridad deben tener en cuenta la fuente de sus componentes, su ensamblaje y el proceso de testeo y calidad utilizado para distribuir los equipos ensamblados. Si el proceso utilizado para implementar la tecnología no se planifica cuidadosamente, un único punto de fallo en el proceso podría ocasionar un fallo de seguridad crítico.

Considere las siguientes cuestiones clave:

- ¿Dónde y quién fabrica el silicio?
- ¿El diseño de silicio ha sido analizado por un equipo humano externo fiable y experto en la seguridad de la información?
- ¿Se fabricará el silicio en una instalación segura?
- ¿Cómo se grabará la EEPROM o la NVRAM con una imagen ejecutable, como por ejemplo la de un gestor de arranque?
- ¿Es seguro el proceso para actualizar la imagen ejecutable?
- ¿Cómo se entregará la imagen ejecutable al fabricante?
- ¿Se verifica la imagen ejecutable una vez que se ha grabado en la EEPROM o la NVRAM?
- ¿Cómo se aprovisionan los secretos criptográficos en el(los) chip(s)?
- ¿Si el fabricante genera los secretos, están utilizando un generador de números aleatorios (RNG) para generar las claves?
- ¿Son todas las claves de seguridad únicas según las recomendaciones de la TCB?
- ¿Cómo se comparten los secretos criptográficos con el Proveedor de servicios IoT? ¿De forma segura?
- ¿Cómo se correlacionan los identificadores únicos de los chips (número de serie, etc.) con los secretos criptográficos, y cómo se comunican al proveedor de servicios IoT?

Si bien la elección de una instalación más segura para construir y ensamblar un producto puede incurrir en un mayor costo, puede ser un paso imprescindible para la empresa. Esto depende del caso de uso del producto, el entorno de despliegue previsto, el cliente previsto y otros factores como la seguridad humana, las aplicaciones militares y las implementaciones de infraestructuras críticas. Donde la vida humana puede verse afectada por la tecnología resultante, la cadena de suministro debe evaluarse para detectar lagunas en la seguridad.

9.7.1 Riesgo

Sin la seguridad en la cadena de suministro, la organización está sujeta a muchos riesgos, algunos de los cuales pueden ser completamente inesperados y, sin embargo, críticos para el negocio:

- Clonación de un Dispositivo Periférico (fabricación ilegal)
- Robo de tecnología (competidores que roban y se saltan al proveedor de servicios)
- Robo de credenciales (interceptación de datos o ataques de suplantación)
- Inyección de implantes ("puertas traseras" maliciosas que pueden activarse más adelante)

9.8 Interceptación Legal

La interceptación legal es el acto de interceptar o manipular legalmente las comunicaciones entre un cliente y un proveedor de servicios. Esto puede funcionar en una de dos formas. En primer lugar, el escenario más común es que un agente responsable de la aplicación de la ley presente una solicitud legal a un operador y solicite acceso a los metadatos o datos reales de las comunicaciones realizadas por un suscriptor específico. En segundo lugar, este agente solicitará al proveedor de servicios de IoT acceso a los datos y/o metadatos de un suscriptor específico. En el escenario donde el agente solicita acceso a través del operador, el proveedor de servicios de IoT podría no ser notificado de que existe un problema, dependiendo del alcance de la solicitud legal. Por lo tanto, el proveedor del servicio debe estar listo para implementar o cumplir con una solicitud legal de este tipo.

Por lo tanto, el proveedor debe identificar qué problemas de privacidad pueden derivarse de una solicitud de este tipo, y debe estar preparado para proporcionar la información relevante siguiendo el modelo legal y la política de privacidad propios de la organización, dentro de sus capacidades legales respectivas.

Recientemente, empresas como Google, Apple y otras grandes entidades han adoptado “Canarios de Seguridad” (“Canary warrants”) para informar legalmente a los usuarios cuando se ha realizado una solicitud secreta a la empresa en nombre de un agente vinculado a una autoridad jurídica. La empresa puede eliminar una frase, imagen u otro objeto que sea la consecuencia de haber estado en contacto con agentes relacionados con la interceptación legal. La eliminación de este objeto es indicativa, por supuesto, de que una solicitud ha sido interpuesta.

9.8.1 Riesgo

No estar preparado para una solicitud de interceptación legal puede poner a una empresa en desventaja si dicho requisito se aplica al negocio al que se dedica. Es posible que la

empresa tenga que cumplir con la solicitud, pero puede que no tenga la infraestructura legal o las políticas de privacidad bien definidas, lo que podría impactar claramente al negocio.

Si no se configuran en el Dispositivo Periférico los protocolos y la plataforma IoT para adoptar políticas de confidencialidad e integridad adecuadas, las comunicaciones se interceptarán en el lado de la red sin que la empresa lo sepa. Esto puede poner a la empresa en riesgo de que los datos del usuario se filtren o se asocien con un evento como las filtraciones en el caso “Snowden NSA”, disminuyendo sustancialmente la confianza del público en la capacidad de la organización para proteger los datos del usuario.

10 Resumen

En resumen, casi todos los riesgos de seguridad en un producto o servicio de IoT pueden ser combatidos por una arquitectura bien definida, introduciendo en los productos inteligencia para identificar los riesgos antes y durante los eventos relacionados con la seguridad, y las políticas y procedimientos para gestionar dichos eventos. Al analizar qué conceptos de seguridad de alto nivel son importantes para el proveedor de servicios de IoT, se pueden revisar las cuestiones y vulnerabilidades de seguridad más frecuentes. Esto debería guiar al equipo de ingeniería hacia las recomendaciones más relevantes para resolver las vulnerabilidades en su arquitectura.

A medida que el equipo progresa en la definición de la arquitectura, puede comenzar por revisar recomendaciones genéricas y a medida que surjan dudas y preocupaciones de seguridad más exclusivas de su propia implementación, resolverlas aplicando recomendaciones mas específicas.

En general, cada equipo de ingeniería enfrentará riesgos de seguridad muy similares. Es imperativo que la empresa comparta sus inquietudes internamente con todos los actores involucrados para construir una base de conocimiento común tanto para los riesgos como para las estrategias de remediación. Juntas, las organizaciones de un empresa o externas que colaboren de alguna manera pueden construir tecnología y conocimiento para ayudarse unos a otros en la implementación de entornos seguros en el futuro de IoT.

Anexo A Ejemplo de Uso de una Arquitectura de Arranque Genérica (“Bootstrap”)

El nivel de seguridad global de una red con múltiples conexiones está definido por el eslabón más débil de la cadena. Por lo tanto, un enlace local entre un Dispositivo Periférico IoT y una pasarela debe asegurarse con un nivel comparable de seguridad como el que se utilizaría en una la red más extendida (WAN) para mantener el mismo nivel de seguridad en todo el ecosistema.

Como un buen candidato, se podría utilizar “Generic Bootstrap Architecture” (GBA) [17] que sirve para la autenticación como para el mantenimiento de la integridad de los datos. Esta tecnología se basa en claves pre-compartidas que a posteriori se usan para generar claves efímeras (tokens) como base de la autenticación y del cifrado en las comunicaciones.

Autenticación es el proceso de determinar si alguien o algo es, de hecho, lo que “dice” que es. En el entorno de IoT, donde miles de Dispositivos Periféricos estarán activos, determinar qué comportamiento de comunicación es genuino y confiable es primordial. El mecanismo establecido para crear esta relación de confianza debe satisfacer el requisito de ser escalable y mantenible en el tiempo. Además, la variedad de servicios de IoT impone el requisito de que el mecanismo de autenticación se pueda adaptar para coexistir con estos servicios y así mantener una infraestructura común. Un mecanismo que ha demostrado a través del tiempo de ser óptimo es la autenticación de red basada en una tarjeta SIM. Esta infraestructura de autenticación tiene la virtud de proporcionar no solo autenticación, sino también capacidades de cifrado basadas en secretos pre-compartidos. La explosión en el número de Dispositivos Periféricos y el alcance global de IoT ha hecho que el uso de la SIM sea limitado debido a los problemas introducidos por el Roaming y la vulnerabilidad intrínseca de seguridad al poder extraer físicamente una tarjeta SIM de un Dispositivo Periférico no atendido. La llegada de tecnologías como el Embedded SIM (pre-soldada) proporciona una infraestructura muy práctica para la autenticación basada en secretos pre-compartidos, expandiendo y mejorando las capacidades de autenticación de red basada en una SIM común. Además, es más probable que el crecimiento de IoT ocurra en forma de redes capilares (como en redes PAN como se muestra en las configuraciones de los ejemplos 2, 3 y 4 en el capítulo 3 de este documento). Estas redes capilares son “enjambres” de Dispositivos Periféricos conectados a una pasarela. La mayoría de estos dispositivos son Dispositivos Periféricos ligeros (es decir, que no tienen SIM ni conectividad celular). Sin embargo, estos dispositivos requieren también de autenticación y capacidades de cifrado. En las redes capilares, la principal responsabilidad de la autenticación recae en la pasarela, reduciendo el número de Dispositivos Periféricos basados en SIM complejos en la red global de IoT. Esta autenticación y tecnologías de seguridad deben extenderse desde la pasarela al Dispositivo Periférico, creando así un canal seguro desde el Dispositivo Periférico a la plataforma de servicios IoT.

La autenticación basada en SIM está destinada a servir una única aplicación, es decir, la autenticación de un único Dispositivo Periférico para la conexión a la red. Los Dispositivos Periféricos tendrán una multitud de servicios, cada uno con una necesidad diferente y exclusiva de autenticación. Se requiere un marco que amplíe la autenticación de red a múltiples servicios. Un marco que fue diseñado para este propósito es GBA. GBA

aprovecha la infraestructura basada en SIM para generar claves compartidas en el tiempo entre dispositivos y funciones de aplicación de red (NAF: “Network Application Functions”). GBA es un método de autenticación estandarizado por 3GPP en la especificación 3GPP TS 33.220 [17]. El método permite la autenticación de un dispositivo con una suscripción 3GPP a un servicio. Las credenciales de la suscripción se encuentran en el dispositivo, generalmente almacenadas en una SIM, en forma de un circuito integrado (UICC), o como credenciales administradas de forma remota, almacenadas y administradas en una SIM embebida, por ejemplo, como la especificada por la GSMA (eUICC) [5].

Las ventajas de este esquema de seguridad son:

- Autenticación mutua basada en PSK únicamente entre un dispositivo y una función de aplicación de red o autenticación de UE basada en clave compartida con autenticación NAF basada en certificado (TS 33.222) [18].
- Las credenciales se pueden asegurar en un entorno de confianza
- Si se usa una eUICC, las credenciales se pueden cambiar con comandos OTA.
- Escalabilidad. La complejidad y el costo del mantenimiento aumenta acorde a la cantidad de dispositivos de manera lineal, ya que la autenticación está integrada en el esquema de funcionamiento seguro empleado.
- Integridad de los datos. Las claves efímeras generadas durante la autenticación se pueden usar para establecer túneles tipo TLS-PSK, por lo que estas conexiones proporcionarán integridad y confidencialidad de los datos apropiadas.

Anexo B Tutorial sobre el Uso de Tarjetas UICC en un Servicio de IoT

La UICC estandarizado en ETSI TS 102 221 es una plataforma de tarjeta inteligente (un elemento seguro inviolable a prueba de fallos y programable) que proporciona una interfaz a un sistema de archivos seguro e interoperable y un marco de aplicaciones seguro para los dispositivos que integren una UICC. ETSI TS 102 221 proporciona un marco para que un dispositivo de este tipo descubra aplicaciones relevantes en la UICC, y cada aplicación en el UICC corresponde a un conjunto conocido de datos relativa a la configuración y aprovisionamiento, así como procedimientos operativos (como autenticación o derivación de claves) que pueden ser compatibles con el dispositivo anfitrión de la UICC según sus necesidades.

En el contexto de IoT, una UICC puede estar disponible en múltiples encapsulados con rangos operativos distintos como se especifica en ETSI TS 102 671. En su realización más simple, la UICC normalmente es propiedad de un operador de red y solo incluye una aplicación de acceso a la red (aplicación SIM según 3GPP TS 51.011, USIM según 3GPP TS 31.102, CDMA CSIM según lo especificado por 3GPP2, WiMAX SIM, etc.). En este caso, la UICC proporciona un soporte estandarizado para albergar la información de configuración y suministro de seguridad, así como procedimientos criptográficos en un dispositivo móvil para habilitar el acceso a la red, con mecanismos adicionales para administrar remotamente el contenido de la UICC, utilizando ETSI TS 102 225 / TS 102 226. El ecosistema de la red móvil cuenta con procedimientos para garantizar la personalización segura y el despliegue de una UICC bajo el control del operador de red, lo que da como resultado el establecimiento de claves simétricas compartidas individuales entre los dispositivos que incluyen una UICC y la infraestructura de red.

Una característica importante de la plataforma UICC es el soporte de Dominios de Seguridad aislados que permiten a múltiples partes interesadas en un ecosistema complejo asignar cada uno su propia área en una UICC y gestionar su contenido con confidencialidad independientemente de los otras partes interesadas. Esta funcionalidad se hereda de la norma ETSI TS 102 226 perteneciente a la especificación de la tarjeta GlobalPlatform [15] enmienda A. Por lo tanto, en un contexto de IoT, una única UICC permite que múltiples interesados almacenen y administren sus propias credenciales de forma independiente.

En general, una UICC puede contener varias aplicaciones de acceso a la red (con una sola activa a la vez) y potencialmente otras aplicaciones que garanticen el acceso a servicios más elaborados, como aplicaciones ISIM para poder acceder a IMS (como se especifica en 3GPP TS 31.103) o, en el caso de servicios IoT, aplicaciones 1M2M SM especificadas en el anexo D de oneM2M TS-0003. Una aplicación 1M2MSM puede admitir el aprovisionamiento directo de credenciales dedicadas exclusivamente a los servicios o aplicaciones de IoT, así como la derivación de las credenciales de acceso a la red preexistentes en la UICC utilizando el mecanismo GBA especificado por 3GPP. Además, permite a un proveedor de servicios IoT personalizar los procedimientos criptográficos de acuerdo con sus necesidades específicas, p.ej. para admitir mecanismos de autenticación de servicios específicos.

Una única UICC también puede contener múltiples aplicaciones 1M2MSM, lo que permite la utilización confidencial de claves simétricas dedicadas a cada proveedor de servicios IoT. El propietario de la UICC (normalmente un operador de red o fabricante OEM en el contexto de IoT) puede compartir espacio en su UICC con los proveedores de servicios de IoT que lo soliciten, de modo que estos últimos pueden utilizar la cadena de personalización UICC acreditada y la infraestructura para la implementación segura de credenciales de acceso a la red, para implementar sus propias credenciales.

Cuando la seguridad de una aplicación IoT depende del uso de criptografía asimétrica, las aplicaciones UICC personalizadas pueden aplicar esta tecnología de manera similar para facilitar el despliegue de pares de claves públicas o privadas, según sea necesario para un servicio IoT específico. Dichas aplicaciones UICC deben especificarse y admitirse en los dispositivos que alojen una UICC en base a una aplicación específica de IoT.

Anexo C Gestión del Documento

C.1 Historia del Documento

Versión	Fecha	Breve Descripción de Cambios	Aprobación Autoridad	Editor / Compañía
1.0	08-Feb-2016	New PRD CLP.13	PSMC	Ian Smith GSMA & Don A. Bailey Lab Mouse Security
1.1	07-Nov-2016	Se agregaron referencias al esquema de evaluación de seguridad de IoT de GSMA. Correcciones editoriales menores.	PSMC	Ian Smith GSMA
2.0	29-Sep-2017	Se agregaron referencias a los recursos disponibles en la GSMA sobre redes LPWA mas pequeñas actualizaciones.	IoT Security Group	Rob Childs GSMA

C.2 Otra Información

Tipo	Descripción
Dueño del Documento	GSMA IoT Programme
Contacto	Rob Childs – GSMA

Es nuestra intención proporcionar un producto de calidad (documento) para su uso. Si encuentra algún error u omisión, contáctenos con sus comentarios. Puede notificarnos a esta dirección: prd@gsma.com.

Sus comentarios o sugerencias y preguntas son siempre bienvenidos.