Diretrizes de Segurança em IoT para o Ecossistema de Endpoints

Diretrizes de Segurança em IoT para o Ecossistema de Endpoints

Versão 2.0

31 de outubro de 2017

Este é um documento de referência permanente e não vinculante da GSMA

Classificação de Segurança: Não confidencial

O acesso e a distribuição deste documento são restritos às pessoas permitidas pela classificação de segurança. Este documento é confidencial para a Associação e está sujeito a proteção de direitos autorais. Este documento deve ser utilizado apenas para os fins para os quais foi fornecido e as informações contidas nele não devem ser divulgadas ou de qualquer outra forma disponibilizadas, no todo ou em parte, a pessoas que não as permitidas no âmbito da classificação de segurança sem a aprovação prévia por escrito da associação.

Aviso de Direitos Autorais

Aviso Legal

A GSM Association ("Association") não oferece garantia (expressa ou implícita) derivada da precisão ou totalidade das informações contidas neste documento. As informações contidas neste documento estão sujeitas a alterações sem aviso prévio.

Aviso Antitruste

As informações contidas neste documento estão em total conformidade com a política antitruste da GSM Association.

Conteúdo

1 Introdução

1.1 Introdução à Série de documentos “Diretrizes de segurança para IoT” da GSMA

Este documento é uma parte de um conjunto de documentos da GSMA que contém diretrizes de segurança destinadas a ajudar a indústria emergente de Internet das Coisas a estabelecer uma compreensão comum dos problemas de segurança a ela relacionados. Esse conjunto de documentos propõe uma metodologia para o desenvolvimento de serviços seguros de IoT para garantir que as melhores práticas na área sejam implementadas ao longo do ciclo de vida do serviço. Os documentos fornecem recomendações sobre como mitigar ameaças e falhas de segurança comuns dentro dos serviços de IoT.

A estrutura do conjunto de documentos de diretrizes de segurança da GSMA é mostrada abaixo. Recomenda-se que este documento de visão geral 'CLP.11 Panorama das Diretrizes de Segurança para IoT' [1] seja lido como referência antes da leitura dos documentos de suporte CLP.12 [2] e CLP.13 [3] (este documento).

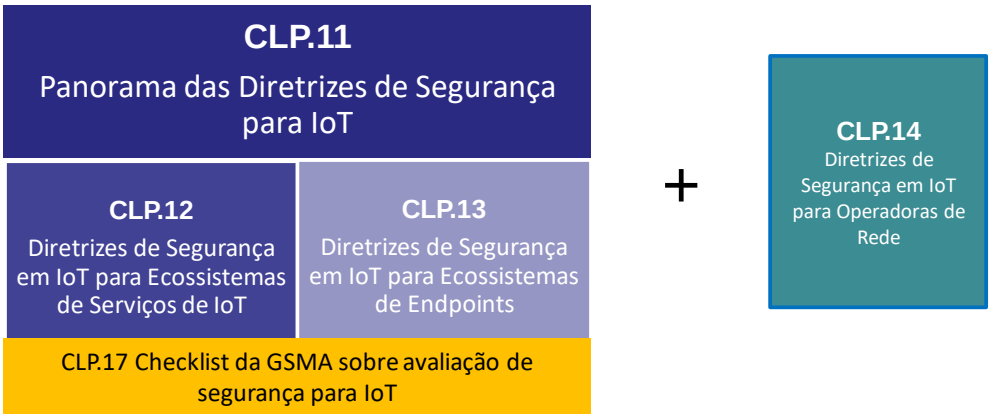

Figura 1 - Estrutura do documento “Diretrizes de segurança em IoT” da GSMA

Operadoras de rede, provedores de serviços de IoT e outros parceiros do ecossistema IoT são aconselhados a ler o documento CLP.14 da GSMA "Diretrizes de segurança em IoT para operadoras de rede" [4], que fornece diretrizes de segurança de alto nível para operadoras de rede que pretendem prestar serviços a provedores de serviços de IoT para garantir a segurança do sistema e a privacidade dos dados.

1.1.1 Checklist da GSMA sobre avaliação de segurança para IoT

Um checklist para avaliação é fornecido no documento CLP.17 [19]. Este documento permite aos fornecedores de produtos, serviços e componentes da IoT autoavaliar a conformidade de seus produtos, serviços e componentes com as Diretrizes da GSMA de Segurança da IoT.

Completar um checklist da GSMA para avaliar a segurança em IoT [19] permitirá que uma entidade demonstre as medidas de segurança que tomaram para proteger seus produtos, serviços e componentes de possíveis riscos relacionados à segurança cibernética.

Avaliações podem ser emitidas por meio do envio de uma declaração completa à GSMA. Consulte o processo no site da GSMA:

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

1.2 Objetivo do documento

Este documento deve ser usado para avaliar os componentes de um serviço de IoT, da perspectiva de um dispositivo endpoint. Um endpoint, a partir de uma perspectiva de IoT, é um dispositivo de computação física que executa uma função ou tarefa como parte de um produto ou um serviço conectado à internet. Um endpoint pode ser, por exemplo, um dispositivo vestível, um sistema de controle industrial, uma unidade embarcada em veículos ou mesmo um drone de uso pessoal. Todas as tecnologias usadas para executar o dispositivo físico devem ser avaliadas em relação aos riscos de segurança. O resultado é um conjunto prático de diretrizes de design que permitem ao leitor identificar e remediar quase todos os riscos potenciais para o serviço de IoT.

O escopo deste documento está limitado às recomendações relativas ao desenvolvimento e implementação de dispositivos endpoint IoT.

Este documento não tem como objetivo estimular a criação de novas especificações ou padrões de IoT, mas fará referência às soluções, padrões e práticas recomendadas atualmente disponíveis.

Este documento não se destina a acelerar a obsolescência dos serviços de IoT existentes. A compatibilidade com versões anteriores dos serviços de IoT existentes da operadora de rede deve ser mantida quando eles são considerados adequadamente protegidos.

Nota-se que a adesão às leis e regulamentos nacionais para um determinado território pode, quando necessário, se sobrepor às diretrizes estabelecidas neste documento.

1.3 Público-alvo

O público principal deste documento é composto por:

- Provedores de Serviços de IoT - empresas ou organizações que procuram desenvolver produtos e serviços conectados inovadores. Exemplos de setores em que os Provedores de Serviços de IoT operam incluem casas inteligentes, cidades inteligentes, automotivo, transporte, saúde, utilities e eletrônicos de consumo, entre outros.
- Fabricantes de dispositivos IoT - provedores de dispositivos IoT para provedores de serviços de IoT com objetivo de permitir serviços de IoT.
- Desenvolvedores de IoT que criam serviços de IoT em nome dos provedores de serviços de IoT.
- Operadoras de rede que são prestadores de serviços de IoT.

1.4 Definições

Termo	Descrição
Nome do ponto de acesso	Identificador de um ponto de conexão de rede ao qual um dispositivo endpoint se conecta. Está associado a diferentes tipos de serviços e, em muitos casos, é configurado pela operadora de rede.

Termo	Descrição
Hacker	Definido, para os propósitos deste documento, como agente de ameaças, ator de ameaças, fraudador ou outra fonte de ameaça a um serviço de IoT. Essa ameaça poderia ser oriunda de um criminoso isolado, crime organizado, terrorismo, governos hostis e suas agências, espionagem industrial, grupos de hackers, ativistas políticos, hackers por hobby e pesquisadores, bem como de falhas involuntárias de segurança e privacidade.
Celular	Qualquer tecnologia 3GPP de rede móvel padronizada (por exemplo: GSM, UMTS, LTE (incluindo LTE-M) e NB-IoT).
Nuvem	Uma rede de servidores remotos na internet que hospedam, armazenam, gerenciam e processam aplicativos e seus dados.
Endpoint de alta complexidade	Este modelo de endpoint tem uma conexão persistente com um servidor back-end por meio de um link de comunicação de longa distância, como celular, satélite ou uma conexão física (hardwired), como Ethernet. Consulte a seção 3 deste documento.
Embedded SIM	Um SIM embutido, isto é, que não se destina a ser removido ou substituído no dispositivo e permite a troca segura de perfis.
Endpoint	Um endpoint IoT é um dispositivo físico de computação que executa uma função ou tarefa como parte de um produto ou serviço conectado à internet. Consulte a seção 3 para obter uma descrição das três classes comuns de dispositivos IoT e exemplos de cada classe de endpoint.
Internet das Coisas	A Internet das Coisas (IoT) descreve a coordenação de várias máquinas, dispositivos e aplicações conectados à internet por meio de múltiplas redes. Esses dispositivos incluem objetos comuns, como tablets e eletrônicos de consumo, e outras máquinas, como veículos, monitores e sensores equipados com comunicações de máquina a máquina (M2M) que lhes permitem enviar e receber dados.
Serviço de IoT	Qualquer programa de computador que aproveite dados gerados por dispositivos IoT para executar o serviço.
Ecossistema de Serviços de IoT	O conjunto de serviços, plataformas, protocolos e outras tecnologias necessárias para fornecer recursos e coletar dados dos endpoints implantados em campo. Consulte o documento CLP.11 [1] para obter mais informações.
Provedor de Serviços de IoT	Empresas ou organizações que buscam desenvolver produtos e serviços de IoT inovadores.
Operadora de Rede	A operadora é proprietária da rede de comunicação que conecta o dispositivo endpoint de IoT ao ecossistema de serviços de IoT.
Raiz Organizacional de Confiança	Um conjunto de políticas e procedimentos criptográficos que regem o modo como as identidades, aplicações e comunicações podem e devem ser criptograficamente protegidas.
Ponto de acesso ao serviço	Um ponto de entrada na infraestrutura de back-end do serviço de IoT por meio de uma rede de comunicações.
Módulo de Identidade do Assinante (SIM)	O cartão inteligente usado por uma rede móvel para autenticar dispositivos para conexão à rede móvel e acesso a serviços de rede.
Âncora de confiança	Nos sistemas criptográficos hierárquicos, uma âncora de confiança é uma entidade autorizada na qual a confiança é presumida e não reproduzida.

Termo	Descrição
Base de Computação Confiável	Uma Base de Computação Confiável (TCB) é um conglomerado de algoritmos, políticas e segredos dentro de um produto ou serviço. A TCB atua como um módulo que permite ao produto ou serviço medir sua própria confiabilidade, avaliar a autenticidade das redes pareadas e verificar a integridade das mensagens enviadas e recebidas pelo produto ou serviço, entre outros. A TCB funciona como a plataforma de segurança básica na qual produtos e serviços seguros podem ser construídos. Os componentes de uma TCB podem ser alterados dependendo do contexto (uma TCB em hardware para endpoints ou uma TCB em software para serviços de nuvem), mas as metas, serviços, procedimentos e políticas abstratos devem ser muito similares.
Ambiente Confiável de Execução (TEE)	Um ambiente que funciona ao lado de um rico sistema operacional e fornece serviços de segurança para esse sistema operacional. Existem várias tecnologias que podem ser usadas para implementar um TEE e o nível de segurança alcançado varia adequadamente.
UICC	Uma Plataforma de Elemento Seguro especificada no ETSI TS 102 221, que pode suportar múltiplas redes padronizadas ou aplicações de autenticação de serviço em domínios de segurança separados criptograficamente. Pode ser incorporado em fatores de forma especificados embutidos no ETSI TS 102 671.

1.5 Abreviaturas

Termo	Descrição
3GPP	3rd Generation Project Partnership
AC	Corrente Alternada
API	Interface do Programa de Aplicações
APN	Nome do Ponto de Acesso
BLE	Bluetooth 4.0 de Baixa Potência
BT	Bluetooth
CLP	Programa Connected Living da GSMA
CPE	Equipamento das Instalações para Clientes
CPU	Unidade Central de Processamento
EEPROM	Memória Somente para Leitura Programável e Apagável Eletronicamente
eUICC	eUICC
FIB	Feixe Dirigido de Íons
GBA	Arquitetura Genérica de Bootstrap
GPS	Sistema de Posicionamento Global
GSMA	GSM Association
IoT	Internet das Coisas
IP	Protocolo de Internet
ISM	Industrial, Científico e Médico

Termo	Descrição
LAN	Rede Local
LPWA	Redes de Longo Alcance e Baixa Potência
LTE-M	LTE para Máquinas
MCU	Unidade de Microcontrolador
NB-IoT	Internet das Coisas de Banda Estreita
NVRAM	Acesso Aleatório à Memória Não Volátil
OMA	Open Mobile Alliance
PAN	Área de Rede Pessoal
PSK	Chave Pré-Compartilhada
RAM	Memória de Acesso Aleatório
ROM	Memória Somente de Leitura
SCADA	Controle de Supervisão e Aquisição de Dados
SPI	Interface Serial de Periférico
SSH	Secure Shell, ou SSH
SIM	Módulo de Identidade do Assinante
SRAM	RAM Estática
TCB	Base de Computação Confiável
TTL	Lógica Transistor-Transistor
UART	Transmissor/Receptor Assíncrono Universal

1.6 Referências

Ref	Documento Nº	Título
[1]	CLP.11	Panorama das Diretrizes de Segurança para IoT
[2]	CLP.12	Diretrizes de Segurança em IoT para o Ecossistema de Serviços de IoT
[3]	CLP.13	Diretrizes de Segurança em IoT para o Ecossistema de Endpoints IoT
[4]	CLP.14	Diretrizes de segurança para a IoT para Operadoras de Rede
[5]	OMA FUMO	Objeto de Gerenciamento de Atualização de Firmware OMA www.openmobilealliance.org
[6]	na	Depurador/programador de circuito interno ST-LINK/V2 http://www.st.com/
[7]	na	Iniciativa Mobile IoT https://www.gsma.com/iot/mobile-iot-initiative/
[8]	na	Scanner de Segurança Nmap https://nmap.org/
[9]	CLP.03	Diretrizes de Eficiência de Conexão de Dispositivo IoT https://www.gsma.com/iot/gsma-iot-device-connection-efficiency-guidelines/
[10]	na	Padrões Federais de Processamento de Informações

Ref	Documento Nº	Título
		www.nist.gov/itl/fips.cfm
[11]	na	EMVCo www.emvco.com/
[12]	na	API Aberta Móvel simalliance.org/key-technical-releases/
[13]	GPD_SPE_013	Plataforma Global de Controle ds Elemento Seguro www.globalplatform.org/specificationsdevice.asp
[14]	GPD_SPE_024	Especificação de API para Plataforma Global de Ambiente Garantido de Execução www.globalplatform.org/specificationsdevice.asp
[15]	GPC_SPE_034	Especificação do Cartão de Plataforma Global www.globalplatform.org/specificationscard.asp
[16]	ISO/IEC 29192-1	Tecnologia da informação - Técnicas de segurança - Criptografia leve www.iso.org/obp/ui/#iso:std:iso-iec:29192:-1:ed-1:v1:en
[17]	TS 33.220	Arquitetura Genérica de Autenticação (GAA); Arquitetura Genérica de Inicialização (GBA) www.3gpp.org
[18]	TS 33.222	Arquitetura Genérica de Autenticação (GAA); Acesso a funções de aplicações de rede usando HTTPS www.3gpp.org
[19]	CLP.17	Checklist da GSMA sobre Avaliação de Segurança para IoT https://www.gsma.com/iot/iot-security-assessment/
[20]	TS-0003	Soluções de Segurança oneM2M www.onem2m.org
[21]	3GPP TS33.163	Segurança eficiente de bateria para dispositivos de comunicação de baixo rendimento (MTC) (BEST) www.3GPP.org

2 O desafio da segurança no endpoint IoT

O desafio de segurança apresentado por um serviço de IoT é, em muitos casos, diretamente relacionado às características específicas do endpoint IoT empregado pelo serviço. Por exemplo, muitos endpoints IoT têm as seguintes características que acarretam desafios de segurança específicos:

2.1 Baixo consumo de energia

- Pode ser necessário um baixo consumo de energia para alcançar uma longa vida útil da bateria (vários anos) para um endpoint em local inacessível ou remoto e que não tenha acesso a uma fonte de alimentação permanente, ou que tenha uma fonte de alimentação permanente, porém limitada, como, por exemplo, abastecimento por energia solar.
- Endpoints de baixo consumo de energia geralmente podem realizar apenas operações criptográficas computacionalmente simples (por exemplo, o endpoint só pode ter suporte a operações criptográficas leves definidas na ISO / IEC 29192 [16]) devido aos altos requisitos de consumo de energia associados a operações criptográficas mais avançadas, e podem apenas ter suporte de comunicações de largura de banda limitada, novamente limitando sua capacidade criptográfica.

2.2 Baixo custo

- A estratégia para muitos serviços de IoT exige que o custo do endpoint IoT seja baixo. Isso geralmente resulta em um dispositivo com pouca capacidade de processamento, pouca memória e sistema operacional limitado. O resultado prático é que o dispositivo pode não ser capaz de executar a criptografia "internet-grade".

2.3 Durabilidade (>10 anos)

- Muitos endpoints, particularmente para aplicações civis e industriais (por exemplo, um medidor de gás inteligente), devem ser duradouros. Isso apresenta um desafio porque as escolhas de design da criptografia feitas quando o dispositivo é desenvolvido terão que permanecer robustas por toda a vida útil do dispositivo. Por exemplo, o poder de processamento por dólar empregado por cada hacker ao longo deste período de 10 anos provavelmente aumentará 16 vezes, enquanto as capacidades do dispositivo provavelmente permanecerão estáticas.
- O gerenciamento de dispositivos de longa duração também é um desafio, especialmente se for encontrada uma vulnerabilidade de segurança que não pode ser corrigida no endpoint IoT.

2.4 Fisicamente acessível

- Muitos endpoints IoT são fisicamente acessíveis ao hacker. Todos os componentes e interfaces de hardware desses endpoints são, portanto, potencialmente sujeitos a ataques e devem ser protegidos pelo desenvolvedor.

O resultado concreto disso é que, em muitos serviços IoT, os endpoints IoT não estão diretamente conectados a redes de comunicação de banda larga e muitos deles não possuem recursos de Protocolo de Internet (IP). Por exemplo, um endpoint IoT pode usar um transceptor de rádio industrial, científico e médico (ISM) para transferir dados para um

gateway local de serviço de IoT que, em seguida, retransmite os dados para a rede de comunicação usando IP, complicando o processo de garantir a comunicação ponta a ponta.

Dependendo das capacidades do endpoint IoT e dos riscos de segurança associados, diferentes métodos de segurança, com diferentes graus de complexidade, podem ser aplicados, como explicado no restante deste documento.

3 O modelo do endpoint IoT

Uma vez considerado um conjunto de tecnologias muito díspares, interagindo com o mundo físico e conectando-se a um servidor em algum lugar na internet para orientação e envio de métricas, o modelo endpoint IoT mudou drasticamente. Na engenharia moderna, a tecnologia da IoT foi colapsada por um modelo previsível composto por diversas variantes. O endpoint IoT está se tornando mais previsível também, e pode adotar um desses modelos:

- Endpoint de baixa complexidade
- Endpoint de alta complexidade
- O gateway (ou “Hub”)

No diagrama abaixo algumas configurações comuns de endpoint IoT são exibidas:

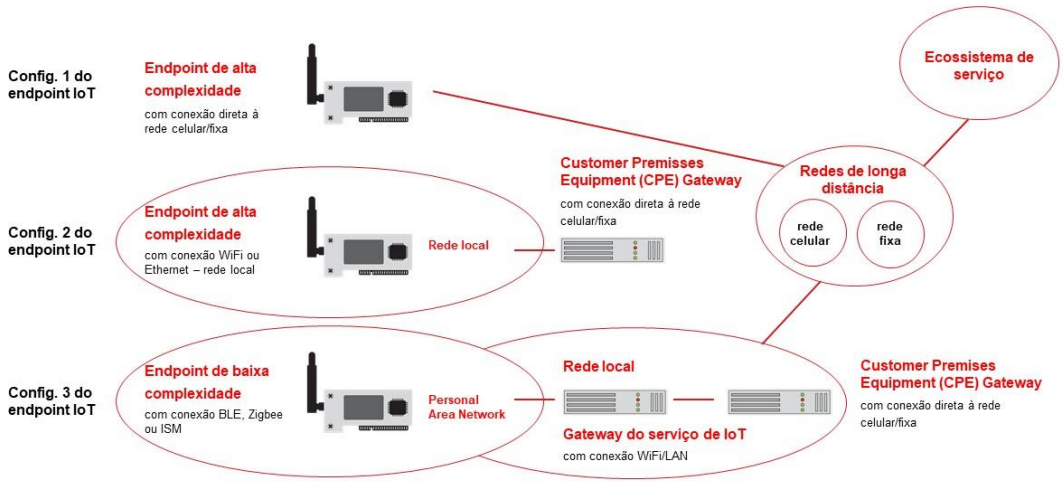

Figura 2 - Exemplo de configurações de endpoint IoT

3.1 O endpoint de baixa complexidade

Esse tipo de endpoint geralmente é um sensor ou dispositivo físico simples, como um interruptor de luz ou uma fechadura com poucas funções. Seu objetivo é atender a um propósito físico singular e fornecer métricas ao ecossistema de serviços ou ao consumidor. Ele geralmente usa uma unidade de processamento muito barata, possivelmente um microcontrolador de oito bits, e um protocolo de rede pessoal de distância curta (PAN) ou

capilar para conectividade, como Bluetooth Low Energy (BLE), Thread ou Zigbee. Geralmente, é de baixa potência e pode ser alimentado por uma pilha, energia solar ou uma pequena bateria de polímero de lítio. Esses dispositivos normalmente estão conectados ao ecossistema de serviço através do gateway do serviço de IoT e do gateway de equipamentos do cliente, como mostrado em 'Exemplo de Configuração do Endpoint nº 3' na Figura 2.

Exemplos de endpoints de baixa complexidade são:

- Vestíveis
- Sensores para segurança residencial
- Beacons de proximidade
- Dispositivos capilares não sem capacidade celular

Devido ao baixo custo dos endpoints de baixa complexidade, as tecnologias de segurança disponíveis para esses dispositivos são mínimas. As tecnologias de segurança que exigem uma quantidade significativa de consumo de energia, custo ou espaço na placa de circuito geralmente não estão disponíveis para esses sistemas. No entanto, endpoints de baixa complexidade ainda podem usar pequenas âncoras de confiança como solução custo-eficiente para implementar uma estrutura de segurança robusta.

3.2 O endpoint de alta complexidade

Esse modelo de endpoint geralmente possui uma conexão permanente com um servidor back-end por um link de comunicação de longa distância, como celular (incluindo Redes LPWA) (consulte 'Exemplo de Configuração Nº 1' na Figura 2), ou conecta usando Wi-Fi ou Ethernet por meio de um gateway CPE (consulte 'Exemplo de configuração de terminal 2' na figura 2). O dispositivo pode conter um processador rudimentar, até mesmo um microcontrolador de oito bits, mas é capaz de executar uma unidade de processamento mais robusta, pois está diretamente conectado a uma fonte de alimentação de corrente alternada (AC) ou contém uma bateria e tem acesso a um sistema de recarga de bateria. Alguns endpoints de alta complexidade se comunicam por protocolos capilares, mas exigem mais energia para executar o aplicativo local de maneira eficiente, como um dispositivo de transmissão de áudio.

Exemplos de endpoints de alta complexidade são:

- Sistemas de iluminação conectados
- Eletrodomésticos como geladeiras ou máquinas de lavar roupa
- Sistemas de controle industrial (por exemplo, Sistemas de Supervisão e Aquisição de Dados**, ou** SCADA)
- Veículos conectados via dispositivos de rastreamento e monitoramento OBD2

Endpoints de alta complexidade são capazes de consumir mais corrente, geralmente implementam processadores mais robustos e têm mais espaço na placa de circuito disponível para tecnologias de segurança. Como resultado, muito mais pode ser feito com endpoints de alta complexidade. Esses dispositivos podem usar quase qualquer tipo de âncora de confiança. Como resultado, são facilmente capazes de implementar um modelo personalizado de chave pré-compartilhada (PSK) ou de base de computação confiável assimétrica (TCB), conforme descrito mais adiante neste documento.

3.3 O gateway (ou 'Hub')

Um gateway é um dispositivo tipicamente conectado a uma fonte dedicada de energia, e que tipicamente gerencia a comunicação entre endpoints de baixa complexidade e os sistemas back-end que os alimentam. O gateway gerencia links de comunicação de longa distância, como celular (incluindo LPWA), satélite, linha fixa, fibra ou Ethernet. Ele aceita comandos dos sistemas de back-end que residem no ecossistema de serviços e os converte em mensagens consumíveis pelos endpoints de baixa complexidade.

Embora a principal função de um gateway IoT seja rotear mensagens de e para endpoints compactos, ele também é capaz de realizar tarefas críticas, como:

- Descoberta de dispositivos
- Implantação do driver de rede
- Funcionalidade de gerenciamento
- Monitoramento de tempo de execução
- Autenticação e Segurança, como configuração GBA ou TLS

Embora os gateways sejam tecnicamente endpoints, eles podem não ser necessariamente gerenciados pelo usuário final e sim pelo provedor de serviços de IoT ou pela operadora de rede (veja abaixo). Independentemente disso, os gateways também podem ser desenvolvidos como endpoints de alta complexidade para fazer uso mais eficiente da distribuição de um uplink para vários endpoints de baixa complexidade em uma rede localizada.

Como os endpoints de alta complexidade, os gateways têm maior poder de processamento, consumo de corrente e normalmente têm mais espaço disponível na placa de circuito. Isso permite que os gateways IoT implementem soluções complexas de base de computação confiável e tecnologias como clientes de autenticação GBA com relativa facilidade.

Esses atributos do gateway também permitem que eles incorporem várias tecnologias de comunicação para rotear mensagens entre tipos diferentes de dispositivos em rede. Isso permite a comunicação entre endpoints que normalmente seriam incapazes de trocar mensagens de maneira efetiva. Desta forma, os gateways funcionam como um ponto de agregação para dispositivos dentro do ecossistema local, permitindo que eles se comuniquem entre si e, se necessário, com os ecossistemas de rede e de serviços.

Normalmente existem dois tipos de gateway - um "gateway de serviço de IoT" e um "gateway CPEs (Customer Premises Equipment)". A diferença é explicada abaixo:

1. Um "gateway de serviço de IoT" é fornecido pelo provedor de serviços de IoT. Pode ser de propriedade do usuário final, mas normalmente é gerenciado pelo provedor de serviços IoT. Esse gateway é normalmente utilizado como um hub para conectar endpoints de baixa complexidade ao ecossistema de serviços (seja diretamente por meio de uma conexão fixa ou celular, ou por meio de um gateway CPE), em que o usuário final adquire um serviço de um provedor de serviços de IoT.

2. Um "gateway CPE" é fornecido por uma operadora de rede. Este é tipicamente um roteador de banda larga conectado à internet por redes fixas ou celulares. Ele pode

ser usado em ambientes residenciais ou corporativos. Nesta configuração, o gateway é geralmente gerenciado e configurado pela operadora de rede.

3.4 O modelo abrangente

Independentemente de qual tipo de endpoint está sendo avaliado ou projetado, todos eles têm modelos semelhantes de subcomponentes sob uma perspectiva de hardware e logística:

- Uma Unidade Central de Processamento (CPU) deve executar o código do aplicativo
- A CPU deve carregar/armazenar dados e o código executável de/para armazenamento permanente
- A CPU deve computar dados no armazenamento temporário
- Uma Base de Computação Confiável deve ser usada para autenticar o ambiente
- O dispositivo deve se comunicar com seu ecossistema de IoT

Destaque-se que os endpoints de baixa complexidade têm menos capacidade de armazenamento e computação que endpoints de alta complexidade ou gateways. Eles normalmente têm menos capacidade de segurança também.

O aspecto mais importante do modelo abrangente é que cada tipo de dispositivo endpoint possui uma tarefa principal: definir uma plataforma confiável, de alta qualidade e segura para a execução de um aplicativo específico. Em outras palavras, semelhante a plataformas de computação mais complexas, como smartphones, servidores em nuvem e mainframes, antes que um aplicativo de alta qualidade possa ser executado com segurança ou interaja de forma segura com seus pares, a equipe de engenharia deve garantir que o hardware ofereça uma plataforma confiável à aplicação.

Os endpoints IoT, por natureza, participam de uma rede com outros endpoints. Eles não são dispositivos autônomos que realizam uma ação sem a influência ou participação de um serviço de supervisão. Para aumentar a confiabilidade de um determinado dispositivo e diminuir a responsabilidade legal devido a falhas de segurança ou confiabilidade, cada endpoint deve ser desenvolvido com a ideia de que a confiabilidade em todo o ecossistema IoT começa com a construção de seu hardware.

Sob essa perspectiva, o mais fácil é desenvolver o tipo de dispositivo endpoint que deve se comportar de maneira confiável e segura, com alta qualidade, pois espera-se que ele participe de uma rede que pode alcançar milhões de outros dispositivos. A maneira como um único endpoint se comporta certamente terá um efeito em todo o seu ecossistema de IoT. Como resultado, os engenheiros devem considerar as implicações do design da arquitetura, muito além dos atributos físicos associados a um determinado dispositivo embutido. Os engenheiros devem pensar em termos das necessidades de segurança, confiabilidade e qualidade de todo o ecossistema IoT.

4 O modelo de segurança

A segurança no endpoint pode ser avaliada de uma perspectiva do componente. Ao avaliar cada componente necessário para construir um determinado endpoint, o engenheiro e o hacker podem criar um conjunto de ataques prováveis que poderá resultar no comprometimento total do sistema sem grande esforço.

Usando o modelo abrangente definido acima, os componentes usados podem ser avaliados em um alto nível. A perspectiva de alto nível de cada componente direcionará um analista às tecnologias que são comumente usadas e que provavelmente serão protegidas de maneira inadequada. Ao priorizar esses componentes com menor nível de especialização, equipamentos e custos necessários para ter sucesso, um analista ou um invasor pode criar um modelo de ataque que avalie rapidamente qualquer endpoint para buscar falhas de segurança.

No ecossistema do endpoint, existem várias camadas de ameaças que serão investigadas pelos invasores, dependendo de seus recursos, acesso à infraestrutura e especialização. Essas camadas de ameaças são:

- Comunicações de rede
- Serviços acessíveis de rede
- Acesso ao terminal
- Canal local de comunicações
- Acesso ao chip

4.1 Ataques às comunicações de rede

A primeira e mais simples etapa na tentativa de comprometer um endpoint IoT geralmente envolve fragilidades no modelo de comunicação. Os analistas devem observar se o modelo de comunicação incorpora as melhores práticas de segurança de comunicações. Se o analista puder facilmente capturar credenciais de login, tokens de comunicação ou outros identificadores que o Ecossistema de Serviços usará para identificar o endpoint, eles comprometeram o dispositivo.

Esta estratégia pode variar do extremamente simples ao extremamente difícil. A razão para isso é o acesso do analista ou invasor aos dados de texto simples que passam pelo canal de comunicação. Um analista suficientemente especializado já terá tecnologia para interceptar comunicações para BLE, 802.15.4 e outros protocolos populares. Como observar ou realizar um ataque man-in-the-middle contra as comunicações a um endpoint normalmente requer pouca ou nenhuma alteração no endpoint, o invasor está em uma posição de superioridade. Muito pouco esforço e trabalho são necessários para implementar esse tipo de ataque.

Entretanto, se o modelo de comunicação usar melhores práticas para impor a confidencialidade e a integridade dos dados, o invasor terá exponencialmente mais dificuldade de acessar dados valiosos. Isso fará com que o invasor passe para outro modelo mais fácil de ataque.

4.2 Ataques a serviços de rede acessíveis

O próximo passo no ataque a um endpoint IoT é uma avaliação dos serviços de rede que são abertos. Na primeira etapa, as mensagens de saída provenientes do endpoint são capturadas para identificar se os dados imediatamente utilizáveis estão acessíveis nas mensagens. Isso permite que um invasor reduza a quantidade de trabalho necessária para extrair dados do próprio endpoint. Se o modelo de segurança de comunicações de saída estiver correto, os serviços de rede serão verificados para avaliar se o sistema operacional do endpoint pode ser acessado ou manipulado a partir da rede.

Uma avaliação será realizada com uma ferramenta como o NMap [8] para determinar se as portas de rede estão abertas. Se a topologia de rede não for compatível com IP, o que é comum em redes BLE ou IEEE 802.15.4, o hacker ainda pode usar ferramentas prontamente acessíveis para se conectar ao endpoint por meio do protocolo apropriado de rádio.

O invasor tentará então enviar mensagens ao endpoint para determinar se este pode ser manipulado para executar comandos ou fornecer acesso remoto ao console para o sistema operacional. Um método comum é avaliar se uma interface de login de rede, como Secure Shell (SSH) ou telnet, está disponível. Se as credenciais de login padrão forem usadas, o hacker poderá efetuar login no endpoint. Isso permitirá que o hacker manipule o sistema operacional local e, potencialmente, abuse das vulnerabilidades locais para obter privilégios e extrair dados do dispositivo.

Outro exemplo comum compreende o abuso de serviços web mal pensados, em que comandos podem ser injetados em scripts CGI (Common Gateway Interface) que não eliminam adequadamente os caracteres de controle dos campos de entrada do usuário, resultando na execução de código no sistema operacional local.

4.3 Ataques de acesso ao terminal

O acesso ao terminal não é exatamente um ataque, é uma estratégia. Normalmente, os terminais precisam estar habilitados nos endpoints para fornecer aos desenvolvedores e técnicos a garantia de qualidade (QA) e a capacidade de diagnosticar anomalias em hardware ou software. No entanto, as informações fornecidas por um terminal são muito valiosas para um invasor. Além disso, um terminal pode fornecer a ele a capacidade de efetuar login no sistema do endpoint, local ou remotamente.

Normalmente, os terminais locais de hardware podem ser encontrados em dispositivos endpoint:

- Procurando por um conector de 5 pinos na placa de circuito indicando uma porta serial TTL
- Procurando as especificações da CPU ou MCU e identificando os pinos UART

Um multímetro pode ser usado para identificar uma porta TTL, pois os pinos obedecerão à especificação típica de tensão para TTL. Alternativamente, um analisador lógico pode ser usado para adivinhar a taxa de transmissão de qualquer dado serial que trafegue pelos pinos do hardware. O analista será capaz de discernir se um terminal estará disponível no hardware local.

Em muitos casos, o simples acesso a uma porta de terminal permite que um analista tenha acesso direto a um prompt de comando no endpoint. Em outros casos, as credenciais de login são necessárias, mas são normalmente previsíveis. Se outra pessoa na internet tiver identificado as credenciais de login e todas as credenciais de login do endpoint casarem, tudo o que um analista precisa fazer é realizar uma pesquisa online no Google para ver se outra pessoa postou estas credenciais.

O acesso remoto ao terminal pode ser obtido por meio de protocolos de diagnóstico de rede, protocolos de acesso ao terminal (por exemplo, SSH ou telnet) ou por outros métodos. Esses métodos de acesso devem ser avaliados para determinar se um invasor pode manipular o canal de acesso, concedendo, assim, a ele o acesso a um terminal remoto.

4.4 Ataques ao canal local de comunicações

Se um prompt de comando não puder ser obtido por meio de um terminal, o invasor ou analista deverá inspecionar o hardware para determinar se o endpoint está facilmente comprometido. Isso assume muitas formas diferentes, mas há medidas simples que devem ser tomadas:

- Se a mídia gravável está presente e pode ser alterada
- Os dados criptográficos transitam desprotegidos no barramento do hardware
- Mensagens podem ser injetadas no circuito de hardware para influenciar o comportamento do aplicativo ou do sistema operacional em favor do invasor

O ataque mais simples é identificar se a mídia gravável está presente. Isso pode ser uma mídia simples de alterar, como uma placa de memória externa gravável (SD/MMC) ou um chip NVRAM ou EEPROM que podem ser alterados com modificações do aplicativo ou de sua configuração para permitir acesso ao prompt de comando ou acesso a tokens seguros armazenados.

Se este vetor estiver adequadamente protegido, o analista determinará se os dados criptográficos trafegam nos barramentos de hardware. Isso poderia envolver o uso de um analisador lógico para interceptar mensagens entre uma EEPROM e uma CPU, um microcontrolador e um adaptador de rede conectado SPI, ou outros ataques. Esses ataques podem variar do extremamente simples e rápido aos complexos e desgastantes, dependendo da complexidade do ataque e da tecnologia explorada.

Se o invasor não puder interceptar dados valiosos usando o método descrito, ele poderá tentar injetar mensagens nos barramentos de hardware para alterar o comportamento de um aplicativo em execução no endpoint. Esse é um ataque difícil e que requer um alto grau de especialização, equipamentos e a capacidade de avaliar os dados específicos do aplicativo e seu contexto.

4.5 Ataques de acesso ao chip

Se os ataques acima forem muito complexos ou desgastantes, o invasor deve passar para ataques ainda mais complexos contra o hardware. Isso normalmente envolve abusar da segurança do chip ou dos vários componentes da placa de circuito. Isso pode incluir:

- Decapando o microprocessador ou a CPU
- Extraindo dados da EEPROM interna ou NVRAM
- Interceptando mensagens internas da SRAM
- Executando uma análise Raio X ou engenharia reversa FIB

Todos esses ataques exigem um alto grau de habilidade, conhecimento de engenharia eletrônica e equipamentos sofisticados. Embora a maioria das organizações não precise temer que um hacker use essas metodologias para fazer engenharia reversa de seus produtos, ainda é uma possibilidade importante a ser considerada. A razão é que esses ataques só precisam ser realizados uma vez se os endpoints não forem provisionados com dados criptográficos exclusivos.

Se eles não forem provisionados com senhas criptografadas exclusivas, um ataque nesta classe extrairá dados que podem afetar toda a linha de produtos. Isso é um risco significativo, porque, se as senhas forem divulgadas ao público por algum motivo, a tecnologia estará sujeita a ataques e abusos até que um patch seja liberado, caso seja possível.

5 Perguntas frequentes sobre segurança

O tema de segurança do endpoint é abordado neste documento em recomendações agrupadas por ordem de prioridade. Mas, para uso prático, é melhor avaliar as recomendações de um ponto de vista prático. Os engenheiros geralmente criam uma lista de recomendações com base em uma meta tecnológica ou influenciados por decisão negocial. Esta seção descreve objetivos comuns de uma perspectiva do endpoint e quais recomendações são relevantes para atingir essas metas.

5.1 Como combatemos a clonagem?

Proteger a propriedade intelectual é uma meta importante para as empresas modernas. As tecnologias de hardware, firmware e comunicações usadas para criar um produto endpoint custam tempo, experiência e investimentos, de modo que empresas dificilmente gostariam de ceder a outra marca ou empresa inescrupulosa. No entanto, não importa o que uma empresa faça, alguém pode usar exatamente os mesmos componentes de hardware para criar um “rip off” ou “clone” de um determinado produto. Não há nada que a empresa possa fazer para evitar isso fora dos contratos e parcerias legais. No entanto, existem maneiras econômicas de impedir que alguém use um clone desse tipo.

A criação de autenticação nas comunicações do endpoint garantirá que cada um deles seja comprovado criptograficamente pelo provedor de serviços de IoT como sendo do fabricante. Sempre que os serviços de back-end, ou um endpoint ponto a ponto, comunicar-se com outro endpoint, ele poderá diferenciar entre um endpoint e um clone forçando-o a se autenticar. Se o dispositivo não puder fazê-lo, o peer ou o serviço poderá rejeitar o endpoint. Isso requer as seguintes recomendações para funcionar:

- Autenticando uma Identidade do Endpoint
- Autenticação Mútua Impropriamente Projetada ou Não Implementada

5.2 Como posso assegurar a identidade do endpoint?

Para autenticar adequadamente um endpoint, o engenheiro deve poder confiar na identidade criptográfica deste. Isso é mais complexo do que parece e requer uma combinação de processos, políticas e tecnologia para atingir seu objetivo. Isso está elaborado na recomendação “Implementar uma Base de Computação Confiável”, mas a maneira como os tokens de autenticação são codificados em um endpoint determinará a segurança geral do sistema.

Em muitas arquiteturas de endpoint, um invasor pode simplesmente copiar tokens criptográficos (se houver algum) do dispositivo de destino para imitá-lo. Se cada endpoint fabricado pelo provedor de serviços de IoT utilizar o mesmo conjunto de tokens criptográficos, o invasor poderá burlar qualquer dispositivo simplesmente comprometendo um único conjunto de tokens.

Assim, construir a TCB apropriada requer as seguintes recomendações:

- Implemente uma base de computação confiável

- Utilize uma âncora de confiança
- Use uma senha confiável resistente à violação
- Use uma API para a TCB
- Use um gerador certificado de números aleatórios
- Utilize embalagem de produto resistente a adulterações
- Aplicar Confidencialidade e Integridade de/para a âncora de confiança

5.3 Como reduzo o impacto de um ataque contra uma âncora de confiança?

Também é importante observar que o modo como um dispositivo é fabricado e fornecido tem um efeito drástico na segurança de um endpoint em produção. O processo de fabricação determinará se os endpoints estão codificados em segurança com chaves. O processo de execução e fornecimento determinará como um endpoint está associado a um consumidor específico e se o dispositivo pode estar comprometido antes ou depois de uma associação ser feita.

- Considere a segurança da cadeia de fornecedores
- Personalize cada endpoint antes da Execução
- Provisionar cada endpoint de forma única
- Privacidade e identificadores únicos de endpoint

5.4 Como reduzo a possibilidade de falsificação do endpoint?

Após a clonagem de dispositivos por razões comerciais, um ataque desejável, do ponto de vista do invasor, é a falsificação de uma identidade ou de um determinado dispositivo. Isso pode ou não estar diretamente associado ao ataque de um indivíduo em particular. Pode ser simplesmente a falsificação de um dispositivo com o objetivo de bypass um controle de segurança, como um bloqueio digital ativado por bluetooth.

Independentemente da lógica, o combate a esse ataque pode ser obtido usando uma TCB, personalização, autenticação e também:

- Perfect Forward Secrecy (“Segurança Futura Perfeita”)
- Bloqueio de áreas críticas de memória

5.5 Como impeço a falsificação de serviços ou pares?

Toda rede IoT é composta não apenas por endpoints, mas também por serviços de rede e pares. Os endpoints precisam ser autenticados pelos serviços, mas os serviços devem também ser autenticados pelos endpoints. Isso garante que serviços críticos, como atualizações de aplicativos, não possam ser subvertidos para comprometer ainda mais a rede.

- Segurança de comunicações no endpoint
- Perfect Forward Secrecy (“sigilo daqui para frente”)
- Use um gerador certificado de números aleatórios
- Atualizações over the air (OTA)
- Autenticação Mútua Impropriamente Projetada ou Não Implementada

- Coleta não autorizada de metadados

5.6 Como impeço a falsificação de firmware e software?

Uma vez que uma raiz de confiança tenha sido estabelecida, o endpoint pode se autenticar a partir de um componente fidedigno. Isso permite que o endpoint estabeleça uma base de confiança e garanta que um novo release do aplicativo não tenha sido alterado de forma não intencional (por meio de NVRAM com defeito, por exemplo) ou intencionalmente por um invasor. Realize isso por:

- Plataforma de execução mínima viável (Roll-Back de aplicação)
- Assinar o aplicativo criptograficamente com imagens
- Carregamento de inicialização fora da EEPROM interna
- Bloqueio de areas críticas de memória
- Carregadores de inicialização inseguros
- Use embalagem de produto resistente a adulterações

5.7 Como reduzo a possibilidade de execução remota de código?

Se a adulteração do firmware ou software físico não produzir resultados adequados, o invasor pode passar para ataques mais complexos, como execução de código no carregador de inicialização ou aplicativos que se comunicam por meio de barramento ou interfaces de rede. Se todos os pares na rede forem autenticados, conforme descrito anteriormente neste capítulo, será muito mais desafiador para um invasor inserir conteúdo mal-intencionado. No entanto, a maioria dos dispositivos exige um pouco de comunicação pública para interagir com dispositivos de outras organizações. Portanto, eles podem não ser capazes de aplicar adequadamente as restrições sobre a origem dos dados.

Assim, a entrada de dados no sistema de computador a partir de interfaces remotas e físicas deve ser amplamente examinado. Para limitar o potencial de exploração de um aplicativo e limitar a exposição depois que um aplicativo for comprometido, considere o seguinte:

- Reforce a proteção à memória
- Use a memória interna para senhas
- Atualizações over the air
- Rode aplicativos com níveis apropriados de privilégios
- Imponha uma separação de tarefas na arquitetura de aplicativos
- Reforce a segurança da linguagem
- Reforce as otimizações de segurança no nível do sistema operacional
- Segurança na interface de usuário
- Audite códigos de terceiros

5.8 Como desabilito a depuração não autorizada ou manipulação da arquitetura?

Um hacker com conhecimento de arquitetura e acesso a ferramentas de depuração normalmente tentará manipular utilitários de depuração e diagnóstico padrão para obter

acesso a senhas do sistema ou para alterar ou inserir código benéfico. Restringir a capacidade de um invasor de fazer isso diminuirá a possibilidade de ataques rápidos e furtivos que podem não ser detectados por um usuário.

- Use uma âncora de confiança resistente à violação
- Registro de logs e diagnósticos
- Bloqueio de áreas críticas da memória
- Detecção de anomalias
- Use uma embalagem de produto resistente a adulterações
- Desabilite tecnologias de depuração e testes
- Segurança na interface de usuário

5.9 Como devo lidar com ataques de canal lateral?

Quando um invasor age fora das opções típicas, ele procura ataques mais heterodoxos para extrair informações de um dispositivo. Esses ataques avaliam como o hardware se comporta para verificar se um padrão no comportamento pode se igualar a um valor, como um ou zero, ou uma instrução específica. Isso, com o tempo, dará ao analista a capacidade de reverter a engenharia dos dados processados pelo sistema embarcado.

Além disso, o invasor pode usar uma tecnologia cara de análise para extrair informações do dispositivo ou para construir circuitos extremamente pequenos que executam conexões por meio de camadas de segurança no silício. Embora esses ataques sejam extremamente difíceis de combater, há algumas coisas que o implementador pode fazer para dissuadir ataques:

- Personalize cada endpoint antes da execução
- Use a memória interna para senhas
- Use uma embalagem de produto resistente a adulterações
- Use memória contaminada via ataques localizados em periféricos
- Implemente limites de bloqueio de ambiente
- Reforce os avisos de bloqueio de energia
- Retire de atividade e sunsetting de dispositivos
- Erradique componentes espionados e bridges não confiáveis
- Derrote um ataque Cold-Boot
- Combata feixes de íons e raios X

5.10 Como devo implementar o gerenciamento remoto seguro?

O gerenciamento remoto é uma parte essencial do ciclo de vida do endpoint IoT que deve ser salvaguardado para garantir que o canal usado para gerenciamento e administração não possa sofrer investidas. Este não é apenas um problema com invasores de terceiros desconhecidos. Abusos internos também podem ocorrer no círculo do cliente ou no provedor de serviços de IoT.

- Gerenciamento de senhas do endpoint
- Administração remota do endpoint
- Registro de logs e diagnósticos

- Perfect Forward Secrecy (“sigilo daqui para frente”)
- Use uma APN privada

5.11 Como detecto endpoints comprometidos?

Dependendo da arquitetura do endpoint, pode ser quase impossível determinar se o hardware ou firmware foi adulterado se o dispositivo estiver se comportando normalmente. No entanto, um dispositivo comprometido pode ser detectado por comportamento anômalo, desde que a infraestrutura esteja rastreando, registrando e alertando quando anormalidades são detectadas. Considere as seguintes recomendações:

- Detecção de anomalias
- Use uma embalagem resistente a adulterações
- Reforce os avisos de bloqueio de energia

5.12 Como faço para instalar um dispositivo com segurança sem uma conexão back-end?

Há certos momentos em que uma conexão com um ambiente back-end não está disponível e nem é desejado. Nesses ambientes, a segurança se torna mais um desafio devido à óbvia incapacidade de gerenciar chaves de segurança, identidades e mecanismos de autenticação dinâmica. No entanto, um nível razoável de segurança pode ser alcançado. Considere o seguinte:

- Implemente uma Base de Computação Confiável
- Definindo uma raiz de confiança organizacional
- Personalize cada endpoint antes da execução
- Perfect Forward Secrecy (“sigilo daqui para frente”)
- Autenticação e Identidade de endpoint
- Ambientes sem conectividade back-end
-

5.13 Como garanto a privacidade do meu cliente?

A privacidade do cliente é uma questão complexa que requer uma análise profunda não apenas da tecnologia Endpoint, mas de todo o produto ou serviço IoT. Cada componente do sistema geral deve ser analisado quanto a potenciais brechas de privacidade. Analise as recomendações a seguir para obter mais informações sobre a aplicação da privacidade:

- Perfect Forward Secrecy (“sigilo daqui para frente”)
- Segurança das comunicações do endpoint
- Gerenciamento de privacidade
- Privacidade e identidade únicas de endpoint
- Utilize uma APN privada
- Coleta não autorizada de metadados
- Riscos de segurança não óbvios (“vendo através das paredes”)
- Intercepção legal

5.14 Como garanto a proteção do usuário ao impor privacidade e segurança?

Segurança é um tópico que deve ser considerado no contexto do aplicativo, sua finalidade, o ambiente pretendido no qual o aplicativo estará, o tipo de consumidor e a tecnologia de comunicação utilizada. Muitas vezes pode parecer que existem trade-offs entre a segurança física e a cibernética. Isso pode não ser verdade; em vez disso, o modelo de arquitetura pode precisar ser alterado para manter a ambas. Sempre que possível, a segurança física não deve ser descartada em favor da cibernética. Ambas devem ser aplicadas, sempre que possível. Embora essa seja uma recomendação filosófica, é importante que a segurança seja constantemente revisada pela equipe de engenharia. Considere as seguintes recomendações para iniciar uma discussão sobre segurança física na IoT:

- Análise crítica de segurança física
- Negação intencional ou não intencional de serviço
- Intercepção legal
- Examine a cadeia de fornecedores

5.15 Quais problemas não deveria esperar resolver?

Em todo sistema existem riscos que não podem ser resolvidos devido às leis da física, custo ou simplesmente falta de soluções tecnológicas. Algumas dessas questões são mencionadas aqui:

- Negação intencional ou não intencional de serviço
- Desativar componentes espionados ou bridges não confiáveis
- Riscos de segurança não óbvios (“vendo através das paredes”)
- Combate a feixes de íons e raios X
- Examine a segurança da cadeia de fornecedores
- Intercepção legal

6 Recomendações fundamentais

As seguintes recomendações, sem as quais o endpoint terá um perfil de segurança incompleto, são fundamentais para definir uma arquitetura segura para o endpoint, de modo a evitar comprometimento por terceiros.

6.1 Implemente uma base de computação confiável no endpoint

O primeiro passo para proteger qualquer sistema embutido é a definição da Base de Computação Confiável (TCB, da sigla em inglês). No contexto de um endpoint (ou dispositivos embarcados similares), uma TCB é um conjunto composto de hardware, software e protocolos que garantem a integridade do endpoint, executa autenticação mútua com redes pareadas e gerencia a segurança das comunicações e das aplicações.

O núcleo da TCB é a âncora de confiança, uma tecnologia de hardware segura que armazena e processa senhas criptografadas, como chaves pré-compartilhadas (PSK, da sigla em inglês) ou chaves assimétricas. Âncoras de confiança, como um UICC, podem ser usadas para autenticar não apenas pares em comunicações de rede, mas podem ser aumentadas para armazenar dados úteis para a aplicação de segurança do endpoint.

Depois que a âncora de confiança é selecionada e integrada à solução do endpoint, as bibliotecas podem ser escolhidas ou desenvolvidas para integrar a senha confiável ao conjunto da TCB. A TCB permitirá que o sistema operacional e as aplicações principais do endpoint gerenciem mais facilmente a segurança como um todo não apenas do dispositivo, mas da rede.

É importante que a equipe de engenharia escolha a âncora de confiança correta para a solução, pois cada combinação de âncora de confiança e TCB resultará em um nível diferente de segurança. Algumas combinações e implementações de chaves de segurança resultarão em uma falsa sensação de segurança.

As variações mais comuns de uma base de computação confiável, na ordem de "menos seguras" para "mais seguras", são:

- Não Implementada (apenas texto)
- Chave Pré-Compartilhada Estática (PSK, da sigla em inglês)
- Chave Pública Estática
- PSK Personalizada
- Chave Pública Personalizada

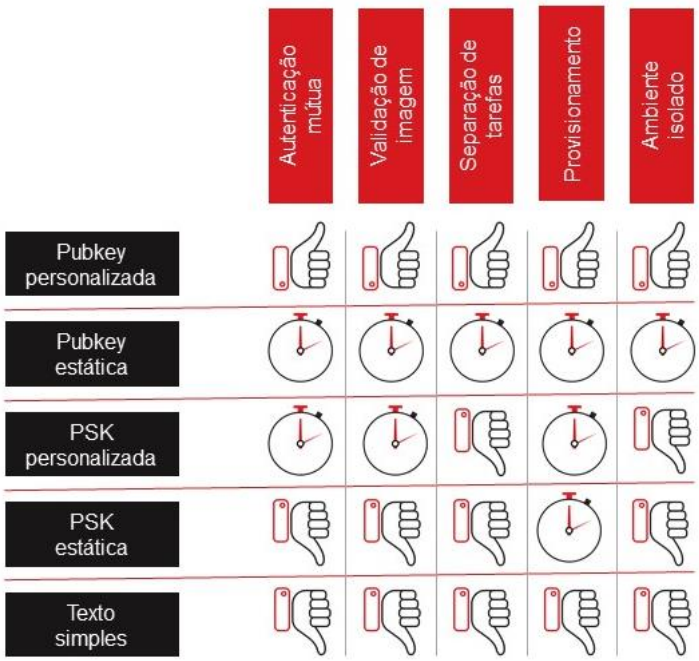

Figura 3 – Garantias de segurança fornecidas por tipo de TCB

Observe a figura acima. Neste diagrama, as capacidades de cada variante da TCB recebem um valor. Um ícone de polegar para baixo indica que o modelo TCB não pode acomodar a estratégia de segurança mostrada na linha superior. Um ícone de cronômetro mostra que a estratégia de segurança pode ser usada, mas estará sujeita a uma quebra de segurança no médio prazo. Um ícone de aprovação mostra que a estratégia de segurança pode ser implementada de forma sólida e que a vida útil da estratégia de segurança provavelmente será duradoura.

Embora uma TCB possa ser usada para proteger muitos aspectos do produto e serviço de IoT de modo geral, este documento focaliza cinco conceitos principais:

- Validação de imagem para execução
- Autenticação mútua de redes pareadas
- Separação de tarefas dentro da arquitetura de segurança em IoT
- Provisionamento e personalização
- Segurança de ambiente isolado (ou segurança do site sem conexão)

Uma TCB que implementa a validação de imagem para execução protege o ndpoint verificando criptograficamente cada imagem a ser carregada e executada pelo dispositivo. Esse processo começa no gerenciador de boot, que deve validar criptograficamente o próximo estágio de execução (geralmente o kernel do sistema operacional). O gerenciador de boot também pode validar a imagem do sistema operacional ou uma imagem de aplicação de firmware armazenado na NVRAM.

Uma TCB que implementa a autenticação mútua de redes pareadas ajuda a fornecer uma raiz de confiança para a autenticação de componentes de rede e se autentica criptograficamente nestes. Isso aumenta a probabilidade de que as redes pareadas representem as identidades que eles afirmam representar. Por exemplo, se o par de rede alega oferecer um serviço de atualização de firmware, a TCB autenticaria o par como parte da rede principal do provedor de serviços de IoT antes de aceitar as atualizações de firmware.

Uma TCB que implementa uma *separação de tarefas* usa uma hierarquia de chaves para identificar componentes ou serviços variados nas ofertas do provedor de serviços de IoT. Por exemplo, um conjunto de chaves criptográficas poderia representar um serviço de atualização de firmware, enquanto um segundo conjunto de chaves criptográficas poderia representar um serviço de "push". Como esses serviços têm funcionalidades completamente diferentes, eles não devem usar as mesmas chaves criptográficas e identidades para comunicação. Como tal, a TCB deve gerenciar e verificar cada identidade para separar cada serviço ou função de outro. Isso reduz a capacidade de um hacker comprometer toda a infraestrutura do serviço de IoT se uma das chaves criptográficas for comprometida. Em outras palavras, se um invasor comprometer a chave do “serviço push”, ele não terá a mesma capacidade de personificar o serviço de atualização de firmware.

Uma TCB que implementa *personalização e provisionamento* garante que o endpoint tenha uma identidade que seja criptograficamente única entre todos os outros de seu tipo. Também garante que todas as identidades de comunicação sejam protegidas para reduzir o potencial de vazamento ou rastreamento de privacidade.

Uma TCB que implementa a *segurança de ambiente isolado* impõe políticas e procedimentos que validam a autenticidade dos pares e a confidencialidade e integridade dos dados, mesmo que não haja serviço de back-end para ajudar no processo. Em outras palavras, se a comunicação com os serviços de back-end for interrompida por um período prolongado, o ecossistema local de IoT ainda poderá funcionar com um alto grau de segurança. Embora a integridade de ambientes isolados seja prejudicada com o tempo, uma TCB bem pensada, que implemente a segurança de ambiente isolado, pode fortalecer a resiliência da rede e prolongar a quantidade de tempo pelo qual o ambiente pode ser considerado seguro.

Nesse contexto, “personalizado” é indicativo de um conjunto exclusivo de chaves associadas a uma chave de segurança específica. O processo de personalização inclui a geração e instalação de chaves únicas, a associação das chaves com chip único, e a disseminação segura dessas informações e dos metadados relevantes para as autoridades competentes. Isso garante que cada chip tenha uma identidade criptográfica única. E “estático”, neste contexto, refere-se ao mesmo conjunto de chaves usado para cada endpoint.

Embora as TCBs possam ser usadas para solucionar praticamente qualquer problema de segurança que um sistema embutido possa ter, há vários problemas importantes que uma TCB deve ser capaz de resolver:

- Validação de imagem da aplicação do endpoint

- Autenticação de rede e/ou autenticação do par
- Separação de tarefas
- Provisionamento e personalização
- Ambiente isolado de provisionamento e comunicação (site sem conexão)
- Randomização

Embora seja óbvio que a escolha de não implementar uma TCB resulte em falha de segurança, há sutilezas nas outras implementações usuais da TCB que devem ser abordadas. Se essas sutilezas não forem abordadas, elas podem resultar em falhas substanciais de segurança.

6.1.1 Modelos de âncoras de confiança

6.1.1.1 Chaves estáticas

Uma implementação de chave estática, seja ela PSK ou chaves assimétricas, é definida como uma solução na qual cada endpoint utiliza o mesmo segredo criptográfico para resolver um determinado problema. Embora diferentes chaves possam ser usadas para resolver diferentes problemas, o conjunto de chaves ainda é o mesmo para cada um deles.

Este modelo parece assegurar que a TCB resolva de modo eficaz cada problema. No entanto, a vida útil da solução como um todo pode variar de longa a extremamente curta. Dependendo da segurança da âncora de confiança, do algoritmo criptográfico e do tamanho de chave escolhidos, invasores podem conseguir quebrar a solução quase imediatamente.

O problema realmente advém do fato que um único comprometimento da chave expõe e compromete todo o sistema de endpoints. Isso desvaloriza a implementação da TCB e desvaloriza o tempo e o dinheiro empregados nesta solução no endpoint e na arquitetura de IoT como um todo. Assim, este modelo é uma TCB perigosa para implementar, pois é, efetivamente, uma bomba-relógio.

6.1.1.2 Senhas Personalizadas

Independentemente de uma solução PSK ou assimétrica ser implementada, a personalização é imprescindível para que cada TCB funcione efetivamente. A personalização neutraliza a capacidade de um invasor de usar uma senha comprometida para subverter a segurança de todo o ecossistema de IoT. Se um invasor puder comprometer apenas um único endpoint de cada vez e precisar de acesso físico para fazê-lo, o processo de comprometer a tecnologia de IoT como um todo será extremamente lento, custoso e complexo. Esta é uma vantagem significativa para o negócio.

Devido à evolução dos padrões em comunicação celular ao longo das últimas décadas, as operadoras de rede aperfeiçoaram o modelo PSK para personalização de âncoras de confiança como o UICC. Como resultado, o UICC pode às vezes ser fornecido para servir como uma âncora de confiança para o endpoint IoT, ajudando a formar uma solução de segurança custo-eficiente para aplicações de IoT. Em um futuro próximo, quando o eUICC estiver disponível, esta opção pode ser utilizada mesmo quando o eUICC já estiver implementado comercialmente.

Hoje, a tecnologia de chave personalizada é a solução de segurança mais eficaz para uma âncora de confiança. Atualmente, as TCBs implementadas em IoT devem se basear em uma solução personalizada para cada TCB. Provedores de serviços IoT devem negociar com sua operadora de rede para determinar se o UICC ou SIM pode ser implementado como uma âncora de confiança na camada da aplicação.

6.1.2 Protocolos e tecnologias TCB

Junto à âncora de confiança, a TCB deve incorporar protocolos, políticas e bibliotecas de software para fornecer segurança ao produto ou serviço de IoT como um todo. Uma vantagem da utilização do padrão de chaves de confiança suportados pela rede celular é a capacidade de prescindir de softwares de provisionamento e personalização que já existem nas operadoras de rede. Tecnologias, protocolos e conjuntos como os seguintes apoiarão a capacidade da TCB de ajudar a autenticar o endpoint na rede:

- Aplicação oneM2M SM UICC como especificado no oneM2M TS-0003
- Arquitetura de bootstrapping genérica (GBA) 3GPP TS 33.220 (veja o anexo A)

O uso dessas tecnologias ajudará a acelerar a implementação do provisionamento e da personalização, pois as bibliotecas e os protocolos foram verificados por engenheiros experientes e analistas de segurança por muitos anos. No entanto, esses protocolos podem não permitir totalmente que a TCB valide a aplicação do endpoint ou garantir que este possa autenticar corretamente as mensagens ou autorizar ações. A TCB deve incorporar outros protocolos para realizar essas tarefas, como validação de firmware, validação de mensagem de atualização over-the-air, dentre outras.

Em um futuro próximo, tecnologias como o eUICC aumentarão os recursos a partir da perspectiva da aplicação, e o UICC proativo ativará a tecnologia de uso recíproco que pode ajudar a inicializar o próprio endpoint, enquanto gerencia a segurança da rede. Esse é um ganho importante, pois as operadoras de rede poderão gerenciar remotamente e com segurança o dispositivo eUICC em nome do provedor de serviços de IoT. Além disso, a funcionalidade de Gerenciamento Confidencial de Conteúdo de Cartão especificada na Plataforma Global de Especificação do Cartão [15] permite que vários agentes nos ecossistemas de serviços de IoT gerenciem suas próprias aplicações independentemente uns dos outros, se for permitido pela operadora de rede.

6.1.3 Risco

A escolha de não implementar uma TCB é uma falha crítica para toda a arquitetura de IoT. Sem uma TCB bem definida, a interação entre a senha confiável e a aplicação central será vagamente definida e pode ter brechas a serem utilizadas pelos invasores. A TCB garante que as comunicações entre a âncora de confiança, a aplicação principal e as redes pareadas sejam seguras, protegidas e atualizadas. Sem uma TCB, não há nenhum componente central para gerenciar o ciclo de vida da segurança do endpoint.

6.2 Utilize uma âncora de confiança

Para que um endpoint faça parte de um ecossistema, ele deve ser capaz de verificar a integridade de sua própria plataforma e deve ser capaz de autenticar sua identidade e de seus pares. Para fazer isso, os endpoints exigem uma âncora de confiança incorporada a um TBC.

Uma âncora de confiança é um elemento de hardware seguro, um chip físico segregado ou um núcleo seguro dentro de uma CPU, capaz de armazenar e processar com segurança senhas criptografadas. Um dispositivo UICC ou eUICC é um exemplo de uma tecnologia segura que pode ser usada como um elemento de confiança para armazenar senhas de autenticação.

Usar um elemento confiável envolve, efetivamente, armazenar, verificar, atualizar e processar dados. Os dados podem ser informações secretas ou públicas que devem ser verificadas criptograficamente. Em ambos os casos, a âncora de confiança deve ser capaz de determinar com segurança se as mensagens e identidades podem ser autenticadas e deve ser capaz de informar com segurança à TCB o resultado de todas as operações de autenticação ou criptografia. Isso permite que a aplicação e a TCB tomem decisões importantes que afetarão a segurança do endpoint como um todo. Por exemplo, uma âncora de confiança pode ajudar um endpoint a determinar se um ponto de rede está personificando um recurso crítico, como um servidor de implantação de atualizações. Se a âncora de confiança não puder validar um par de rede, a TCB e a aplicação no endpoint deverão optar por não interagir com esse par e devem alertar o usuário, quando possível, sobre o recurso de rede fraudulento.

Graças ao barateamento dos componentes e a um aumento acentuado na demanda, as âncoras de confiança estão se tornando cada vez mais disponíveis. Isso inclui não apenas a atual tecnologia de âncoras de confiança, mas bibliotecas e interfaces aprovadas para uso com esta tecnologia. Isso permite que a equipe de engenharia crie uma solução de âncora de confiança em muito pouco tempo e ajudará a garantir que a longevidade da tecnologia não seja enfraquecida por softwares customizados ou padrões mal implementados. Sempre que possível, os padrões devem ser usados para diminuir o potencial de falhas na segurança.

Outro desafio na implementação de uma âncora de confiança em endpoints de baixa complexidade é o tamanho do componente. Se uma âncora de confiança externa for utilizada, será necessário manter um perfil de componente mínimo. Conseguir esse perfil é difícil quando o fator “tamanho” incorpora tecnologias como UICC. No entanto, o padrão ETSI TS 102 671 resolve esse problema introduzindo um fator de tamanho muito pequeno de aproximadamente 6 milímetros por 5 milímetros. Esses aprimoramentos “MFF1” e “MFF2” para o formato UICC de smart card permitem acesso total às tecnologias com suporte a UICC e, ao mesmo tempo, garantem os requisitos físicos mínimos. Uma segurança extra é adicionada ao se utilizar o formulário fornecido localmente, que faz parte do dispositivo, dificultando a transferência de identidade de um dispositivo para outro pelos invasores.

Despesas inclusas no desenvolvimento e implantação de uma âncora de confiança podem incluir:

- O custo básico da tecnologia, seja embarcado na CPU ou em um chip separado
- O custo para integrar a tecnologia no circuito, se necessário
- O custo de engenharia ou integração do driver no sistema operacional e na TCB
- O custo de engenharia da aplicação para usar a âncora de confiança

- Manutenção da âncora de confiança, quando necessário
    - Manutenção de chaves de segurança, revogação de chaves e desativação de identidades
    - Manutenção da infraestrutura necessária para proteger e gerenciar as chaves e os metadados
- Monitoramento da identidade da âncora de confiança do lado do serviço
    - Implementação de lista negra de dispositivos, quando necessário
- Integração de serviços da operadora, quando disponíveis, para monitorar e gerenciar âncoras de confiança, como UICC

6.2.1 Risco

Os riscos de não utilizar uma âncora de confiança são muitos, mas todos derivam do mesmo problema básico: a possibilidade de um invasor de roubar chaves relevantes para todo o ecossistema de IoT. O resultado disso é que um invasor pode:

- Clonar identidades de endpoints
- Falsificar serviços de IoT
- Distribuir patches e atualizações não autorizadas
- Fazer alterações não autorizadas no software do endpoint

Essas falhas na segurança podem resultar em problemas dispendiosos para a empresa ao longo do tempo e permitir que não apenas os invasores, mas os concorrentes abusem da infraestrutura em seu benefício.

6.3 Use uma âncora de confiança resistente à violação

Algumas âncoras de segurança têm segurança física adicional para proteger contra certas classes de ataques, como FIBs, análise de canal lateral e glitching. Embora alguns ataques, como a utilização de um FIB, sejam quase impossíveis de serem evitados se o custo for considerado, a criação de âncoras de confiança pode usar tecnologias modernas para tornar os ataques mais trabalhosos. Quanto mais trabalhoso for um ataque, menor a chance de que este seja usado contra endpoints aleatórios. Em vez disso, os ataques serão focados em alvos que valham o esforço.

No futuro próximo, alguns fabricantes de âncoras de confiança planejam lançar variações de sua tecnologia que são aprovados pelos Padrões Governamentais de Processamento de Informações (FIPSI, da sigla em inglês) [10]g, EMVCo [11] e “Common Criteria”. Os engenheiros que desenvolvem novas tecnologias devem determinar se os projetos atuais darão suporte à transição para módulos compatíveis com esses padrões no futuro próximo.

Para mais informações, consulte a versão mais recente de cada padrão para avaliar o nível de recursos oferecido pelo fabricante. Observe que alguns níveis de segurança são intencionalmente quase inviáveis para dispositivos orientados ao usuário final devido ao custo e à complexidade das implementações.

6.3.1 Risco

O risco de não usar uma âncora de confiança resistente a adulterações é extremamente alto. Por exemplo, se uma âncora de confiança consistir simplesmente de chaves criptográficas embutidas na NVRAM, qualquer invasor com as ferramentas e habilidades para extrair essas chaves poderá subverter toda a infraestrutura. No entanto, se as chaves forem armazenadas em uma âncora de confiança eficiente e resistente a adulterações, o custo para extrair informações será alto, o que reduzirá bastante a possibilidade de serem extraídos, diminuindo, assim, a atratividade da âncora de confiança como potencial alvo de ataque.

É importante notar que, se a implementação de uma âncora de confiança for fraca, sua quebra pode resultar em comprometimento considerável. Qualquer comprometimento invalidará o investimento durante as etapas de engenharia, arquitetura, produção e atendimento. Isso pode resultar em uma perda financeira significativa. Portanto, garantir que a organização tenha desenvolvido a implementação correta é imperativo.

6.4 Utilize uma API para TCB

Uma vez que a raiz de confiança tenha sido estabelecida dentro da TCB, um protocolo deve ser usado para incorporar as capacidades da TCB e da raiz de confiança de forma eficaz. A API deve garantir que:

- Toda verificação de assinatura é executada pela TCB
- Nenhuma chave privada da TCB é exposta
- A troca de chaves pode ser realizada pela TCB em nome da aplicação
- A descriptografia deve ser executada pela TCB
- A criptografia deve ser executada pela TCB
- A assinatura de mensagens deve ser executada na TCB
- O preenchimento seguro de mensagens deve ser realizado na TCB
- Confidencialidade e Integridade entre a TCB e a aplicação

Esse conjunto de recursos ajudará a garantir que a TCB não irá expor ativos de segurança críticos a um ambiente de hardware ou aplicação inseguros. Isso pode ser feito usando uma especificação existente que aplica esses requisitos de maneira uniforme. Avalie:

- SIM Alliance Open Mobile API [12]
- Plataforma Global de Controle de Acesso ao Elemento Seguro [13]
- Plataforma Global de Ambiente de Execução Confiável (TEE, da sigla em inglês) Especificação API [14]
- Grupo de Computação Segura (TCG, da sigla em inglês)
- oneM2M TS-0003 [20]

Muitas âncoras de confiança vêm com bibliotecas de software que podem ser implementadas como uma TCB. Essas bibliotecas terão APIs que os engenheiros podem usar para interagir com a TCB. As bibliotecas fornecidas pela âncora de confiança devem ser priorizadas, quando disponíveis, pois provavelmente foram examinadas por especialistas no campo do desenvolvimento de âncoras de confiança. No entanto, a equipe de engenharia deve avaliar a lista de requisitos estabelecidos nesta recomendação e deve garantir que a biblioteca responda adequadamente a essas preocupações.

Além disso, as TCBs só devem ser acessíveis a partir de aplicações com privilégios de execução no endpoint. Uma interface TCB nunca deve ser acessível a partir de uma aplicação sem privilégios ou não confiável (de terceiros) em execução no endpoint. Todo o acesso deve ser intermediado por proxy e por meio de um serviço confiável que avalie as solicitações e, opcionalmente, alerte o usuário quando solicitações suspeitas ou centradas na privacidade forem feitas por aplicações não confiáveis.

O desafio na implementação deste protocolo é garantir que nenhuma mensagem possa ser adulterada entre o ponto de origem dos dados e a TCB e vice-versa. A eficácia será maior se uma área da EEPROM, que pode ser acessada a partir da aplicação, puder executar essas funções em nome da aplicação. Ao isolar o peso do código da API para a EEPROM interna e usar a RAM interna para processar as mensagens, menos dados críticos serão expostos aos barramentos externos.

6.4.1 Risco

Se uma interface de protocolo da aplicação não estiver bem definida, o uso de uma TCB pode ter resultados indesejados ou efeitos colaterais. Ao definir o protocolo antecipadamente e examiná-lo quanto a problemas de lógica e segurança, a equipe de engenharia pode identificar com mais rapidez e eficácia as falhas que podem resultar em problemas de segurança posteriormente. Assim, a definição do protocolo deve incorporar a avaliação das APIs existentes que incorporam as necessidades do provedor de serviços de IoT. Sempre que uma tecnologia existente e bem estabelecida puder ser identificada, esse caminho deverá ser favorecido em relação a uma solução customizada.

6.5 Definindo uma raiz organizacional da confiança

Uma raiz organizacional da confiança é um conjunto de políticas e procedimentos de criptografia que controlam identidades, aplicações e comunicações que podem e devem ser protegidos criptograficamente. Criptografia forte deve ser usada, seja na forma de chaves simétricas únicas, certificados ou chaves públicas. Isso depende do modelo disponível para uso na TCB, dos recursos da âncora de confiança e do que fizer sentido para a equipe de engenharia.

Uma chave de raiz privada, simétrica ou assimétrica, deve ser usada para assinar digitalmente outras chaves usadas na hierarquia. Por exemplo, se nossa empresa de exemplo, a IoT Company LTDA, quiser criar uma raiz organizacional da confiança, ela geraria uma chave raiz em uma máquina confiável. Essa chave representará a raiz organizacional. Eles então gerariam novas chaves representando cada sub organização que deveria ter hierarquias de segurança independentes. Exemplos podem ser:

- Chave de assinatura de código
- Chave de comunicação do servidor
- Chave de comunicações ponto a ponto
- Chave de identidade do endpoint
- Chave de revogação mestre

Cada uma dessas chaves deve ser assinada pela chave de raiz organizacional. Todas essas chaves, suas assinaturas correspondentes e a chave de raiz devem ser armazenadas na senha confiável usada pela TCB. Em seguida, sempre que a aplicação vinculada a uma

determinada chave for usada, a aplicação poderá usar as chaves específicas para validar as mensagens enviadas pelos canais de comunicação.

Esse modelo ajuda a garantir que todas as mensagens sejam protegidas pela hierarquia criptográfica. Ao separar as tarefas entre tipos de chaves específicas, as chaves comprometidas podem ser revogadas pelo mesmo processo de comunicação.

Alguns protocolos existentes que auxiliam na implantação desse método são:

- Camada de Segurança de Transporte (TLS, da sigla em inglês); a especificação válida mais recente
- Secure Shell (SSH2)
- Protocolo de Status de Certificado Online (OCSP) IETF RFC 2560
- Arquitetura Genérica de Inicialização (GBA, da sigla em inglês) (Ver Anexo A) 3GPP TS 33.220

Dificuldades surgem quando os serviços que precisam das chaves criptográficas devem ser implantados. Em vez de colocar um ativo de segurança crítica, como a chave de comunicação do servidor, em um servidor web acessível pela internet, um certificado ou par de chaves separado deve ser gerado especificamente para essa camada de servidor. Em seguida, esse certificado pode ser assinado pela chave de comunicação do servidor. Dessa forma, qualquer endpoint pode verificar se o serviço foi autenticado pela raiz de confiança, mas a chave organizacional crítica não será exposta aos invasores.

Se uma chave for comprometida, seu uso pode ser revogado utilizando a chave mestra para autenticar a revogação.

É evidente que todas as chaves do núcleo da raiz de confiança organizacional são críticas para a segurança da infraestrutura. Essas chaves devem ser fortemente protegidas e usadas apenas por membros confiáveis da equipe principal. A utilização de um Módulo de Segurança de Hardware (HSM, da sigla em inglês) aprovado para armazenar, acessar e usar as chaves é altamente recomendada.

Embora um HSM possa muitas vezes representar um custo significativo no início da implantação de uma tecnologia, o retorno financeiro no longo prazo é positivo. Em vez de incorrer em uma despesa futura significativa com análise forense e engenharia para diagnosticar e combater um risco específico que poderia ter sido resolvido por uma TCB e um HSM, esse investimento inicial relativamente pequeno vale a pena.

6.5.1 Risco

O risco de não usar uma raiz organizacional de confiança é que qualquer comprometimento com uma única chave pode resultar em comprometimento de todo o ecossistema. Ao separar a organização em uma hierarquia e implantar chaves separadas, as chaves podem ser alternadas em intervalos regulares e de acordo com a prioridade da aplicação ou sub organização à qual a chave se refere. Isso cria uma separação de tarefas entre as vertentes da organização e diminui a possibilidade de uma chave comprometida subverter a segurança de toda a infraestrutura.

6.6 Personalize cada dispositivo endpoint antes da execução

Os endpoints devem ser ativados com identidades exclusivas de criptografia para garantir que invasores, concorrentes e amadores não sejam capazes de personificar outros usuários ou dispositivos em ambiente de produção. Para adequadamente alcançar esse objetivo, o processo de personalização deve ser executado na fabricação. Isso pode ser feito pelo fabricante da solução TCB específica ou durante o processo de montagem da placa de circuito impresso (PCB/A).

Para resolver o processo de personalização, faça o seguinte:

- Gere uma chave criptográfica única
- Assine a chave usando a Chave de Assinatura de Endpoint organizacional (ou um derivado desta)
- Armazene a chave na âncora de confiança da TCB
- Gere (ou use) um único identificador interno para cada endpoint específico
- Armazene o identificador único na âncora de confiança da TCB
- Salve a assinatura única, a chave e a assinatura no sistema de autenticação back-end do serviço de IoT

Observe que a personalização da plataforma do endpoint é separada da personalização da identidade da rede. A utilização de um UICC para autenticação de rede é benéfica por vários motivos e, quando possível, o UICC pode ser usado como uma âncora de confiança. No entanto, se a âncora de confiança da rede puder ser usada apenas para autenticação da rede, a personalização da âncora de confiança da aplicação deverá ser executada separadamente. A exclusividade criptográfica da âncora de confiança da aplicação é necessária para garantir que a plataforma da aplicação seja verificada antes da execução da aplicação do endpoint.

Usando um contrato apropriado com uma operadora de rede ou terceiro, os UICCs podem às vezes ser fornecidos antes da entrega para servir como uma âncora de confiança centrada na aplicação. Em um futuro próximo, os desenvolvedores de endpoint deverão avaliar se a tecnologia eUICC é adequada para uso em produtos e serviços de IoT. Essas tecnologias permitirão o provisionamento em campo de chaves criptográficas de maneira semelhante a uma âncora de confiança centrada na aplicação. Como a indústria móvel é líder no processo de personalização e provisionamento, pode haver uma vantagem significativa em usar um eUICC como uma âncora de confiança.

Além disso, essas tecnologias irão incorporar recursos de gerenciamento remoto e Canal Seguro para comunicação protegida entre a aplicação e a âncora de confiança eUICC. Esses recursos fornecerão customização em campo, o que reduzirá o custo de customização e provisionamento para cada endpoint.

O anexo B contém um breve tutorial sobre o uso de cartões UICC em um ecossistema de serviços de IoT.

O desafio vem com o gerenciamento das identidades do endpoint e o processo de assinatura. Cada identidade deve ser catalogada junto aos identificadores exclusivos que correspondam à identidade em um sistema que não pode ser adulterado. Embora o

processo geralmente seja executado na instalação de PCB/A, uma conexão desse recurso com a empresa deve ser configurada para trafegar com segurança os dados de identidade.

A implementação dessa solução pode ser direta com alguns fabricantes que estão mais familiarizados com a personalização criptográfica. Outras instalações de fabricação podem não ter um processo para resolver isso. A indústria móvel tem tido sucesso nessa tarefa devido à sua capacidade de controlar a fabricação e o cumprimento de tecnologias embarcadas como o UICC. Embora a indústria móvel seja líder nesse processo há algum tempo, o processo de personalização e provisionamento de aplicações de endpoints IoT ainda está engatinhando.

Esteja preparado para averiguar se a identidade do endpoint deve, ou poderia ser gerenciada por um gateway ou uplink. A avaliação da arquitetura do produto ou serviço de IoT ajudará a determinar se esse atributo do gerenciamento de identidades afetará o processo de personalização. Embora a proteção possa ser distribuída para gateways, a organização deve determinar se ela pode ser adequadamente delegada sem diminuir a segurança do sistema de comunicações e autenticação como um todo.

Os custos envolvidos na personalização geralmente incluem, mas não estão limitados a:

- Implementação do processo de personalização no fabricante do chip
- Coordenação ou entrega de dados personalizados únicos no fabricante e no provedor de serviços de IoT
- Implementação e gestão de identidades personalizadas

6.6.1 Risco

Se a organização optar por não implementar a personalização do endpoint, corre o risco de não conseguir diferenciar um do outro. Se todas as chaves estiverem em conformidade nos sistemas endpoint, não importa se os números de série são exclusivos. A razão para isso é que, se as chaves forem extraídas de um único endpoint, o invasor seria capaz de personificar qualquer um deles.

A personalização combate isso ao forçar o invasor a extrair os segredos criptográficos de cada endpoint que eles desejam clonar ou personificar. Como o custo desse processo pode ser muito alto, a customização usando uma senha confiável é o método mais robusto para combater a clonagem e a personificação.

6.7 Plataforma mínima para execução viável (Roll-Back de aplicação)

Uma Plataforma Mínima de Execução Viável (MVeP, da sigla em inglês) é a quantidade mínima de trabalho que deve ser executada para criar um ambiente de execução confiável ao se comunicar com a âncora de confiança. Normalmente, isso significa:

- Configurar o relógio interno ou oscilador
- Configurar periféricos principais (memória, armazenamento)
- Habilitar diversas bridges de hardware ou dispositivos periféricos
- Autenticar o próximo trecho de código a ser executado pela CPU
- Executar a próxima etapa do código
- Gerenciar o roll-back da aplicação de imagem

Com o modelo de MVeP definido, o gerenciador de boot mínimo pode usar a âncora de confiança para verificar outro gerenciador mais robusto, ou pode executar o restante da inicialização depois de verificar aplicações externas. Isso permite que um ambiente consistente seja definido com esforço mínimo para autenticar as cadeias subsequentes de código que definirão a plataforma de aplicações.

Outro benefício é que, com o uso do modelo MVeP, até mesmo processadores com uma quantidade mínima de NVRAM interna ou EEPROM podem inicializar uma arquitetura confiável usando uma âncora de confiança interna ou externa.

Por fim, uma MVeP é importante para reverter a versões estáveis de uma plataforma específica. Se uma MVeP puder ser definida com a funcionalidade mínima necessária para verificar a integridade das imagens de firmware da aplicação e configurar o ambiente de execução, sua funcionalidade poderá ser separada da funcionalidade principal da aplicação. Portanto, se uma atualização de firmware falhar por algum motivo, a MVeP ainda poderá ser usada para se reconectar à rede back-end e fazer o download de outra imagem de firmware (a mesma imagem ou uma imagem mais antiga). Isso também permite que os endpoints com chips NVRAM danificados ainda se comuniquem com os serviços de back-end e enviem informações para diagnóstico.

6.7.1 Risco

Embora possa parecer benigna, a definição de uma MVeP garante que a arquitetura do endpoint como um todo verifique criptograficamente cada etapa do processo de inicialização. Essa etapa é essencial para garantir que um endpoint possa se autenticar na rede e em seus pares. Se a MVeP estiver mal projetada, isso poderá resultar em falhas de segurança no processo de inicialização que podem ser exploradas por invasores, destruindo o modelo da segurança.

6.8 Provisionamento único para cada endpoint

Embora a customização garanta que cada dispositivo seja único depois de fabricado, o provisionamento garante que um dispositivo único seja ativado, atualizado e associado a uma identidade específica do cliente. O processo de provisionamento ajuda a separar os dispositivos que foram fabricados a partir de dispositivos que foram comprados e/ou implementados em um ambiente de IoT. Isso ajuda o provedor de serviços de IoT:

- Distinguir entre dispositivos ativos e desativados
- Associar endpoints com redes ou outros recursos conectados com um cliente particular
- Customizar um endpoint segundo as necessidades do cliente
- Determinar mais facilmente se um determinado cliente ou endpoint foi comprometido

O processo de provisionamento não ocorre durante a fabricação, mas depende do processo de customização implantado na fabricação. Normalmente, o provisionamento ocorre em campo, com base no cliente que inicializa o processo de ativação. No entanto, para que o processo seja seguro, o provisionamento depende dos tokens de segurança exclusivos definidos durante o processo de customização para garantir que o endpoint único esteja

vinculado a um cliente único. Dessa forma, um hacker não pode registrar arbitrariamente, ou cancelar o registro de endpoints simplesmente adivinhando detalhes deles. Em vez disso, exigiriam que cada token criptográfico exclusivo fosse gerado e configurado durante o processo de customização, o que é computacionalmente inviável.

Desse modo, o provedor de serviços de IoT pode garantir matematicamente que é improvável que os invasores possam falsificar ou registrar arbitrariamente endpoints à vontade. Isso leva a um ambiente de IoT mais seguro e estável, em que o relacionamento entre clientes e dispositivos pode ser mais confiável.

6.8.1 Risco

Não implementar o processo de provisionamento pode resultar em uma dessincronização entre a organização e seus endpoints. Será mais difícil para a organização rastrear endpoints e estabelecer quais dispositivos foram habilitados para uso no ecossistema ou desativados. Além disso, pode ser difícil estabelecer quais dispositivos estão associados a clientes específicos, o que aumentará a dificuldade de rastrear um dispositivo problemático ou potencialmente comprometido em campo.

6.9 Gerenciamento de senhas endpoint

Dispositivos que apresentam interfaces de usuário devem ser capazes de gerenciar senhas com eficiência. Isso requer várias coisas:

- Mitigação de ataques de força bruta
- Desabilitação do uso de senhas padrão ou armazenadas
- Aplicação de melhores práticas de senhas
- Impossibilitar a exibição de credenciais do usuário em interfaces de login
- Impor limites e atrasos incrementais para tentativas inválidas de senha

Os usuários precisarão ser protegidos contra o ataque mais simples possível, como outro usuário tentando adivinhar sua senha. Isso pode ser aliviado simplesmente negando o potencial de um ataque de força bruta. Isso pode ser feito aumentando o limite de tempo entre as tentativas de quebra de senha. A cada tentativa falha de login, deve haver um atraso maior antes que a próxima senha possa ser inserida. Um limite deve ser implementado de tal forma que não mais que N tentativas possam ser tentadas de uma só vez. Caso contrário, um período de bloqueio razoável deve ser imposto. O usuário deve ser alertado sobre a tentativa de força bruta quando as credenciais reais forem inseridas.

Senhas fracas ou óbvias nunca devem ser usadas em sistemas IoT. Nunca deve haver uma "senha de back door" administrativa para entrar em um sistema. Nunca deve haver uma conta privilegiada com credenciais padrão. Isso é essencial para proteger os dispositivos do usuário contra intrusões não autorizadas por usuários que correm aleatoriamente na internet em busca de segurança fraca.

As senhas devem atender aos requisitos mínimos de qualidade, representativas das melhores práticas atuais de segurança da informação. Isso garante que forçar uma senha com força bruta será difícil, e ajuda o usuário a se proteger contra falsificações. Considere revisar as diretrizes OWASP ou SANS para segurança de senha para garantir que a aplicação esteja em conformidade com as práticas recomendadas mais recentes.

As senhas nunca devem ser exibidas na tela de um usuário. Sempre esconda a senha com asteriscos ou outro sigilo equivalente.

Além disso, todas as interfaces que aceitam senhas devem utilizar a tecnologia de mitigação de força bruta. Também é importante que a tecnologia que valida a senha obedeça essa obrigação. Por exemplo, o JavaScript embarcado em uma página Web renderizada em um navegador não deve implementar a mitigação de força bruta. Qualquer invasor experiente da Web pode contornar esses controles interagindo com o servidor de autenticação back-end pela internet. A tecnologia de mitigação deve ser implementada do lado do servidor neste modelo. Em aplicações para dispositivos móveis, em que um PIN ou senha local é incorporado na região de armazenamento seguro da aplicação, o dispositivo móvel deve atenuar os ataques de força bruta na interface.

Além disso, após cada tentativa de senha inválida, o sistema de atenuação deve aumentar o atraso necessário entre as tentativas permitidas. Também deve haver um limite máximo para tentativas inválidas de senha. Após esse limite ser atingido, o usuário deve ser bloqueado, restando apenas a autenticação de dois fatores ou outro modelo mais eficiente.
Dificuldade

Esse processo é extremamente simples de implementar e exige muito pouco esforço da equipe de engenharia.

6.9.1 Risco

O risco de não implementar esta recomendação é:

- A capacidade de subverter dispositivos por meio de roubo de senha com força bruta
- Ataques de internet "Drive by" podem subverter a segurança dos sistemas de IoT, simplesmente usando senhas codificadas
- Os usuários podem ficar comprometidos por meio do ataque “espiar sobre o ombro” se a interface do usuário exibir a senha real que está sendo inserida no sistema

6.10 Use um gerador certificado de números aleatórios

Determine se a sua TCB é capaz de gerar números verdadeiramente aleatórios. Isso é importante, pois, sem isso, o processo de verificação criptográfica pode ser prejudicado, tornando os dados criptografados mais previsíveis e enfraquecendo a integridade dos dados.

Isso também é extremamente importante para a geração de chaves criptográficas únicas. Dado um conjunto de condições ambientais, um invasor não deve ser capaz de influenciar o ambiente para fazer com que uma TCB gere números previsíveis durante a geração de chave, assinatura ou preenchimento de mensagem criptográfica.

Esse processo é tão simples quanto identificar se a TCB é capaz de geração de números aleatórios aprovados pelo FIPS [10], EMVCo [11] ou Common Criteria.

6.10.1 Risco

Utilizar criptografia sem um gerador de números aleatórios é perigoso por vários motivos. Embora os motivos sejam muitos para listar aqui, há alguns pontos fracos importantes que a serem observados:

- A geração de chaves criptográficas pode ser comprometida, causando a geração de chaves fracas ou previsíveis
- Senhas únicas, blocos ou chaves podem ser fracos ou previsíveis
- O preenchimento de mensagens usado para negar o potencial de repetição de mensagens pode estar comprometido

Essas questões podem resultar em falhas significativas na integridade da segurança criptográfica de todo o ecossistema de IoT. Esse risco não afeta apenas o dispositivo endpoint, afeta toda a rede.

6.11 Assinar criptograficamente Imagens da aplicação

Todas as aplicações armazenadas fora da EEPROM principal de uma CPU devem ser autenticadas criptograficamente. Para fazer isso, basta seguir o procedimento:

- Identifique os metadados que representam a versão da imagem da aplicação
- Gere um hash criptográfico da imagem da aplicação, incluindo os metadados
- Valide se os metadados da aplicação correspondem aos metadados internos
- Valide se o valor de hash corresponde ao valor interno da senha confiável
- Valide criptograficamente a assinatura com a chave da aplicação
- Valide criptograficamente se a chave de assinatura de aplicação foi assinada pela senha da raiz organizacional

Esse processo é ordenado a executar primeiro as atividades mais simples e as operações com menor probabilidade de falhar por último. Dessa forma, a menor quantidade de trabalho é realizada para observar os riscos mais prováveis.

Além disso, é um processo excepcionalmente fácil de implementar, especialmente quando a TCB é capaz de dar suporte ao processamento em nome da aplicação. O desafio real é mais sutil: é a que está executando a operação.

Uma aplicação que não foi verificada criptograficamente não pode realizar a operação, pois não tem como saber se seu próprio código foi subvertido por um invasor. A alteração de código na NVRAM é uma maneira comum de os invasores manipularem sistemas incorporados se o sistema embarcado não verificar a aplicação.

Uma aplicação na EEPROM interna deve, em vez disso, executar este procedimento primeiro, em qualquer imagem da aplicação no armazenamento permanente externo. Em seguida, essa aplicação pode executar a operação por conta própria ou solicitar uma aplicação codificado na EEPROM interna para executar esses tipos de testes em seu nome.

6.11.1 Risco

Se a imagem da aplicação armazenada no firmware do endpoint (NVRAM) não for assinada criptograficamente, o sistema não poderá diferenciar entre código autorizado e código inserido por um invasor. Isso pode permitir que não apenas um hacker adultere o código executável para manipular um endpoint comprometido fisicamente, mas pode permitir que um competidor instale seu próprio software em um endpoint.

6.12 Administração remota do endpoint

Embora nem todos os endpoints exijam administração remota, os que necessitam devem ser pensados de uma maneira que garanta que terceiros não possam usar suas credenciais administrativas para comprometer alguns (ou todos) os endpoints em utilização. A solução apropriada dependerá dos recursos do endpoint. No entanto, as seguintes diretrizes devem ser usadas:

- Não salve componentes criptográficos privados em um armazenamento inseguro em endpoints, como chaves privadas SSH, chaves privadas TLS ou senhas
- Sempre que possível, gere tokens administrativos (chaves criptográficas ou senhas) para cada endpoint
- Onde senhas são usadas, imponha o uso das que estejam em conformidade com as melhores práticas relacionadas à complexidade e extensão.
- Sempre que possível, utilize autenticação de dois fatores para administradores
- Assegure-se de que o usuário seja informado quando um administrador acessar remotamente o endpoint
- Considere restringir o acesso administrativo a uma rede virtual privada (VPN)
- Não introduza recursos administrativos remotos em uma aplicação ou API acessível publicamente, use um canal de comunicação separado e distinto
- Reforce a confidencialidade e a integridade no canal de comunicações administrativas
- Diminua o potencial de repetição dos comandos administrativos, garantindo que o protocolo de comunicações tenha a entropia adequada usando um protocolo de comunicação que seja padrão na indústria

6.12.1 Risco

A falha em definir, implementar e aplicar uma política sobre administração remota pode resultar no comprometimento remoto dos endpoints. Se não houver um modelo de segurança rígido para o acesso de superusuário aos dispositivos endpoint, os invasores poderão fazer engenharia reversa da tecnologia ou extrair chaves de segurança dos endpoints que resultarão em acesso a todos os endpoints do ecossistema. O acesso administrativo é, em geral, uma das primeiras tecnologias usadas por invasores em sistemas embarcados, pois muitas vezes são mal configurados ou tecnologicamente fracos.

6.13 Registro de logs e diagnóstico

Para avaliar problemas com endpoints, o provedor de serviços de IoT deve avaliar constantemente seu comportamento e determinar se este está funcionando dentro do conjunto de comportamentos certificados. Para conseguir isso, três estratégias devem ser usadas:

- Detecção de anomalias
- Registro de logs no endpoint

- Diagnóstico do endpoint

Um endpoint deve registrar seu próprio comportamento e carregar de forma intermitente esse registro para serviços back-end para processamento. Esse log deve ser composto de atividades normais, como logs ao kernel, logs em aplicações e outros metadados.

As informações de diagnóstico também devem ser observadas em intervalos regulares e entregues ao serviço back-end, juntas ou separadas dos registros normais. As mensagens de diagnóstico devem incluir o máximo possível de dados sobre o ambiente do endpoint, incluindo temperatura, duração da bateria, uso de memória, tempo de execução, listas de processos (quando aplicável) e muito mais. Essas informações ajudarão a identificar quando - e quais - serviços estão relacionados a um evento problemático ou anômalo.

A detecção de anomalias em rede deve ajudar a detectar um problema que não pode ser revelado por meio de análise de logs ou diagnóstico. Também ajudará a classificar os pontos que podem ser observados nos logs ou diagnósticos, ou atribuir as questões a um componente específico que pode estar reagindo mal em campo. Por exemplo, um módulo celular que continua se reconectando à rede ou um sensor que gera dados incorretos.

Juntas, essas informações não apenas ajudarão a identificar se uma falha na tecnologia é observada no campo. Também ajudarão a identificar se o comportamento anômalo é indicativo de um evento de segurança.

6.13.1 Risco

Não implementar registro de logs e diagnósticos pode fazer com que a organização perca informações críticas. Essas informações podem não apenas afetar a segurança do ecossistema, mas podem ajudar a diagnosticar falhas críticas de engenharia do produto.

6.14 Reforce a proteção à memória

Os sistemas embarcados geralmente são projetados com microcontroladores que não são capazes de tecnologia robusta, como unidades de gerenciamento de memória (MMU) e unidades de proteção de memória (MPU). No entanto, essas tecnologias devem ser usadas em qualquer plataforma que pretenda:

- Executar aplicações sem privilégios
- Executar aplicações não certificadas (de terceiros)
- Executar um emulador ou máquina virtual em um processo sem privilégios

Qualquer ambiente que exija a execução de uma aplicação sem privilégios deve ser capaz de se proteger de aplicações nocivas ou comprometidas. Isso garante que estes não possam acessar áreas de memória que controlam recursos privilegiados, como a TCB, o driver da âncora de confiança ou os registros de logs de hardware periférico.

O desafio nessa área é frequentemente migrar de uma plataforma de microcontrolador de oito bits para uma plataforma mais robusta, como um microcontrolador de 32 bits ou uma arquitetura de processamento completo. No entanto, existem muitos sistemas operacionais disponíveis gratuitamente ou com uma taxa de licença nominal para sistemas embarcados que implementam corretamente a proteção de memória com um MPU ou MMU.

6.14.1 Risco

Se essas tecnologias não forem usadas, aplicações mal intencionadas ou comprometidas não serão impedidas de alterar os recursos principais, como drivers, logs de periféricos ou até serviços privilegiados, como o kernel e outras aplicações. A falta de proteção à memória permite que qualquer aplicação tenha acesso total à faixa completa de memória presente no microcontrolador ou processador. Aplicações não privilegiadas *devem* ser impedidas de usar esses recursos.

6.15 Inicialização fora da EEPROM interna

A maioria dos códigos de inicialização está embutida na EEPROM (Memória somente de leitura apagável eletricamente), interna à CPU. Isso nem sempre é o caso, no entanto. Determine se sua CPU carrega seu inicializador de uma fonte externa. Se a CPU não possui EEPROM, o que lhe permite verificar o código do inicializador, ele pode ser manipulado por um invasor local para configurar a CPU de maneira benéfica para ele.

Dependendo do nível de proteção fornecido ao chip ou à região da memória que hospeda o inicializador, um invasor pode usar um barramento local (como SPI - Serial Peripheral Interface) ou uma API remota (como firmware Over-the-air) para manipular o código embarcado. Isso permitirá a um invasor subverter a plataforma de computação colocando código personalizado no ponto de execução mais confiável, o primeiro estágio do código executável. Outro ataque poderia ser um hacker simplesmente trocar um chip inicializador por seu próprio chip contendo instruções personalizadas retirando e solda e soldando o novo chip. Sem uma maneira de verificar a integridade do código externo, o usuário não conseguirá distinguir entre software aprovado e não aprovado.

Para personalizar um inicializador, um invasor precisaria desenvolver ou terceirizar o desenvolvimento do gerenciador de boot. Dependendo dos recursos disponíveis e do processador de destino, a dificuldade dessa ação pode variar muito, do extremamente fácil ao extremamente difícil.

Considere o uso de uma CPU ou MCU/MPU com uma EEPROM interna ou NVRAM com capacidade de bloqueio para armazenar o inicializador. Isso ajudará a garantir que a plataforma possa ao menos verificar o primeiro executável carregado e executado pela arquitetura, resultando em um dispositivo mais confiável.

6.15.1 Risco

Não avaliar a cadeia de segurança e impor uma verificação de integridade para o inicializador da CPU pode resultar em um comprometimento total do sistema. Essa etapa é essencial para proteger o dispositivo endpoint IoT e, portanto, o ecossistema.

6.16 Bloqueando áreas críticas da memória

Aplicações críticas armazenadas em regiões executáveis de memória, como inicializadores de primeiro estágio ou TBCs, devem ser armazenadas como read-only. Isso garante que o dispositivo possa ser inicializado em uma configuração válida sem intervenção de um invasor. Sem essa garantia, o código executável carregado após o primeiro estágio de execução não poderá confiar que foi inicializado em uma configuração ou estado válido.

Embora seja verdade que os invasores ainda possam subverter o sistema, substituindo essas seções críticas de memória por seu próprio código, é necessário que eles criem sua própria versão personalizada do software, o que pode ser um processo complexo e desafiador. Isso aumenta enormemente o esforço total do ataque e a habilidade necessária para obter sucesso. Além disso, se customização e provisionamento forem usados, essas etapas obrigarão o invasor a recriar o processo para cada endpoint, personalizando sua solução para as características criptográficas exclusivas do sistema local. Isso torna o ataque de modo geral excepcionalmente caro e diminui sua viabilidade.

Para prevenir esse risco, simplesmente identifique se a tecnologia que armazena áreas críticas de memória é capaz de ser bloqueada. Alternativamente, comece com uma tecnologia EEPROM com trava.

Certifique-se de que, se um bloqueio for usado, o bloqueio não será definido no software. Bloqueios definidos por software são ativados somente após ele ter executado a respectiva funcionalidade para ativar o bloqueio. Haverá uma brecha de poucos milissegundos em que um hacker pode abusar do estado desbloqueado para seu ganho. Assim, bloqueios de hardware, como fusíveis ou bits de bloqueio, devem ser empregados sempre que possível.

6.16.1 Risco

Sem um bloqueio ou estado de read-only, áreas críticas de memória podem ser facilmente alteradas por um invasor. Isso pode dar a eles privilégios suficientes para comprometer toda a plataforma endpoint sem ações adicionais, subvertendo todos os controles de segurança subsequentes usados no sistema.

6.17 Iniciadores inseguros

O trabalho de um gerenciador de boot não é apenas configurar a CPU para execução de uma aplicação primário, mas também fazer o boot e transferir o controle executivo para a aplicação. Para conseguir isso, o inicializador normalmente encontra e carrega a aplicação principal na memória principal da CPU. O problema surge quando inicializadores padrão são usados em certos tipos de sistemas.

Muitos gerenciadores de inicialização usados por fornecedores de microcontroladores, por exemplo, permitem que um firmware externo seja carregado na memória da CPU para execução, ou permite atualizações de firmware por meio de interfaces seriais. Outros gerenciadores de inicialização podem apresentar ao usuário locais que possuam cópias de aplicações, permitindo, assim, que um usuário execute qualquer aplicação escolhida.

Embora essa funcionalidade seja esperada em um ambiente como um desktop, laptop ou até mesmo servidor, isso é inaceitável em sistemas embarcados. Isso porque, se um inicializador carregar e executar uma aplicação não verificada e não confiável, não há garantia quanto à confiabilidade ou segurança da aplicação executado, deixando o dispositivo embarcado em estado duvidoso.

Portanto, para prevenir esse problema:

- O inicializador deve ser capaz de verificar criptograficamente a cópia da aplicação a ser executada

- O inicializador padrão não deve ser usado para permitir imagens alternativas ou flash de firmware
- O inicializador não deve permitir imagens de aplicações carregadas de locais de armazenamento arbitrários
- O inicializador não deve ter cópias locais armazenadas em locais de armazenamento arbitrários

Além disso, o projeto de um inicializador deve estar sujeito a auditoria por um analista de segurança externo. Comprometer um inicializador por meio da manipulação de erros no software pode levar à execução de código personalizado ou a um desvio de verificação de integridade. Isso pode levar ao jailbreak, que pode não ser benéfico para o negócio. Assegure-se de que todos os inicializadores usados no sistema sejam totalmente auditados quanto a falhas de programação de software que possam levar a riscos de segurança.

6.17.1 Risco

Um gerenciador de boot inseguro pode ser tão prejudicial quanto um processo de inicialização mal desenhado. Proteger o inicializador é um passo crítico para garantir a integridade do endpoint em IoT.

6.18 Perfeita antecipação de sigilo

O Perfect Forward Secrecy (PFS) lida com a divulgação de chaves criptográficas trocadas durante a configuração de comunicações entre dois endpoints. Geralmente, estes terão certificados assimétricos usados para autenticar suas identidades. Após a conclusão da fase de autenticação, uma chave simétrica é gerada e mutuamente acordada usando criptografia assimétrica para proteger a negociação de chave. Uma vez que esta chave é gerada e acordada, ela será usada para proteger o resto da sessão entre as duas entidades. Isso é usado para diminuir o investimento computacional envolvido na criptografia assimétrica. A criptografia simétrica é computacionalmente mais barata, o que significa tanto mais rápida quanto menos intensiva em tecnologias embarcadas ou de baixo consumo de energia.

No entanto, há um porém. Este modelo de combinação de chave comum presume que as chaves assimétricas são sempre mantidas em segredo. Isso pode não ser o caso. No futuro, uma entidade suficientemente financiada pode ser capaz de computar a chave privada para qualquer dada chave assimétrica pública. Se o invasor salvar todas as sessões de comunicação entre uma entidade de destino e seus pares, esta será capaz de descriptografar todas as mensagens de comunicação do passado, gerando a chave privada em algum momento no futuro.

Além disso, a chave criptográfica de um servidor pode ser comprometida por terceiros anônimos ou até mesmo por pessoas de dentro da empresa. Se isso ocorrer, qualquer um que tenha armazenado mensagens de comunicação protegidas pela chave assimétrica roubada poderá então descriptografar essas mensagens.

Uma solução para esse problema é gerar um par de chaves assimétricas efêmeras durante o processo de negociação de chaves. Somente a chave pública para este par de chaves temporárias é passada para cada lado do link de comunicação, que pode ser usada para trafegar uma chave simétrica.

Essa chave efêmera deve ser gerada com entropia suficiente e um tamanho de chave grande o suficiente para negar o potencial de um ataque de exaustão computacional dentro de um período de tempo razoável. Isso garantirá que o processo de negociação de chaves seja sustentável e menos provável de estar sujeito a ataques no futuro.

Além disso, essa metodologia garante que os pares usem sua chave assimétrica permanente apenas para autenticação, não para confidencialidade e integridade. Se esta chave assimétrica for roubada ou exposta ao público, afetará apenas o processo de autenticação, não a confidencialidade e integridade do canal de comunicação.

Para tornar esse processo ainda mais resistente ao ataque, a chave assimétrica usada para autenticação deve estar sujeita a um processo de revogação segura que garanta que um endpoint possa verificar se uma chave foi comprometida. O endpoint não deve mais confiar nessa chave para autenticação se tiver sido notificado que esse comprometimento ocorreu.

6.18.1 Risco

A não implementação do PFS pode expor todas as comunicações de rede a um invasor se este obtiver acesso a uma chave privada usada para proteger o canal de comunicação. A qualquer momento no futuro, se o invasor capturar a chave privada, todas as comunicações capturadas por ele no passado serão decifradas. Isso implicará sérias consequências.

6.19 Segurança de comunicações do endpoint

Embora esse item tenha sido coberto por várias outras recomendações e riscos ao longo deste guia, é importante observar de forma sucinta que a segurança das comunicações do endpoint é a maior ameaça para os endpoints IoT. A capacidade de um invasor manipular o canal de comunicação é a maneira mais simples de comprometer um endpoint.

Como resultado, desenvolvedores de endpoints devem implementar a segurança de comunicações das seguintes perspectivas:

- Autenticação de redes pareadas
- Confidencialidade de dados
- Integridade das mensagens

Embora as mensagens de texto não criptografado possam ser enviadas e recebidas para interoperar com os endpoints projetados por outras organizações, os dados transmitidos por qualquer canal que incorpore comandos, dados de privacidade do usuário ou mensagens críticas do sistema devem ser protegidos. O primeiro passo é autenticar o dispositivo pareado para garantir que ele é o que afirma ser. Isso é especialmente importante se o par representar um serviço do sistema.

Em seguida, a confidencialidade dos dados é necessária para garantir que terceiros não possam ler dados críticos transmitidos por um canal de comunicação.

Finalmente, a integridade da mensagem é necessária para garantir que mensagens secretas não tenham sido adulteradas por um invasor.

Esses três atributos combinados resultarão em um modelo de comunicação que pode ter uma vida útil de vários anos com poucas alterações de engenharia.

Esse processo é simplificado com o uso de protocolos de segurança existentes e bem analisados, tais como, mas não limitados a:

- O padrão aprovado de TLS mais recente
- O padrão aprovado de DTLS mais recente
- SSH2 para autenticação e troca de chaves
- GBA para a geração e troca de chaves
- OAuth2 para autenticação
- BEST, segurança eficiente da bateria para dispositivos de comunicação para máquina de produtividade muito baixa (MTC) [21]

Embora a equipe de engenharia possa usar qualquer conjunto que atenda aos requisitos mencionados anteriormente, a utilização de um conjunto de protocolos de comunicação padrão reduzirá o número de erros que serão observados em campo. Isso ocorre porque os especialistas em segurança da informação e criptografia estão envolvidos no desenvolvimento de protocolos padronizados.

As propriedades de segurança da tecnologia de comunicações celulares baseadas no 3GPP, incluindo as tecnologias de rede padronizadas do LPWA NB-IoT e LTE-M, podem ser encontradas no documento GSMA PRD CLP.14 [4]

6.19.1 Risco

Embora não seja necessário dizer que a segurança das comunicações é um requisito, às vezes é confuso o motivo pelo qual a segurança é um requerimento. A segurança das comunicações não garante apenas que um invasor não possa ler os dados. Também garante que:

- Um endpoint não possa ser falsificado
- Um serviço crítico não possa ser falsificado
- Mensagens violadas possam ser detectadas
- Alterações no software ou configurações de segurança possam ser feitas de forma segura.

Sem segurança nas comunicações, não há garantias quanto à qualidade, confiabilidade ou privacidade de um produto ou serviço de IoT.

6.20 Autenticando a identidade de um endpoint

Se cada endpoint possui uma identidade criptograficamente exclusiva, como um número de série exclusivo, o dispositivo deve ser capaz de provar que ele realmente representa esse número de série. Para fazer isso, a TCB deve assinar criptograficamente uma mensagem com uma chave conhecida apenas pela TCB e pelo serviço de back-end de IoT, uma complexidade que pode ser gerenciada com tecnologias como GBA. A mensagem deve conter a identidade exclusiva (número de série ou outro token) e os metadados respectivos para o endpoint.

A mensagem a ser assinada pela TCB também deve conter um desafio emitido pelo sistema de back-end. Isso nega a possibilidade de um invasor reproduzir uma mensagem de autenticação já enviada da TCB para o back-end. Se houver entropia suficiente no desafio, não há possibilidade de repetição de mensagem.

Para desafiar a identidade de um endpoint:

- Receber uma solicitação do endpoint que contém o token de identidade exclusivo
- Gerar um desafio exclusivo e enviá-lo ao endpoint
- Receber a resposta do desafio do endpoint contendo a assinatura e a mensagem
- Verificar se a assinatura está correta usando a chave compartilhada
- Garantir que a mensagem assinada contenha o token de identidade correto e quaisquer outros metadados relevantes
- Reconhecer a assinatura verificada

Para processar um desafio:

- Conectar-se ao sistema de back-end
- Receber a identidade criptográfica do sistema de back-end
- Autenticar criptograficamente a identidade do sistema de back-end usando a TCB
- Enviar uma mensagem contendo a identidade do endpoint e outros metadados para o back-end
- Receber um desafio do back-end
- Gerar uma mensagem contendo o token de identidade exclusivo, os metadados e o desafio
- Assinar a mensagem
- Enviar a mensagem e sua assinatura para o back-end
- Verificar se o sistema de back-end aprovou a mensagem assinada
-

6.20.1 Risco

O risco de não implementar esta recomendação é que os endpoints serão clonáveis ou vulneráveis a ataques de falsificação de identidade. Isso pode abrir a infraestrutura da organização para ataques de concorrentes e hackers. Concorrentes podem usar a falta de autenticação de identidade do endpoint para construir uma plataforma rival a partir da mesma Lista de Materiais, mas a um custo menor.

Alternativamente, um concorrente pode usar a falta de autenticação para vender hardware autônomo da infraestrutura da organização. Essas situações podem resultar em perda de receita para a empresa e em aumento de despesas operacionais, pois o concorrente pode se beneficiar do uso da infraestrutura de rede da empresa, mesmo que não esteja pagando para usá-la. Como a largura de banda da rede, os servidores em nuvem, o uso da CPU, o uso do disco e outros recursos têm um custo mensurável, esse tipo de negócio parasitário pode ter um impacto sério em uma organização vulnerável.

7 Recomendações de alta prioridade

Recomendações de alta prioridade representam o conjunto de recomendações que devem ser implementadas, mas apenas se a arquitetura do endpoint assim o exigir. Por exemplo, nem todas as arquiteturas de endpoint exigem embalagem de produto resistente a violação. Essas recomendações devem ser avaliadas para determinar se o business case o considera um requisito.

7.1 Utilize a memória interna para informações sensíveis

Quando possível, os processadores devem usar memória interna da CPU para o processamento de informações seguras centrais e chaves criptográficas não contidas em uma âncora de confiança. Isso garantirá que, se um hacker estiver monitorando ou for capaz de manipular o barramento de memória, ele não obterá informações do núcleo, mas verá apenas os efeitos do uso desses segredos em uma aplicação em execução.

Este modelo criará longevidade em relação aos segredos criptográficos, forçando o invasor a desvendá-los. Em vez disso, o invasor precisará contar com a manipulação de bits na RAM, o que equivale aos efeitos do uso de tais segredos. Isso exigirá que o invasor altere os bits na memória toda vez que os segredos forem usados internamente, aumentando consideravelmente a complexidade do ataque.

Nem todos os sistemas operacionais definem modelos para utilizar RAM interna para o processamento de segredos. Portanto, pode ser necessário que a equipe de engenharia implemente isso sozinha. Embora esse processo não seja difícil, também não é trivial. O código executável deve garantir que suas rotinas de memória usem áreas específicas garantidas para representar a memória interna do processador. Isso pode exigir trabalho extra, dependendo do sistema operacional e do conjunto de ferramentas do compilador utilizado.

7.1.1 Risco

A maioria dos microprocessadores e algumas CPUs têm uma pequena quantidade de SRAM interna dedicada à execução do código a partir da EEPROM ou NVRAM internas. Essa SRAM é normalmente inacessível a periféricos externos, a menos que seja propositalmente exposta usando tecnologia como DMA. Se mantidos em sigilo, os segredos criptográficos processados pelo código têm uma probabilidade muito menor de serem expostos a hackers capazes de interceptar as comunicações da RAM.

Embora não seja um risco alto, os segredos criptográficos não devem passar por barramentos acessíveis por terceiros a fim de diminuir a possibilidade de ataque. Hackers bem equipados capazes de interceptar comunicações de RAM a velocidades potencialmente altas podem capturar dados como segredos criptográficos. No entanto, seria necessária uma engenharia reversa habilidosa para capturar mensagens na RAM que poderiam ser atribuídas a operações criptográficas.

Como resultado, embora essa seja uma recomendação importante, pode não ser essencial garantir a segurança física. Se as chaves criptográficas centrais forem armazenadas na âncora de confiança e somente as chaves de sessão forem processadas pela aplicação, o processamento das chaves na RAM externa provavelmente não resultará em um comprometimento imediato. No entanto, isso pressupõe que a arquitetura criptográfica limita as chaves expostas àquelas que não são críticas para as principais operações de IoT, como rotação de chave, geração de chave de sessão e revogação de certificado.

7.2 Detecção de anomalias

A modelagem do comportamento do endpoint é uma parte essencial da segurança da IoT. Isso ocorre porque um endpoint comprometido pode ser indistinguível de um endpoint se comportando normalmente se apenas interações bem-sucedidas com o dispositivo forem

registradas e analisadas. Para uma perspectiva mais abrangente de um ambiente de IoT, a pegada comportamental completa de um dispositivo deve ser catalogada para identificar anomalias que possam ser indicativas de comportamento mal intencionado.

O comportamento anômalo oriundo de um endpoint pode incluir:

- Reinicializações erráticas ou redefinições de dispositivos
- Deixar ou ingressar em uma rede de comunicação em intervalos erráticos
- Conectando-se a serviços de endpoints anormais ou conectando-se a serviços de endpoints em horários inadequados
- Uma impressão digital de tráfego de rede significativamente diferente da normal
- Múltiplas mensagens mal-formadas enviadas do endpoint para endpoints do servidor

Se o comportamento normal de um tipo de endpoint for catalogado pelo provedor de serviços de IoT, a organização poderá identificar padrões de comportamento que devem indicar um desempenho anormal. Ao definir uma linha de base de comportamento e, em seguida, monitorar continuamente possíveis valores discrepantes, a organização pode diagnosticar mais rapidamente problemas de segurança e desempenho em ambientes de produção.

A catalogação da pegada comportamental também pode ajudar a organização a vincular mais rapidamente um conjunto defeituoso de funcionalidades a um recurso ou condição ambiental em particular. Isso pode levar a soluções de engenharia em um ritmo mais rápido do que se os dados comportamentais não forem coletados.

7.2.1 Risco

Sem a detecção de anomalias, pode levar um tempo excessivamente grande para detectar um endpoint comprometido no ecossistema IoT. Se o comportamento anômalo do endpoint for visível apenas fora das operações normais, a equipe administrativa poderá não ter motivos para desconfiar dele. No entanto, se a detecção de anomalias for implementada em todo o ecossistema, o comportamento malicioso pode ser detectado - e, portanto, contido - muito antes.

7.3 Use um gabinete resistente à violação

O dispositivo físico não deve ser apenas resistente a adulterações no nível do chip, ele também deve ser resistente à violação no nível do produto. O gabinete usado no produto deve fornecer proteção contra hackers ou usuários curiosos. Isso pode ser feito de várias maneiras:

- Circuitos que invalidam a NVRAM quando um gabinete é violado
- Sensores que queimam fusíveis de segurança quando luz é detectada
- Sensores que acionam um alerta quando a localização de um dispositivo estático é deslocada
- Revestimento de epóxi nos componentes do circuito principal

O uso dessas metodologias pode melhorar a resistência a adulterações de um endpoint físico. No entanto, pode ser mais rentável melhorar o design do próprio circuito. Embora essas metodologias diminuam a possibilidade de comprometimento por amadores ou concorrentes, elas não conterão analistas de segurança bem equipados e experientes.

Assim, esses métodos melhoram a capacidade da organização de garantir que o produto não pode ser adulterado enquanto estiver fora do controle do proprietário. Em outras palavras, se um usuário deixar o dispositivo em casa ou em qualquer outro lugar, um invasor deve ser impedido de conseguir acesso físico para comprometer o dispositivo, como também deve conseguir suplantar os controles de segurança invioláveis, para alterar e substituir o dispositivo. Isso neutraliza a habilidade de comprometer e substituir dispositivos rapidamente, o que é uma melhoria valiosa para a segurança física do dispositivo.

No entanto, se o modelo de ameaça ignora esse aspecto e se concentra em corrigir o ataque físico em geral, inclusive de invasores avançados e preparados, ele não corrige totalmente essa ameaça. Nesse caso, aditivos resistentes à adulteração retardarão um hacker, mas não deterão o invasor com experiência e perícia.

Assim, deve ser encontrado um equilíbrio entre o que é rentável e o modelo de ameaça do dispositivo em questão. Um caixa eletrônico é um bom exemplo de tal dispositivo. Para a segurança do caixa eletrônico, a resistência à adulteração no encapsulamento é necessária para garantir que um invasor não possa abrir e alterar este encapsulamento físico para, digamos, capturar dados de tarjas magnéticas e registrar números de acesso. No entanto, hackers experientes criaram componentes locais, skimmers, para serem adaptados sobre o caixa eletrônico. Com isso, a blindagem física só consegue alcançar parte do resultado desejado. O design da aplicação e do hardware deve dar o passo extra para diminuir os ataques físicos.

Engenheiros e executivos devem avaliar o modelo de ameaça de um determinado produto ou serviço e equilibrar o risco de ataque com as medidas de inviolabilidade implementadas no dispositivo. Cada tipo de resistência à adulteração incorrerá em um custo, que dependerá do processo, engenharia e materiais envolvidos. E, ainda assim, o esforço pode não resultar no nível de segurança desejado.

Um exemplo deste problema são os chips revestidos com com epóxi. Embora esse processo seja valioso, há duas coisas que um invasor pode facilmente fazer para contornar o uso de epóxi:

- Circuitos derivados originários do componente revestido de epóxi
- Remover o epóxi

Embora o epóxi oculte o componente do chip, ele não impede (e não é capaz de impedir) que os elétrons viajem através dos circuitos originários do chip revestido com epóxi. Assim, se informações críticas forem comunicados pelo barramento de hardware, o epóxi não impedirá o invasor de interceptar esses dados.

Além disso, o próprio epóxi pode simplesmente ser removido. Técnicas amadoras caseiras que surgiram nos últimos anos descrevem claramente um método prático para remover epóxi de um circuito usando produtos químicos e processos prontos para utilização. Embora

o processo possa ser cáustico e possivelmente perigoso, os procedimentos idealizados por engenheiros versados em engenharia reversa são sólidos e podem ser implementados por qualquer pessoa com um laboratório ou escritório adequadamente ventilados.

Assim, uma avaliação de risco deve ser realizada para medir claramente os benefícios da tecnologia resistente a violações em relação à facilidade de comprometimento. Se cada dispositivo deve ser simplesmente protegido de um invasor que deseja manipular ou abusar aleatoriamente com facilidade, a resistência à adulteração deve ser empregada. Se o requisito é que mitigar a possibilidade de hackers habilidosos interceptarem mensagens através dos barramentos de hardware, uma arquitetura de segurança mais resiliente para a aplicação e o sistema operacional deve ser priorizada acima da capacidade de resistir adulterações.

7.3.1 Risco

Conforme relatado na seção anterior, o risco de não implantar a resistência à adulteração varia enormemente com os requisitos do dispositivo. Se o requisito é que o dispositivo deve alertar o usuário se este foi fisicamente violado, quebrado ou alterado, a resistência à violação é importante. Se o requisito for que o dispositivo deve ser protegido da análise por um pesquisador ou analista de segurança amador ou qualificado, a segurança da arquitetura provavelmente é a solução correta para o risco.

Em ambos os casos, o risco de não implantar resistência à adulteração é tal que o usuário não será capaz de determinar se um invasor adulterou o dispositivo físico. Embora isso não signifique muito para aplicações com hardware robusto e reforçado e aplicações de arquitetura de segurança, isso significará muito para produtos que ofereçam serviços essenciais a seus usuários, como dispositivos médicos, sistemas de telemática e sistemas de segurança ou automação residencial.

7.4 Reforce a confidencialidade e integridade para/da âncora de confiança

Todas as comunicações de e para a âncora de confiança devem ser autenticadas e devem impor confidencialidade e integridade. A única exceção a esse modelo é se a âncora de confiança for interna ao núcleo do processador. Qualquer âncora de confiança externa, como um UICC, só pode ser confiável se as mensagens recebidas e enviadas puderem ser garantidas.

Para fazer isso, escolha âncoras de confiança que sejam capazes de autenticação e encriptação e valide se todas as mensagens contendo respostas a desafios são enviadas confidencialmente e, quando possível, com integridade verificável.

UICCs que podem ser gerenciados com o canal seguro são capazes de confidencialidade e integridade. O provedor de serviços de IoT deve discutir com a operadora de rede se a tecnologia UICC Secure Channel pode ser usada para ajudar na segurança da aplicação. No futuro, eUICC terá capacidade para aplicações de segurança. O canal pode, então, ser usado para facilitar a segurança da aplicação do endpoint do estágio do inicializador para o estágio da autenticação de rede.

Embora isso pareça um exercício simples, há sutilezas nesse processo. Testar cada aspecto da camada de comunicação é necessário. Algumas mensagens de várias âncoras de confiança podem não ser confidenciais ou ativadas com integridade. Por exemplo, uma

mensagem que indica se uma operação foi bem-sucedida ou falhou pode parecer benigna, mas deve ser protegida para garantir que um invasor não envie uma resposta personalizada, burlando a aplicação.

Algumas âncoras de confiança podem não ser capazes de integridade no canal de comunicação. Ter integridade é preferível, e esta deve ser usada para garantir que uma mensagem não tenha sido adulterada. Mas, para isso, é necessária uma base de confiança no processador host e na âncora de confiança, o que pode não ser o mais recomendável para a aplicação.

Como todos os sistemas embutidos podem vir a ser comprometidos por um hacker suficientemente equipado, pode ser um exagero exigir uma raiz de confiança em ambos os processadores, simplesmente para comunicações de barramento local. No entanto, em aplicações em que a segurança física é crítica, a integridade deve ser implementada.

7.4.1 Risco

O risco de não impor confidencialidade e integridade é interessante. Esse risco pode variar de um comprometimento completo do sistema a uma coleta bem-sucedida de informações. Isso ocorre porque certas mensagens *podem* ser testadas. Por exemplo, se uma TCB solicitar que a âncora de confiança verifique a integridade de uma mensagem, ela passará a mensagem pelo barramento de hardware por intermédio desta.

Se a âncora de confiança for interna à CPU, é improvável que um hacker possa alterar essa mensagem sem equipamento sofisticado e caro. No entanto, se a âncora de confiança for um chip separado na placa de circuito, pode haver uma oportunidade para o hacker alterar a mensagem, unindo o circuito e inserindo seu próprio hardware. Se, por exemplo, a âncora de confiança receber a mensagem e simplesmente responder à consulta declarando “Sim, esta mensagem é válida” sem qualquer integridade, a TCB não poderá verificar se a mensagem foi manipulada por um invasor com acesso físico ao barramento de comunicações.

Além disso, mesmo que a resposta seja verificada quanto à integridade, um invasor com acesso físico ao barramento pode simplesmente comprometer o circuito, capturar o pedido de mensagem da TCB, emitir sua própria mensagem para a âncora de confiança e deixar a resposta nesta âncora de confiança real passar através da TCB. Se o barramento de comunicação de hardware não estiver adequadamente protegido, esse ataque também será possível, impedindo a âncora de confiança de executar seu trabalho.

No entanto, esperar que a CPU e a âncora de confiança tenham senhas confiáveis internas individuais cria um paradoxo. Como uma CPU inicializável pode confiar em si mesma se ela mesma puder ser alterada por um invasor, mas a CPU precisa usar sua própria EEPROM para verificar a integridade da âncora de confiança? Isso cria um enigma, mas que pode ser resolvido.

Uma solução é inserir uma chave pública na ROM da CPU. Essa chave pode ser usada para verificar a integridade das mensagens enviadas pela âncora de confiança. Se uma mensagem arbitrária (a ser verificada) for transmitida pelo barramento de hardware para a âncora de confiança, esta poderá responder com uma mensagem assinada que *inclua a mensagem original* como parte da resposta. Isso verifica se a mensagem realmente se

originou da âncora de confiança e se a mensagem que está sendo processada é, de fato, a mensagem que se esperava que fosse processada. A única preocupação que resta é garantir que as peças usadas no preenchimento de mensagens garantam que as mensagens criptográficas não sejam reproduzíveis.

Com isso em mente, é fácil perceber que a criptografia pode falhar devido a problemas muito sutis, não apenas na criptografia, mas nos algoritmos que dão suporte às comunicações criptográficas. É por isso que implementar (corretamente) confidencialidade e integridade é tão importante.

7.5 Atualizações de aplicações over the air

Atualizar remotamente a imagem de uma aplicação do endpoint pode ser um processo simples e direto. A complexidade vem da excessiva engenharia da solução que não abordem falhas de segurança realistas. De uma perspectiva de armazenamento permanente, o processo de engenharia é muito simples:

- Definir um local para a cópia da aplicação ativo
- Definir um local para a cópia da aplicação de backup (se houver)
- Definir um local para a cópia da aplicação de emergência
- Se existir um espaço de backup para a cópia da aplicação, atualizar este espaço com a cópia ativa
- Verificar criptograficamente a cópia ativa usando a assinatura armazenada na TCB
    - Isso garante que a mídia de armazenamento não esteja corrompida, e que um hacker não modificou os bits durante o processo de gravação
- Fazer o download da nova cópia na íntegra ou em deltas e seus metadados e assinatura
- Corrigir a cópia ativa com os deltas
- Verificar a assinatura criptográfica usando a TCB
- Reinicializar com a nova cópia

Se o processo falhar em qualquer etapa, o sistema deve reverter para uma cópia de backup para garantir que a aplicação seja executada conforme necessário ou o sistema de emergência possa ser acionado para convocar e notificar o ecossistema de serviços de IoT que ocorreu uma falha.

A dificuldade vem da criação de um modelo de armazenamento que aborda dois problemas

- Um invasor tentando manipular o processo de atualização
- Uma anomalia de hardware

Sem um sistema de backup ou uma partição de emergência, o dispositivo não terá opção a não ser falhar. Como os sistemas embutidos normalmente não têm interfaces de usuário robustas, isso pode representar um ponto significativo de estresse entre a empresa e seus clientes. Falhar o mais eloquentemente possível é imperativo não apenas para a confiança do usuário, mas também para a confiabilidade do sistema.

É importante notar que alguns invasores possam querer corromper o processo de atualização de propósito, para forçar um sistema a se tornar permanentemente vulnerável. Por exemplo, se uma vulnerabilidade explorável for encontrada na versão ativa da

aplicação, mas uma correção estiver disponível na versão mais recente da aplicação.

O benefício desse modelo é que, mesmo que o invasor corrompa o processo de negociação da rede, o sistema de back-end terá a oportunidade de registrar esse evento. Se a rede back-end identificar que um nó está funcionando normalmente, exceto para atualizações, um alerta deve ser gerado para a administração determinar se esse endpoint está sendo violado.

7.5.1 Risco

Se o processo de atualização da aplicação over-the-air não for adequadamente projetado, isso poderá resultar em invasores inserindo remotamente códigos executáveis nos endpoints. Se o invasor tiver uma posição privilegiada na rede, isso poderia vir a afetar milhares de endpoints de uma só vez. O resultado do ataque pode variar de simples execução de código a negação de serviço (brickando os endpoints) ou alterar completamente a finalidade do endpoint.

7.6 Autenticação mútua impropriamente projetada ou não implementada

Nos ambientes de comunicação, os pares se comunicam pela semelhança de *identidade* do protocolo. Isso tem significados diferentes em diferentes contextos, mas em qualquer ambiente um *endereço* identifica o destino de uma mensagem. Qualquer módulo de comunicação que implemente um determinado protocolo é capaz de declarar que é o proprietário de um determinado endereço. Mesmo que uma implementação específica de um protocolo seja designada, ou emulada, para usar o endereço de hardware de um módulo de rádio local, não há nenhuma regra que garanta que um usuário possa alterar fisicamente a EEPROM desse módulo e alterar seu endereço de hardware. Mesmo se a implementação se recusar a permitir que um usuário altere o endereço de hardware dinamicamente, ele ainda poderá ser manipulado para alterá-lo. O resultado dessa funcionalidade é, essencialmente, a falsificação: ou o ato de assumir a identidade de outro computador para fins de interceptação de mensagens destinadas a este.

7.6.1 Autenticação de cliente

Todos os ambientes são vulneráveis a falsificação. Por exemplo, qualquer celular pode sinalizar que é o proprietário de qualquer “Identidade Internacional de Assinante Móvel” (IMSI, da sigla em inglês), seja ela verdadeira ou não. Qualquer notebook pode alterar seu endereço de rede, representando outros computadores na rede local (LAN). Independentemente da topologia atravessar um espaço físico ou um espaço aéreo, a identidade de um endpoint de comunicação pode ser representada por outro.

A proteção contra isso é autenticação. Por exemplo, na rede celular, qualquer pessoa com o equipamento certo pode reivindicar a propriedade de qualquer IMSI escolhido. No entanto, as operadoras móveis reforçam a autenticação codificando uma chave criptográfica no Módulo de Identidade do Assinante (SIM, da sigla em inglês), que é único para esse assinante (IMSI). Quando um dispositivo celular se comunica com uma estação base afirmando que está representando um determinado IMSI, a estação base emitirá um desafio criptográfico que só pode ser resolvido por alguém com a chave criptográfica exclusiva armazenada no SIM card aprovisionado para essa identidade específica. Se o hacker não puder resolver o desafio criptográfico, a estação base pode verificar que ele não representa o IMSI em questão e pode impedir que esse usuário se associe à rede.

O modelo relatado acima descreve a *autenticação orientada ao cliente.* Esse é o modelo em que o subsistema do servidor (incluindo estações base) permite que os clientes (endpoints) ingressem e saiam da rede, desde que eles possam autenticar criptograficamente sua identidade. No entanto, existe um problema inverso que expõe esses clientes a manipulação: *autenticação do servidor*.

7.6.2 Autenticação do servidor

No modelo 3GPP, apenas os endpoints (chamados de User Equipment) são autenticados. Os endpoints não autenticam as estações base às quais eles se conectam. Assim, qualquer estação base pode reivindicar servir em nome de qualquer operadora móvel. Indivíduos capazes de manipular ou construir uma estação base celular podem, então, representar qualquer operadora móvel de sua escolha. Uma estação base celular customizada custa atualmente menos de US$ 1.000 para ser construída, mas o resultado permite apenas a interceptação de mensagens na área local. Uma vez que a falsa torre é construída, a estação base pode representar uma operadora móvel local e interceptar chamadas telefônicas, mensagens de texto e até dados, a partir de endpoints na área local.

Novos protocolos de rede 3GPP, como UMTS e LTE, reforçam a autenticação mútua de ambas as entidades. Isso permite que os endpoints verifiquem criptograficamente se a estação base está servindo em nome da operadora móvel que ela afirma ser. Um invasor deve então quebrar a criptografia da operadora móvel para representar uma estação base, aumentando significativamente a complexidade, a dificuldade e o custo de um ataque.

7.6.3 Interceptadores de celulares ou falsas estações base

Há exceções a essa regra, no entanto, como os interceptadores de celular. Esses dispositivos, normalmente usados por contratados do governo, governos e serviços de inteligência, são codificados com chaves criptográficas fornecidas a essas entidades pelas operadoras móveis, para fins de segurança nacional. Esses sistemas usam essas chaves para interceptar passivamente as comunicações bidirecionais ou para executar ativamente ataques “man-in-the-middle” contra alvos específicos.

No modelo moderno de ameaças de comunicação, no entanto, o acesso a essa tecnologia não se limita aos agentes de governo e das áreas de inteligência. Atualmente, esses sistemas podem ser construídos a partir de componentes que custam apenas algumas centenas de dólares, resultando em uma estação base de baixo custo, capaz de interceptar ou representar comunicações celulares.

7.6.4 Segurança em comunicações é segurança ponta a ponta

A criação de interceptadores de celular ajuda a resumir essa seção de forma bastante adequada, abordando a ideia de que a segurança das comunicações não é absoluta. Apenas protege o canal de comunicação entre duas entidades. Essas entidades, no entanto, atuam como portais que permitem a entrada e saída de dados dos ecossistemas aos quais essas entidades estão conectadas.

Por exemplo, um determinado SIM card pode ser fornecido para uso em um sistema de controle industrial, como um dispositivo de monitoramento de poço de petróleo. Um SIM card, por design, é um componente removível. Qualquer pessoa com acesso físico ao dispositivo de monitoramento de poços de petróleo pode extrair o SIM card e colocá-lo em

um notebook. Se o notebook tiver um software que possa simular a funcionalidade do dispositivo, o servidor de back-end não conseguirá diferenciar entre o dispositivo para petróleo real e o notebook. No entanto, o laptop será autenticado na rede celular por causa do SIM card! Assim, a rede móvel autenticou o SIM card, mas não o notebook.

7.6.5 Resolvendo autenticação mútua

Cada par em um ecossistema de IoT deve autenticar todos os outros pares que participam dele. Para conseguir isso, uma TCB deve ser usada para garantir que a arquitetura criptográfica adequada esteja conduzindo a tecnologia de comunicação. A autenticação mútua não pode ocorrer se as chaves forem facilmente expostas a invasores. Reveja a seção TCB deste documento para mais informações.

Uma vez autenticado, cada par deve criptografar e assinar mensagens enviadas a outros pares na rede. Cada par que receba uma mensagem deve validar criptograficamente os dados antes de agir sobre ela. Como nem todos os protocolos de comunicação são capazes de autenticação mútua ou possuem criptografia forte, é imperativo que o engenheiro de aplicações projete um protocolo suficiente para impor confidencialidade e integridade, em vez de confiar no protocolo de comunicações.

Protocolos ainda mais robustos que incorporam autenticação mútua, como o LTE, não abordam a segurança da infraestrutura além da rede de comunicações do celular. Somente a segurança do protocolo de camada superior pode resolver o risco de fragilidades na infraestrutura além do controle da operadora móvel.

7.6.6 Risco

O risco de não aderir a fortes medidas de segurança para a aplicação é que o endpoint deve confiar na segurança da camada de comunicações. Conforme descrito nesta recomendação, pode não ser adequado confiar somente na rede para resolver os problemas de segurança na aplicação. Mesmo que a operadora móvel seja confiável, as mensagens podem passar por várias partes da infraestrutura de rede, que não pertencem ou não são controladas pela operadora móvel, antes que os dados cheguem aos servidores do provedor de serviços de IoT. Portanto, o provedor de serviços de IoT corre o risco de qualquer pessoa com o controle desses sistemas interceptar, alterar ou fabricar mensagens para/de sistemas endpoint.

7.7 Gerenciamento de privacidade

Um aspecto imperativo da tecnologia da IoT é sua capacidade de conectar o mundo físico ao mundo digital. O resultado disso é uma brecha na privacidade, já que o ambiente físico do usuário está diretamente associado às coisas que ele gosta e visualizam online. Isso pode causar efeitos indesejáveis ao longo do tempo.

Como resultado, é importante que os prestadores de serviços de IoT considerem a privacidade de seus consumidores e desenvolvam interfaces de gerenciamento de privacidade que sejam integradas no endpoint, quando possível, e na interface web do produto ou serviço.

Essa tecnologia deve permitir que o usuário determine quais atributos de sua privacidade estão sendo utilizados pelo sistema, quais são os Termos de Serviço e a capacidade de

desativar a exposição dessas informações à empresa ou a seus parceiros. Esse sistema de granularidade e de desativação ajudará a garantir que os usuários tenham o direito e a capacidade de controlar as informações que compartilham sobre si mesmos e sobre seu mundo físico.

7.7.1 Risco

Os riscos potenciais de não proteger a privacidade do consumidor são muitos. Problemas de persecução, assédio, perfil, ameaças e muitas outras consequências são resultados realistas e práticos de não se proteger os dados do usuário

7.8 Identidades únicas e privacidade de endpoint

Cada endpoint é conhecido digitalmente por uma impressão digital. Essa impressão digital é composta de endereços, números de série e identidades criptográficas exclusivas do endpoint específico. No entanto, esses tokens também podem associar diretamente um dispositivo a um usuário, local ou serviço específico. Em muitas situações, isso é indesejável. Por exemplo, os smartphones podem ser rastreados porque o endereço Wi-Fi interno do telefone foi usado ao procurar ativamente por pontos de acesso 802.11. Esses endereços poderiam ser rastreados enquanto trafegam de um local para outro. Isso permitiria que qualquer pessoa pudesse associar um determinado endereço Wi-Fi a um usuário específico e assistir a seus movimentos em todo o mundo. Para combater isso, os fabricantes de software smartphones geram endereços aleatórios de usuários Wi-Fi ao procurar pontos de acesso, tornando praticamente impossível rastrear os telefones dessa maneira.

Os endpoints IoT podem ser rastreados de maneira semelhante por meio de endereços Bluetooth Low Energy (BLE), endereços 802.15.4, Wi-Fi ou até mesmo IMSI celular. Sempre que possível, o provedor de serviços de IoT deve desenvolver sua tecnologia endpoint de forma que um endereço de rádio aleatório seja usado para conectar-se a novos ambientes, permitindo que a privacidade do usuário permaneça intacta.

Isso também é verdadeiro para chaves criptográficas, como chaves públicas SSH. Embora os usuários normalmente desejem que suas chaves públicas sejam conhecidas por outros, as chaves públicas criptográficas nos endpoints irão expor a identidade do usuário de um endpoint específico, o que não é desejável. Em vez disso, o usuário deve poder selecionar se deseja que sua identidade seja conhecida quando estiver se conectando a um novo ambiente.

7.8.1 Risco

O não abrandamento adequado desse risco permitirá que usuários com endpoints móveis sejam rastreados à medida que seus dispositivos saiam e entrem nas redes. Isso abre brechas significativas na privacidade que equipes jurídicas, legisladores e até empresas de seguro estão analisando no momento. Não implementar adequadamente a privacidade para diminuir a possibilidade de rastreamento pode expor um novo provedor de serviços de IoT a consequências legais em um futuro próximo.

7.9 Executar aplicações com níveis apropriados de privilégios

As aplicações em execução em um endpoint geralmente não exigem privilégios de superusuário. Na maioria das vezes, as aplicações exigem acesso a drivers de dispositivo

ou a uma porta de rede. Enquanto alguns desses dispositivos, portas ou outros objetos podem exigir privilégios de superusuário para acessá-los inicialmente, este privilégios não são necessários para executar as operações subsequentes. Portanto, é uma boa prática usar somente privilégios de superusuário no início da aplicação para obter acesso a esses recursos. Depois, eles devem ser descartados.

Descartar privilégios de superusuário é um processo comum que é bem documentado e foi implementado excepcionalmente bem em aplicações como o Secure Shell (SSH), o Apache2 e outros servidores bem desenvolvidos. O processo geralmente engloba:

- Iniciar a aplicação com privilégios elevados
- Acessar todos os recursos que requerem privilégios elevados
- Reconhecer a identidade de usuário (por exemplo, ID de usuário UNIX e ID de grupo) que pode executar a aplicação
- Alterar completamente a identidade do processo para o ID do usuário / grupo de destino, removendo, assim, privilégios de superusuário da aplicação em execução

Um modelo mais complexo pode ser visto na implementação SSH do *privsep*, que executa um serviço privilegiado cuja única finalidade é inicializar a aplicação principal sob uma identidade de usuário/grupo de destino. Dessa forma, se o serviço for encerrado, ele poderá ser reiniciado facilmente sem o comprometimento de recursos privilegiados.

Para mais informações consulte SSH segregação de privilégios:
http://www.citi.umich.edu/u/provos/ssh/privsep.html

7.9.1 Risco

Executar aplicações com níveis elevados de privilégios pode resultar em comprometimento total do sistema se uma única aplicação for comprometida. Como os privilégios de superusuário concedem a uma aplicação acesso total a todo o sistema em execução, não há como conter um invasor depois que ele compromete tal aplicação. Eliminar privilégios ajuda a contê-los e limita sua possibilidade de aumentar seu privilégio dentro do sistema embutido. Esta pode ser a diferença entre um comprometimento total do sistema e um pequeno contratempo.

7.10 Impor uma separação de tarefas na arquitetura de aplicações

Aplicações em execução em um endpoint devem ter diferentes identidades de usuário associadas a cada processo exclusivo. Isso garante que se uma aplicação for comprometida, uma aplicação separada no mesmo endpoint não poderá ser comprometida sem um segundo ataque bem-sucedido. Essa etapa extra exigida em nome de um invasor geralmente é um obstáculo crítico ao processo de desenvolvimento de exploração como um todo e aumenta o custo e a complexidade de um ataque contra um endpoint.

Por exemplo, um serviço de rede que permita que um usuário adquira informações sobre o estado do endpoint não deve também ser capaz de manipular a TCB sobre o mesmo processo. Essa capacidade estaria fora do escopo em relação à finalidade do serviço. Essas duas operações distintas devem ser tratadas em aplicações separados e executadas

em IDs de usuário distintos no sistema operacional local, ajudando a separar as obrigações da aplicação e a reduzir o risco de abuso se um componente for comprometido.

Para implementar isso corretamente, a proteção de memória deve ser ativada na arquitetura de hardware subjacente, e o sistema operacional deve ter um conceito de níveis de privilégio. O software restringido deve ser impedido de acessar recursos privilegiados, como drivers, arquivos de configuração ou outros objetos.

Os serviços devem fazer solicitações para acessar recursos privilegiados, mas por meio de uma API restrita, como uma chamada de sistema, para garantir que todas as mensagens sejam bem elaboradas e atendam aos requisitos da arquitetura de segurança.

O conceito de multicamadas de privilégio é um conceito com meio século de existência. No entanto, em sistemas embutidos, isso é frequentemente ignorado, já que os usuários não têm permissão para efetuar login no terminal e executar suas próprias aplicações. Como resultado, todos os serviços são frequentemente implantados como um usuário privilegiado. No entanto, isso é falho.

Cada aplicação ou serviço deve ser implementado usando um privilégio customizado. Na maioria dos ambientes, essa é uma identidade de usuário distinta. Essa separação de tarefas ao impor identidades de usuários diferentes garante que, se um serviço for comprometido, ele não poderá afetar diretamente os recursos usados por outro serviço no mesmo sistema. Para comprometer outros serviços e usuários, explorações secundárias devem ser encontradas no sistema operacional local para aumentar privilégios.

Isso requer planejamento e uma boa arquitetura de aplicações que use corretamente a separação de privilégios.

7.10.1 Risco

Se uma separação de tarefas não for aplicada, qualquer comprometimento com um único serviço no endpoint resultará em um comprometimento de todo o dispositivo, pois cada serviço ou aplicação em execução no dispositivo compartilhará o mesmo usuário e/ou identidade de grupo. Se a recomendação for implementada, um serviço com poucos privilégios comprometidos na rede não resultará imediatamente em comprometimento de todo o sistema.

Como essa recomendação é simples de implementar, é essencial para a segurança dos endpoints IoT. Deve-se salientar que muitas vezes é necessário um grande conhecimento para comprometer remotamente um serviço de rede. Se o invasor também for obrigado a elevar privilégios implementando uma exploração no nível do kernel, ou outra exploração secundária, para obter o controle do sistema completo, o invasor pode não ter tempo, habilidades ou equipamentos para executar o ataque.

Aumentar a dificuldade de um ataque com uma simples alteração de configuração como essa ajudará muito a garantir a longevidade do dispositivo.

Além disso, como os serviços comprometidos podem ser detectados por meio do monitoramento de processos e outras análises, qualquer comprometimento de serviço pode alertar o ecossistema de serviços que um dispositivo foi comprometido. Isso permite que os administradores atuem para proteger o sistema antes que seu comprometimento total seja

alcançado. Isso também permite que os administradores detectem e atualizem o software vulnerável antes do abuso desenfreado da vulnerabilidade em particular. Isso dá ao negócio uma vantagem significativa contra os invasores qualificados.

7.11 Reforçar a segurança da linguagem

As linguagens de programação possuem diferentes graus de segurança, dependendo do propósito da linguagem e de seu alto nível. Algumas linguagens fornecem instruções para limitar o acesso à memória física e impõem restrições sobre como ela é usada. A equipe de engenharia deve identificar uma linguagem que seja capaz de fornecer segurança em ambiente de execução da aplicação ou do binário resultante.

O compilador ou ambiente de execução deve ser protegido, sempre que possível, para restringir a possibilidade de uma vulnerabilidade ser usada por um invasor. Em um ambiente de execução bem definido, até mesmo uma falha de programação fácil de acionar pode ser extremamente difícil de explorar completamente. Isso pressupõe que os aprimoramentos de segurança sejam usados para proteger a maneira como a aplicação é executada, acessa a memória e tem suporte dos reforços de segurança do sistema operacional.

7.11.1 Risco

O risco de não proteger a linguagem de programação e a aplicação resultante é um elemento fácil de explorar. Alguns sistemas de programação, como o PHP, notoriamente possuem bugs e nunca devem ser usados por uma equipe de engenharia profissional. Outras linguagens, como o Python, são adequadas para ambientes de produção, mas possuem riscos de segurança sutis que devem ser avaliados. Assim, a volatilidade do risco resultante pode estar em qualquer lugar, desde um nível crítico até um nível benigno. A equipe de engenharia deve usar o processo de avaliação de riscos e modelagem de ameaças para avaliar de maneira adequada qual linguagem é a melhor para seu ambiente de produção.

7.12 Implementar teste permanente de vulnerabilidades

A realização de uma auditoria de segurança apenas no momento da implantação não é suficiente para a maioria das implantações de IoT, nas quais novos endpoints podem ser introduzidos em campo e configurados a qualquer momento. Recomenda-se usar uma abordagem de teste permanente de vulnerabilidades para obter uma detecção antecipada de fragilidades de software do endpoint e de configurações inseguras.

A implementação de uma estratégia permanente de teste de vulnerabilidades pode fornecer uma detecção rápida e o gerenciamento antecipado das ameaças identificadas, aumentando a velocidade de mitigação e reduzindo a duração da exposição a ameaças.

Uma estratégia completa de teste permanente de vulnerabilidades deve fornecer uma maneira automática e programada de executar: pesquisa de ativos para criar um inventário dos ativos acessíveis, identificação e análise destes, verificação e exploração de vulnerabilidades conhecidas, verificação de configurações inseguras, e relatórios e alertas apropriados que devem ajudar na correção.

7.12.1 Risco

O risco de não implementar uma estratégia de teste permanente de vulnerabilidades é que as auditorias de segurança podem ser executadas apenas uma vez no momento da implantação, mas novos enpoints e configurações nunca são avaliados. Essa situação pode levar a um conjunto de terminais vulneráveis que nunca são identificados como expostos até que sejam comprometidos por um invasor.

8 Recomendações de media prioridade

O conjunto de recomendações de média prioridade engloba o conjunto de recomendações que são relevantes dependendo das opções de design da tecnologia endpoint. Por exemplo, impor aprimoramentos de segurança no nível de sistema operacional só é válido se houver um em execução no endpoint. Se o endpoint for composto por uma aplicação de kernel monolítico ou por um Sistema Operacional em Tempo Real (RTOS, da sigla em inglês) embutido com uma única aplicação incorporada, a recomendação poderá não se aplicar. Onde as recomendações se aplicam ao design do endpoint, elas devem ser implementadas.

8.1 Reforce os aprimoramentos de segurança no nível do sistema operacional

Aplicações executadas em um sistema operacional devem ser desenvolvidas para usar (de forma transparente ou intencional) os aprimoramentos de segurança deste sistema e do kernel subjacentes. Isso inclui tecnologias como:

- ASLR
- Memória não executável (Stack, Heap, BSS, ROData, etc.)
- Proteção de desreferenciação do user-pointer (UDEREF)
- Proteção contra vazamento de estrutura (divulgação de informações)

Cada sistema operacional usado em um sistema embutido fornecerá diferentes variações e combinações dessas tecnologias, às vezes sob nomes diferentes. Determine o que o sistema operacional e o kernel são capazes de fornecer e ativar essas tecnologias, sempre que possível, para aprimorar a segurança das aplicações.

O desafio vem de identificar o que cada sistema operacional é capaz de fazer. Por exemplo, aplicações em execução em plataformas que não possuem Unidade de Gerenciamento de Memória (MMU, da sigla em inglês) podem não ser capazes de ASLR. No entanto, o equivalente de UDEREF pode ser aplicado mesmo em ambientes com apenas uma unidade de proteção de memória (MPU, da sigla em inglês). Avalie qual tecnologia está sendo usada e seus recursos, e determine qual nível de segurança pode ser obtido por meio da combinação de proteções de arquitetura, kernel, sistema operacional e aplicações.

8.1.1 Risco

Não impor essa recomendação resultará em um ambiente de execução da aplicação que é substancialmente mais fácil de explorar. Esses aprimoramentos restringirão significativamente o número de invasores capazes de explorar um serviço vulnerável.

Assim, se uma aplicação desenvolvida pela organização tiver uma falha de segurança que possa ser usada para obter recursos de execução remota de código, a possibilidade de abuso pode ser reduzida pela aplicação de ASLR, NX, UDEREF e outras tecnologias. Isso limitará a possibilidade de um invasor desenvolver uma exploração em um período razoável, já que o invasor precisará usar técnicas avançadas e desafiadoras que devem ser customizadas individualmente para cada alvo. Isso aumenta não apenas a dificuldade, mas o tempo e o investimento necessários para viabilizar uma exploração totalmente funcional.

Sem esses aprimoramentos, uma exploração totalmente funcional pode ser desenvolvida usando softwares prontos para uso e disponíveis gratuitamente dentro de horas.

8.2 Desabilite tecnologias de depuração e teste

Quando um produto está sendo desenvolvido, ele é frequentemente ativado com tecnologias de depuração e testes para facilitar o processo de engenharia. Isso é totalmente normal. No entanto, quando um dispositivo está pronto para entrar em produção, essas tecnologias devem ser removidas do ambiente antes da definição de Configuração Aprovada.

A configuração aprovada com a qual um produto é implantado nunca deve conter interfaces de depuração, diagnóstico ou testes que possam ser usadas por um invasor. Essas interfaces são:

- Interfaces de linha de comando
- Terminais com depuração detalhada, diagnóstico ou mensagens de erro
- Portas de depuração de hardware, como JTAG ou SWD
- Serviços de rede usados para depuração, diagnósticos ou testes
- Interfaces de administração, como SSH ou Telnet

Todas essas tecnologias deveriam ser desabilitadas na configuração aprovada.

Portas seriais que podem ser removidas pelo sistema também devem ser fisicamente removidas da placa de circuito. No entanto, muitas vezes, portas seriais, como UART/USART, são habilitadas por meio de pinos de hardware no microcontrolador ou no processador. Se esses pinos ainda estiverem habilitados como um terminal, um invasor pode simplesmente tocar nos pinos para interagir com ele. Remover a própria porta serial física, como uma interface DB9, não o desativa.

Além disso, a depuração de portas, como JTAG e SWD, não deve ser simplesmente desativada por meio de software. Esses dispositivos devem ser desativados, alterando os fusíveis ou bloqueios de segurança. A desativação dessas tecnologias a partir do software oferece uma janela de oportunidade para um invasor se conectar ao JTAG, SWD ou a uma interface de depuração de hardware semelhante antes do momento em que o software

desativa a interface. Essa oportunidade é geralmente suficiente para que um invasor seja bem-sucedido.

8.2.1 Risco

Sem implementar essa recomendação, as organizações ficam vulneráveis à extração de informações críticas da unidade central de processamento. Isso pode permitir que os invasores carreguem seu próprio firmware em NVRAM ou EEPROM, o que lhes permite extrair ou alterar informações críticas que comprometem ainda mais a rede ou o dispositivo IoT.

Desativar portas de depuração é uma etapa essencial para garantir a integridade do produto ou serviço IoT. No entanto, é importante que a organização avalie o risco de desativar essas tecnologias e avaliá-las em relação ao benefício de poder diagnosticar e depurar problemas identificados em campo. Pode ser significativamente mais desafiador corrigir falhas em nível de produção do produto, se não houver como depurar um sistema em execução.

8.3 Memória contaminada via ataques baseados em periféricos

Os sistemas de processamento dependem da consistência para garantir que a saída dos algoritmos seja previsível em relação a um conjunto de dados fornecidos. Os sistemas de processamento também esperam que os componentes atuem de maneira confiável, e que, para cada bit escrito, este seja estável e inalterado até ser executado pelo processador. Dentro de sistemas fechados, esta teoria é aplicável. Quando ocorrem anomalias nesse modelo, elas podem comprometer ou simplesmente danificar um ambiente de processamento.

A segurança da informação apresenta a classe de anomalias propositalmente induzidas para obter acesso a objetos que de outra forma estariam inacessíveis. Uma janela abusiva para a indução de comportamento anômalo benéfico para um invasor é o acesso direto à memória (Direct Memory Access - DMA). Simplificando, o DMA é uma tecnologia que os processadores podem usar para permitir que componentes externos (periféricos) obtenham acesso à memória do processador principal sem interferência da CPU. Em outras palavras, a CPU pode conceder um acesso direto periférico a uma região de memória. Este periférico pode ler ou gravar nessa região da memória.

Se o processador não restringir adequadamente a região de memória utilizável pelo periférico, este poderá ter acesso a mais do que o necessário da memória principal para a funcionalidade pretendida. Em outras palavras, se o periférico (digamos, um controlador ethernet) receber uma região DMA destinada a uso como um buffer circular para quadros ethernet recebidos, e a região DMA alocada compreender toda a extensão da memória principal, o firmware no controlador ethernet pode ler e gravar arbitrariamente em toda a memória do sistema. A CPU não terá como impedir que o firmware do controlador ethernet grave na memória.

O resultado deste ataque é duplo. Os dados podem ser vazados da memória principal e codificados em pacotes de rede ou informações de aplicações para infiltração secreta ou imediata. Como alternativa, um invasor pode inserir secretamente um backdoor (malware) na memória principal sobrescrevendo o código executável de uma aplicação.

Do ponto de vista do processador, pouco pode ser feito para identificar se uma janela de memória excessivamente permissiva foi abusada por um dispositivo periférico mal-intencionado. Para combater esse ataque, identifique se o processador usado no sistema endpoint é capaz de restringir o DMA a pequenas regiões de memória previsíveis. Nesse caso, verifique se cada região da memória é definida para cada dispositivo periférico que a requer. Não habilite a janela de memória arbitrária, quando possível, para periféricos.

Alguns processadores podem não permitir restrição segmentada no tamanho ou local na memória linear ou virtual de uma janela DMA. Como os ataques de DMA devem ser considerados uma ameaça real aos endpoints IoT para aplicações essenciais, avalie se faz sentido considerar um processador alternativo com recursos mais granulares.

Para plataformas que expõem portas como IEEE1394, Thunderbolt, Express Card ou outras portas que permitem acesso direto a DMA de interconexão de componentes periféricos (PCI), ataques prontos e baratos já estão disponíveis.

Para plataformas onde um ataque baseado em DMA requer o abuso de um componente de hardware local, a dificuldade certamente aumentará, mas não está fora do escopo de um compromisso de segurança baseado em regravar o firmware de um periférico para subverter o DMA por comprometimento de um endpoint local. Custo, tempo e experiência, no entanto, serão um fator, fazendo com que o agente provavelmente seja um invasor patrocinado (pago).

8.3.1 Risco

A escolha de não restringir a capacidade do DMA de ser usado por componentes externos pode sujeitar a plataforma a um comprometimento total ou, pelo menos, a extração de segredos principais, dados centrados na privacidade ou propriedade intelectual do endpoint.

8.4 Segurança da interface de usuário

Endpoints IoT que possuem interfaces de usuário, como telas sensíveis ao toque, displays avançados ou tecnologias de interface alternativas, devem ser capazes de processar informações para o usuário e obter informações de maneira segura.

Embora os atributos da interface do usuário, como senhas, já tenham sido abordados neste documento, há alguns problemas mais sutis que devem ser discutidos:

- Sistemas de alerta
- Confirmações de ação

Quando uma anomalia ocorreu, como adulteração física ou uma aplicação se comportando de maneira não intencional, o usuário deve receber um alerta visível. Como alternativa, o usuário deve poder revisar alertas do sistema a partir da interface do usuário.

Além disso, todas as ações executadas pelo dispositivo que são conduzidas por codificações ou transições continuadas de uma interface para outra devem ser confirmadas pelo usuário. Um exemplo disso é se a câmera do dispositivo lê um código QR ou um pedido de interação NFC ou RFID e que o dispositivo se conecta a uma URL. Nesses casos, o usuário deve ser solicitado a confirmar a ação e validar quando executada e é

aceitável. O usuário deve ter a opção de cancelar a ação. O usuário deve poder visualizar todos os detalhes sobre a determinada ação, incluindo a URL completa que será conectada.

8.4.1 Risco

Se esta recomendação não for implementada, os usuários estarão vulneráveis a ataques que não podem ser detectados. Embora alguns projetistas de sistemas apreciem a perfeição da transição de um chip RFID para, digamos, o site do produto correspondente, pode haver efeitos indesejáveis desse comportamento. Os usuários podem ser forçados a visualizar conteúdos indesejáveis sem o seu consentimento ou os usuários podem ser levados a visitar sites ou realizar ações que enfraquecem sua postura de segurança ou privacidade.

Além disso, os usuários que têm dificuldade em analisar seus alertas podem não entender os riscos de usar um dispositivo possivelmente violado. Isso pode diminuir a segurança física do usuário e o colocar em risco.

8.5 Auditoria de código de terceiros

Sempre que uma seção de código, como um gerenciador de inicialização, é um componente crítico na construção de uma plataforma de ambiente de execução segura, ela deve ser auditada em busca de riscos. Se um inicializador puder ser manipulado por um invasor para executar um código não confiável, ou para ignorar a sequência de autenticação, ele será ignorado. Isso anularia o investimento, o tempo e a experiência utilizados pela organização na implantação dessa tecnologia, aniquilando as despesas de engenharia.

Uma falha na segurança nessa área também pode resultar em uma vantagem do concorrente contra a empresa por meio de falsificação, abusos de API, interceptação de dados, clonagem de dispositivos e até mesmo remarcação de dispositivo. Assim, é imperativo que seções críticas de código sejam auditadas por uma terceirizada aprovada, para garantir que a tecnologia não corra risco de abuso. Portanto, para encontrar uma equipe de segurança da informação adequada para executar a auditoria, avalie quais tipos de código serão auditados. Normalmente, nesse modelo, isso significa: C, Assembly e, possivelmente, C ++ ou Java.

Identifique uma equipe bem versada nessas linguagens, bem como a arquitetura subjacente. Embora muitas equipes de segurança da informação realizem auditoria de código-fonte, muitas delas não podem realizar auditorias na plataforma específica usada pelo negócio de IoT. Cada plataforma tem diferenças sutis, e é melhor usar uma equipe familiarizada com a plataforma que está sendo usada.

8.5.1 Risco

Embora a contratação de terceirizadas para avaliar a tecnologia desenvolvida internamente possa ser um desafio, é um requisito para a segurança. Isso ocorre porque os engenheiros que desenvolvem a tecnologia devem ser capazes de mostrar que sua arquitetura é confiável. Isso é difícil de fazer se os engenheiros que desenvolvem a arquitetura forem os únicos a revisá-la. Os engenheiros tendem a visualizar sua base de código a partir da arquitetura que eles tentaram projetar e implementar, e não a partir da implementação real.

Assim, as terceirizadas são frequentemente necessárias para encontrar sutilezas na arquitetura e implementação que possam causar falhas na segurança.

8.6 Utilize um APN privado

Nas redes celulares 3GPP, um nome de ponto de acesso (APN, da sigla em inglês) atua como uma rede privada configurada especificamente para um conjunto de dispositivos autenticados. Normalmente, um APN privado (também chamado de “APN seguro”) é uma rede privada acessível somente a dispositivos autenticados associados a uma empresa específica. Ao utilizar um APN, as empresas podem restringir o que os endpoints podem conectar à sua infraestrutura de serviços pela rede do celular. Isso ajuda a reduzir a quantidade de usuários que têm acesso direto aos serviços de IoT na infraestrutura de back-end.

Outros atributos de um APN privado podem ajudar a diminuir a possibilidade de endpoints fraudulentos abusarem do ecossistema de IoT. Os firewalls podem limitar quais serviços ou computadores podem ser conectados a partir do APN. Um APN bem configurado impedirá que os endpoints façam conexões diretas entre si, o que impede que um endpoint comprometido migre horizontalmente através da infraestrutura de rede para outros endpoints.

Envolver-se com a operadora móvel ou com o operador de rede virtual móvel (MVNO) com o qual a organização está trabalhando para determinar quais tecnologias estão disponíveis no APN seguro. Outros serviços, como monitoramento, bloqueio de dispositivos anômalos e vinculação de identidades de usuários a ações, podem estar disponíveis.

8.6.1 Risco

Utilizar um APN privado pode aliviar muitos tipos de ataques. Por exemplo, os APNs privados permitem que a empresa reduza a quantidade de conexões que podem ser feitas do endpoint diretamente para a internet. Os endpoints nunca devem ter permissão para se conectar diretamente a recursos não confiáveis da internet. Somente organizações parceiras devem ser confiáveis e esses serviços devem ser autenticados.

Sem o uso de um APN privado, os endpoints comprometidos podem se comunicar com qualquer serviço ou protocolo da internet sem restrições. Isso pode permitir que um hacker invada o endpoint para lançar um ataque secundário em uma infraestrutura separada. Isso poderia envolver um ataque de negação de serviço (DoS), ou poderia ajudar a facilitar um ataque mais perigoso contra outra empresa, governo ou cidadão.

É notável, no entanto, que um APN privado não atenue o risco de um hacker ser capaz de comprometer o link de comunicação entre o endpoint e o APN privado. Além disso, o APN privado atua apenas como um gateway para os serviços de back-end e não impõe nenhuma segurança entre o APN e os serviços de back-end na rede privada do provedor de serviços de IoT. Essas possíveis brechas na segurança devem ser abordadas separadamente, independentemente das melhorias que são concedidas pelo uso de um APN privado.

8.7 Implemente limites de bloqueio ambiental

Componentes dentro de um sistema embutido são projetados para serem usados dentro de certos limites ambientais. Isso inclui níveis de voltagem, consumo de corrente, temperatura

ambiente ou operacional, e umidade. Cada componente é normalmente classificado para determinadas hierarquias de níveis aprovados. Se o dispositivo estiver sujeito a estados acima ou abaixo de uma determinada hierarquia, o componente pode agir de forma irregular ou comportar-se de uma maneira que seja útil para um invasor.

Portanto, é importante detectar alterações nesses níveis ambientais para determinar se o dispositivo deve continuar em execução ou se deve ser desligado. Deve-se notar, no entanto, que o desligamento pode ser um efeito esperado, e que o invasor possa abusar desta decisão de engenharia para alavancar uma negação de serviço. A equipe de engenharia deve avaliar esse modelo para determinar se é mais benéfico tentar desligá-lo ou permanecer online.

Independentemente disso, o modelo geralmente incorpora:

- Detecção de brown-out e black-out, quando a voltagem cai muito
- Proteção contra circuitos em picos de tensão para garantir que os níveis não excedam um limite
- Circuitos limitadores de corrente para garantir que o consumo de corrente não diminua ou exceda certos níveis
- Monitoramento interno da temperatura para CPUs, MCUs e outros componentes que controlam os indicadores internos
- Opcionalmente, os níveis de umidade podem ser avaliados para determinar se o ambiente está se tornando muito úmido ou árido

A temperatura é extremamente importante, pois altas temperaturas podem indicar um problema no circuito acionado pelo usuário, o ambiente ou até mesmo um problema de hardware ou software. Monitorar a temperatura permitirá que o sistema operacional ou a aplicação desabilite recursos (ou todo o dispositivo) para garantir que um incêndio ou outro problema não seja causado pelo endpoint.

Os níveis de temperatura baixa também alteram o comportamento de um dispositivo. Isso pode retardar o circuito ou fazer com que seus componentes reajam de maneiras inesperadas. Pode ser útil para um invasor se a temperatura puder causar uma anomalia previsível que afete a aplicação ou o circuito de maneira benéfica.

A dificuldade em limites de bloqueio se manifesta ao analisar a temperatura e a umidade. Os níveis de tensão e corrente devem ser mitigados pelos circuitos brown-out e black-out na placa de circuito ou no processador. Como os engenheiros poderão procurar os números relacionados aos limites de tensão e corrente de um chip, eles podem implementar facilmente proteções para esses problemas.

Para temperatura e umidade, a decisão de agir é um pouco mais desafiadora, pois esses níveis podem ser fabricados por um invasor sem tocar no dispositivo físico. No quesito temperatura, os níveis que podem ser indicativos de um evento de segurança emergentes devem fazer com que o dispositivo tome medidas adequadas para baixar a temperatura. No entanto, em ambientes críticos, como sistemas de controle industrial ou dispositivos médicos, o dispositivo deve tentar continuar realizando operações críticas, sempre que

possível. Se os níveis excederem um determinado ponto definido com o qual os engenheiros e líderes de negócios concordam, só então o dispositivo deve ser desligado.

8.7.1 Risco

Para tensão e consumo de corrente, o risco de abuso está relacionado a falhas e outros ataques de canal lateral que podem se beneficiar de alterações nesses níveis. Se a detecção de brown e black-out for implementada no processador, o risco de abuso é reduzido. Caso contrário, o risco está relacionado a picos de voltagem ou corrente que podem causar problemas de segurança com o dispositivo físico ou permitir que um invasor ataque instrumentos (e similares) para subverter a segurança dos componentes.

Esses problemas devem ser contornados através do uso de circuitos PCB que protege os componentes contra picos anômalos ou quedas de tensão ou corrente.

Para mudanças no nível ambiental que são dramáticas, o risco está relacionado à segurança do usuário. Altas temperaturas causadas pelo uso excessivo da CPU ou outras anomalias podem causar queimaduras, queimaduras químicas ou até incêndios.

8.8 Imponha limites e alertas de energia

Endpoints que fornecem serviços críticos para o usuário devem ser ativados com um limite de alerta que indica eventos relacionados à energia. Esses eventos podem incluir:

- Nível baixo de bateria
- Nível baixo crítico de bateria
- Eventos de Black-ou
- Eventos de semi-blecaute
- Alternar para eventos de backup de bateria

O usuário deve ser avisado a tempo para compensar a perda de energia. Isso pode ser feito ativando um LED que indique um estado de energia específico, como verde para OK, laranja para baixo e vermelho para crítico.

Os sistemas conectados à energia de corrente alternada devem ser configurados para avisar o usuário quando ocorrerem eventos de black-out ou brown-out. Além disso, o endpoint deve registrar esses eventos na memória permanente para garantir que o usuário e a administração possam recuperar as informações posteriormente. A informação deve ter o horário da ocorrência.

O desafio neste processo é identificar que a taxa de bateria está se esgotando e a energia extra necessária para notificar o usuário sobre uma mudança no estado desta. Tudo isso pode ser alcançado por meio de engenharia elétrica e não deve ser um processo muito desafiador para empresas de engenharia experientes.

8.8.1 Risco

Sem um sistema de alerta de energia bem definido, os usuários não poderão se preparar adequadamente para eventos possivelmente críticos relacionados à energia. Embora isso

possa ser benigno no caso de dispositivos simples, como contadores de passos, temporizadores e outros dispositivos wearable, e dispositivos mais críticos, como rastreadores pessoais, sistemas de telemática e sistemas de segurança doméstica, podem ser seriamente afetados com quedas de energia.

8.9 Ambientes sem conectividade back-end

8.9.1 Método

Endpoints, especialmente gateways ou endpoints que atuam como gateways, devem ser capazes de impor a segurança das comunicações mesmo em ambientes em que a conectividade com a rede de back-end não esteja disponível. Independentemente de essa falta de conectividade ser temporária ou não, o gateway ou o endpoint deve ser capaz de garantir a segurança como se o sistema de back-end estivesse disponível.

Para conseguir isso, a TCB deve ser usada para autenticar todos os pares para os quais o endpoint deve comunicar dados centrados na privacidade, configuração ou comando. A TCB pode ser usada para garantir que as mensagens enviadas e recebidas de seus pares estejam sendo enviadas e recebidas de uma entidade que tenha sido fornecida pela mesma organização. Isso reduz a probabilidade de que um dispositivo concorrente esteja sendo comunicado.

A interoperabilidade ainda pode ser alcançada pela comunicação com outros dispositivos que não podem ser autenticados. No entanto, o tipo de informação que é comunicada a esses dispositivos deve ser restrita a tipos de dados interoperacionais e indiferentes.

O desafio surge ao decidir quais endpoints serão autenticados e quais se comunicarão em texto puro. A organização deve decidir quais tipos de dados são classificados e devem ser mantidos em pares não autenticados. Quando essa classificação de dados for alcançada, a organização poderá determinar quais pares são razoavelmente confiáveis mesmo sem a assistência dos principais serviços de IoT.

8.9.2 Risco

O risco de implantar soluções em ambientes sem comunicação é que isso abre uma oportunidade para a concorrência abusar da infraestrutura. Os concorrentes podem prejudicar o negócio oferecendo interoperabilidade e usando sites sem conexão como campo de testes.

Ao invés disso, a organização pode optar por permitir a interoperabilidade, mas até certo ponto. Determinadas propriedades intelectuais e serviços centrais podem ser reservados apenas para pares autenticados que são validados por meio do uso de uma TCB. Isso ajuda a reduzir a exposição do negócio a problemas de propriedade intelectual e concorrentes.

8.10 Desativação e cancelamento de dispositivos

Todos os dispositivos endpoint têm um ciclo de vida, conforme discutido em outras partes deste documento. Alguns dispositivos devem ser desativados porque um usuário cancelou sua assinatura, enquanto outros dispositivos devem ser cancelados devido a comportamento anômalo ou adverso. Independentemente do motivo, a empresa deve estar preparada para desativar o dispositivo de forma segura usando a TCB e o modelo de comunicação.

Desativação, como discutido em outras partes deste documento, é o processo de desligamento de toda uma rede de dispositivos e serviços que lhes dá suporte. Um produto ou serviço que tenha sido suspenso por uma empresa, ou uma empresa que decide

encerrá-lo, deve desligá-lo de sua rede e dispositivos para diminuir o risco de abuso por parte de concorrentes que invadam a rede com a desativação.

Para isso, a TCB e os protocolos de suporte devem ser usados. Geralmente, o processo é:

- Emitir uma mensagem de cancelamento ao ecossistema de serviços
- Customizar uma mensagem ao endpoint exclusivo que a receberá
- Assinar a mensagem usando o PSK de desativação ou a chave assimétrica
- Enviar a mensagem para o endpoint
- Receber uma mensagem do endpoint reconhecendo criptograficamente a desativação
- Invalidar o endpoint na lista de dispositivos autenticados
- Proibir comunicações futuras deste endpoint

Do lado do dispositivo, a aplicação em execução deve:

- Conectar-se a serviços de back-end essenciais acima do ecossistema de serviços
- Consultar o serviço para mensagens críticas
- Receber a mensagem de desativação
- Verificar a assinatura da mensagem usando a TCB e a âncora de confiança
- Gerar a mensagem de confirmação e assinar criptograficamente usando o PSK customizado ou chave assimétrica
- Realizar a operação de desativação
- Retornar a mensagem ao serviço crítico

É importante que a mensagem seja assinada e preparada para transmissão antes da desativação, pois este processo inclui a invalidação e remoção de chaves de segurança da âncora de confiança. Devido a este processo, as chaves usadas para assinar a mensagem de desativação não mais estarão disponíveis. O serviço requer a recepção de uma mensagem com integridade verificável para garantir que o endpoint realmente receba e processe a mensagem.

A dificuldade com este processo é principalmente que a desativação de um dispositivo possivelmente comprometido presume que este não se considera comprometido ao ponto de aceitar o comando. Se tiver sido suficientemente comprometido, não poderá confirmar o comando de desativação.

Como resultado, é imprescindível que o sistema back-end executado no ecossistema de serviços proíba que o endpoint seja capaz de se comunicar com serviços críticos. Se o dispositivo tentar interagir com pares em rede ou serviços críticos, o sistema back-end deverá emitir um alerta e informar à administração que o evento anômalo ocorreu.

8.10.1 Risco

Os riscos de não implementar a desativação e o desligamento são muitos, desde o sequestro completo da rede inteira por invasores até permitir que dispositivos comprometidos continuem usando serviços em rede. O risco mais comum está associado a usuários que cancelaram sua assinatura com um provedor de serviços de IoT. Se esses usuários não forem desligados da rede, eles poderão continuar a se comunicar com outros pares na rede do endpoint IoT ou poderão acessar serviços que não devem mais estar acessíveis a ele. Isso gera um custo em nome do provedor de serviços de IoT, que deve pagar pela largura de banda, tempo de CPU e armazenamento no ecossistema de serviços.

8.11 Coleta não autorizada de metadados

A IoT moderna é projetada para unir o mundo físico ao mundo digital. Neste novo modelo, os efeitos da tecnologia são possivelmente muito mais invasivos do que no passado. Usando metadados, empresas ou terceiros podem rastrear e monitorar intencionalmente o comportamento de consumidores aleatórios ou específicos.

A análise de metadados é usada quando a comunicação entre duas entidades de rede é criptografada, mas as estruturas de protocolo que identificam o tipo de mensagem ou a identidade do remetente e/ou do receptor são expostas. Esses metadados podem ser usados para influenciar.

Considere o cenário em que os automóveis emitem mensagens contendo metadados que são atribuíveis a um consumidor específico. Qualquer pessoa com a capacidade de rastrear (local ou remotamente) esses metadados pode monitorar o comportamento do consumidor e influenciá-lo ou em suas intenções. Se houver falhas de segurança que possam ser exploradas no sistema de telemática do veículo, talvez seja possível rastrear e segmentar o sistema de um consumidor específico, colocando-o em risco de dano físico.

Organizações legais e seguradoras estão preocupadas sobre como esses riscos afetarão o futuro do financiamento automotivo, e estão começando a se envolver em leis e normas que determinarão como os engenheiros devem projetar equipamentos de telemática. Essa mudança acabará por chegar aos setores menos ativos da IoT, à medida que mais tecnologia for desenvolvida.

Para combater a coleta de metadados, criptografe o máximo de dados possível e use identificadores binários exclusivos para módulos de comunicação. Aplique uma política que impeça que usuários externos consigam usar a API do sistema IoT para derivar números de série de hardware e outras identidades rastreáveis de perfis de usuários. Sempre que possível, não permita que a estrutura de uma mensagem seja exposta a terceiros. Não permita que ações, atividades ou comportamentos sejam expostos a terceiros. Reforce a confidencialidade e integridade de todos os dados relacionados à privacidade do usuário.

8.11.1 Risco

O uso de segurança de comunicação fraca pode permitir a coleta de dados ou metadados que coloque em risco o usuário final ou exponha a privacidade do usuário final. Como as empresas de seguros estão criando um caso para aplicar os requisitos de privacidade do usuário final à tecnologia, e o negócio pode se colocar em risco se não assumir a responsabilidade pelos dados gerados por seus dispositivos.

9 Recomendações de baixa prioridade

Recomendações de baixa prioridade abrangem o conjunto de recomendações que se aplicam a riscos que são extremamente caros para combater ou que provavelmente não afetam o design do endpoint. Embora essas recomendações sejam valiosas e as informações detalhadas nas recomendações sejam importantes, as estratégias de mitigação ou remediação discutidas podem estar fora do escopo em relação aos negócios. Avalie cada recomendação e determine se os riscos descritos são relevantes ou importantes para o negócio e seus clientes. Se os clientes exigirem que esses riscos sejam abordados, aplique as recomendações.

9.1 Negação intencional ou não intencional de serviço

Para as comunicações de rádio, há uma ameaça constante de interferência, ou a transmissão intencional de ruído ou padrões que podem ser usados para embaralhar sinais legítimos. Como os sinais de rádio são simplesmente compostos de elétrons trafegando pelo ar em um padrão específico, é bastante fácil inventar uma série de sinais que interrompam ou manipulem o padrão que molda os dados de comunicação.

Normalmente, o objetivo de tal ataque é uma interrupção simples, para proibir ou negar o serviço a usuários legítimos. Em outros casos, o abuso pode ser mais ostensivo. Por exemplo, os protocolos de comunicação que não possuem mecanismo de autenticação podem ser falsificados. Para conseguir isso, o sinal real deve estar congestionado para que o sinal falsificado do hacker tenha maior probabilidade de atingir o alvo pretendido.

Um exemplo disso é a falsificação de sistemas de posicionamento global (GPS). Sinais de GPS civis não possuem criptografia e autenticação, pois são, essencialmente, um sinal de radiodifusão de texto simples que qualquer um pode receber. Também é um sinal de rádio relativamente fraco e é facilmente atenuado por anomalias ambientais, como pré-amplificadores de Ultra Alta Frequência (UHF) para receptores de televisão e microondas.

Para dispositivos que exigem que as informações de localização funcionem corretamente, um sinal de GPS congestionado pode resultar em um risco de confiabilidade que pode se transformar em um risco de segurança da informação, especialmente se a falsificação for posteriormente utilizada.

Para combater o congestionamento e outras formas de ataques de negação de serviço (DoS) intencionais, desenvolva um protocolo de comunicação robusto que se concentre em métodos para invalidar as quebras de serviço. A rede deve detectar se os dispositivos se desligaram, repentina ou anormalmente, da rede. Cada endpoint deve "dar um adeus" quando pretende sair da rede. Se isso não acontecer, a anomalia deve ser anotada para análise estatística.

Além disso, as chaves de segurança de comunicação devem ser renegociadas toda vez que um dispositivo ingressar novamente na rede. A mesma chave de segurança de comunicações não deve ser repetida. Ela deve ser inicializada pela mesma chave criptográfica assimétrica, mas qualquer chave simétrica derivada da negociação de chave deve ser nova para cada sessão de comunicação.

A interferência não intencional pode ocorrer em um rádio por vários motivos: condições ambientais que não permitem a propagação de sinal, equipamentos com defeito ou mesmo

equipamentos adjacentes operando na mesma frequência. Independentemente do motivo, os engenheiros que dependem de comunicações de rádio esperam que haja condições temporais que causem degradação ou perda de sinal. Essas perdas devem ser compensadas através do projeto da aplicação e do protocolo de comunicação de rede.

Recomenda-se que os desenvolvedores leiam as Diretrizes de Eficiência de Conexão da GSMA [9], que contêm orientações sobre como se proteger contra ataques de negação de serviço não intencionais e fornecem orientações sobre o DHIR (Relatório de identidade do host do dispositivo).

9.1.1 Risco

Não combater o risco de DoS intencional resultará em comportamento anormal ou inseguro do endpoint. Se ele sempre usar a mesma chave de sessão, isso pode ser uma maneira de invasores se aproveitarem da arquitetura de rede para coletar informações sobre a chave simétrica usada para proteger as comunicações. A construção adequada de uma sessão segura após cada sessão desconectada é imperativa para a segurança das comunicações do endpoint.

9.2 Análise crítica de segurança

A maioria dos produtos da Internet das Coisas irá incorporar algum aspecto do mundo físico com a tecnologia digital. Como resultado, é provável que um humano tome uma decisão no mundo físico com base em informações fornecidas por um endpoint IoT. Como alternativa, um endpoint IoT poderá tomar uma decisão que afeta o mundo físico com informações obtidas no mundo digital.

Portanto, é imperativo que os provedores de serviços de IoT avaliem seus produtos de uma perspectiva de segurança para determinar se, como e quando a vida humana pode ser afetada pela tecnologia. Se salvaguardas adequadas não forem colocadas em prática para garantir que a tecnologia não possa ser usada a fim de causar danos físicos, seus clientes podem ser colocados em risco.

Para ajudar a resolver questões da segurança, converse com as equipes executivas, jurídicas e de seguros do provedor de serviços de IoT. Certifique-se de que essas equipes entendam os recursos e as limitações da tecnologia usada no produto ou serviço. Determine se essas tecnologias podem atender às necessidades do negócio e oferecer aos clientes o nível de segurança necessário para a aplicação pretendida.

9.2.1 Risco

Claramente, o resultado de não ter tempo para avaliar o impacto do produto ou serviço na segurança dos clientes pode resultar em perda de receita, acidentes inesperados ou até mesmo perda de vidas.

9.3 Erradique componentes falsificados ou bridges não confiáveis

Componentes no circuito físico normalmente não usam qualquer imagem de confidencialidade e integridade ao se comunicar uns com os outros ou com a unidade central de processamento. Como resultado, qualquer invasor pode ler ou gravar dados transmitidos nesses barramentos. O efeito dessa brecha na segurança das comunicações é a possibilidade de um invasor falsificar os dispositivos legítimos no circuito físico. Se o

invasor puder escolher, ele poderá personificar um componente crítico, como NVRAM, RAM ou até mesmo uma âncora de confiança.

O objetivo desse ataque seria ignorar a segurança empregada entre dois componentes no barramento. Um exemplo típico desse cenário é utilizar esse ponto fraco para ignorar o processo de validação de integridade da análise de uma imagem da aplicação armazenada na NVRAM. Quando a CPU recupera a memória armazenada na NVRAM, o hacker pode usar um sistema de passagem para fornecer o conteúdo real da memória à CPU. Quando a aplicação em execução na CPU verifica a integridade da imagem da aplicação, o hacker pode então instrumentar as comunicações no barramento físico para trocar seletivamente os conteúdos da NVRAM que são benéficos para o invasor. Em outras palavras, a CPU verifica uma imagem da aplicação (a imagem original), mas, em seguida, carrega a imagem do hacker na RAM e a executa.

Uma maneira de se proteger contra esse ataque é:

- Carregar o conteúdo da NVRAM para a RAM
- Validar a cópia da aplicação carregada na RAM
- Executar diretamente na RAM, ou no cache, o conteúdo da RAM

Também deve ser relatado que um invasor poderia subverter a RAM também, enfraquecendo este processo. No entanto, executar um ataque man-in-the-middle contra RAM é muito mais complexo e caro do que um ataque contra NVRAM porque a velocidade dos padrões de canais e acesso é muito mais rápida e errática do que com a NVRAM, que é acessada principalmente em blocos.

Como alternativa, o invasor pode criar checksums para regiões menores de conteúdo válido na NVRAM e verificar periodicamente as assinaturas nela. Se os checksums forem diferentes, o conteúdo está sendo manipulado. Isso pode ter sucesso, mas tem uma possibilidade de sucesso menor porque o invasor pode manipular apenas uma pequena quantidade de dados que não são verificados aleatoriamente pela aplicação em execução.

Deve-se notar que, embora a melhor maneira de se proteger contra esse ataque seja validar o conteúdo da NVRAM e carregá-lo na RAM executável, não há solução completa para esse problema. O custo de proteger componentes físicos é tão alto que não é prático resolver esse ataque de maneira mais completa, a menos que o cliente exija essas garantias de segurança.

Esse ataque é ainda mais simplista quando um protocolo de comunicação física mais básico, como o I2C, é usado. Canais como o I2C são essencialmente sistemas de transmissão físicos. Assim, qualquer componente assentado no barramento I2C pode fingir ser qualquer outro componente. Isso permitiria que um invasor personifique outros dispositivos no canal que não reforcem a confidencialidade e a integridade no canal de comunicação. Onde isso for uma preocupação, imponha confidencialidade e integridade ao protocolo de aplicação usado sobre o protocolo de barramento físico.

9.3.1 Risco

O risco de não implementar uma solução resultará na possibilidade de um invasor ignorar as verificações de integridade na aplicação. Isso permitirá que o invasor comprometa a aplicação que está sendo executada por um código com maior privilégio, como inicializadores ou TCBs.

Deve ser notado, no entanto, que este ataque é muito menos provável do que ataques mais simples contra o inicializador. Realizar um ataque de hardware man-in-the-middle contra componentes como NVRAM, ou componentes de alta velocidade como RAM, é desafiador, complexo e caro atualmente. Embora sempre seja possível para um invasor subverter um sistema embutido dessa maneira, pode ser muito dispendioso fazer isso.

Portanto, carregar o código na RAM e verificar a integridade pode ser uma solução razoável que ignorará a maioria dos ataques, se houver.

Além disso, pelas razões descritas acima e algumas outras, as chaves criptográficas não devem ser mantidas com privilégios inseguros como esses. Elas devem ser armazenados em uma âncora de confiança e usadas pela TCB, não armazenada em mídia, como NVRAM, que possam ser representadas ou comprometidas.

9.4 Derrote um ataque de inicialização

Um ataque à inicialização [REFERÊNCIA] é uma estratégia de ataque físico contra sistemas de computador que rouba seus segredos em execução removendo a memória física do computador e a transporta para um sistema secundário controlado pelo invasor. O objetivo desse ataque é que o invasor pode executar um sistema operacional customizado que despeja o conteúdo da RAM em um armazenamento permanente. Isso permitirá que o hacker vasculhe os dados recuperados e determine se há tokens relacionados à segurança que podem ser usados. Isso pode incluir:

- Senhas criptografadas ou chaves privadas
- Credenciais de login (nomes de usuários e senhas)
- Informações de identificação pessoal (PII, da sigla em inglês)
- Acesso a tokens de serviços web

O objetivo do ataque é comprometer os segredos que permitem ao hacker obter acesso de longo prazo a um recurso que estaria fora de seu alcance. Por exemplo, quebrar os algoritmos criptográficos usados no padrão mais recente do TLS seria impossível para o hacker mediano. No entanto, comprometer o certificado de cliente privado usado em um serviço TLS de autenticação mútua permitiria que o hacker simulasse o cliente de um sistema mais conveniente.

Para ter sucesso neste ataque, o hacker deve ser capaz de remover a RAM do sistema de computador sem os bits armazenados na troca de chip. Conforme detalhado no trabalho de pesquisa, isso pode ser feito resfriando os chips de memória. No entanto, a RAM deve ser facilmente removível. Se a RAM estiver soldada na placa de circuito, isso complicaria enormemente o ataque e exigiria que o hacker usasse uma pistola de solda para extrair memória, possivelmente danificando seu conteúdo.

É importante observar que a limpeza da memória no desligamento é sempre útil, e é aconselhável, para aumentar a privacidade de um endpoint. No entanto, um ataque à inicialização pode ocorrer a qualquer momento, mesmo enquanto o sistema está em execução. Portanto, a limpeza da memória pode ser útil, mas não garante a derrota de um ataque do mundo real.

Um combate mais eficaz para esse ataque é processar ações centradas na segurança usando a RAM interna da CPU. Muitas CPUs, MCUs e MPUs possuem uma pequena quantidade de SRAM interna que pode ser usada por uma aplicação em execução. Se a aplicação limitar o uso de tokens de segurança críticos (como chaves privadas) a essa RAM interna, o conteúdo da RAM removível (ou externa) terá menos valor para um invasor.

9.4.1 Risco

Desconsiderar o risco de um ataque à inicialização pode fazer com que chaves de segurança críticas sejam roubadas usando um modelo de ataque simples. Se as chaves de segurança forem universais para todos os endpoints no ecossistema do provedor de serviços de IoT, um grande comprometimento poderá ser plausível.

Para mais informações consulte: https://citp.princeton.edu/research/memory/

9.5 Riscos não óbvios à segurança (“vendo através das paredes”)

Apesar de permitir e impor a autenticação mútua, a confidencialidade e a integridade na rede de comunicações, os padrões de tráfego podem relacionar-se diretamente a eventos. Quando os dados são trafegados em resposta a determinados eventos físicos, uma correlação pode eventualmente ser feita entre eventos físicos e dados. Isso pode permitir que um invasor monitore os padrões de sinal e, em seguida, obtenha o significado destes padrões, independentemente dele ter ou não acesso direto aos dados de texto puro.

Um exemplo disso é a tecnologia de automação residencial que reage com base na presença física de um usuário em um ambiente específico. Um invasor capaz de monitorar remotamente o sistema de comunicações pode ser capaz de observar quantos usuários estão em uma determinada casa, onde as pessoas se situam na casa e quem é o usuário, observando apenas padrões de comunicação entre endpoints, gateways e sistemas back-end.

O invasor pode ser capaz de diferenciar facilmente entre uma casa com várias pessoas e apenas um indivíduo, e onde esse indivíduo está dentro dela. Empresas de seguros e entidades legais precisarão entender como isso possivelmente aumenta o risco para os proprietários e outros inquilinos no espaço de convivência.

Combater esse risco pode ser difícil. O modelo mais comum e mais simples para fazer isso é enviar amostras com uma taxa pré-definida, independentemente de haver um usuário presente para retirar amostras. Se a confidencialidade e a integridade forem aplicadas, impedindo que invasores remotos avaliem o texto simples dos dados, um observador não conseguirá diferenciar entre uma amostra contendo atividades do usuário e uma amostra vazia.

No entanto, existem observações sobre esse modelo, como o aumento da saturação de espectro, aumento do consumo de energia para tecnologia de baixa potência, ou habilitada

por bateria, e aumento do nível de processamento necessário para descriptografar, verificar e interpretar os pacotes de amostra vazios.

Uma alternativa é enviar amostras em intervalos aleatórios, com disparos variados. Esse tipo de padrão é menos dispendioso, menos consumidor de energia e requer menos poder de processamento. Ainda assim, ainda é possível observar mudanças sutis que indicam a presença do invasor. Por exemplo, qualquer sistema verdadeiramente entrópico é totalmente aleatório e imprevisível. O comportamento do usuário, no entanto, é totalmente previsível. Se um usuário entrar em uma sala e os sensores nessa sala reagirem e começarem a enviar dados para endpoints da rede, a introdução de um comportamento repetitivo pode confirmar sua presença.

Qualquer equipe que desenvolve tecnologia sujeita a esse tipo de risco deve investigar os possíveis efeitos da exposição da privacidade e consultar a equipe jurídica para determinar se a tecnologia afetaria a postura legal ou o modelo de seguro da empresa.

9.5.1 Risco

Se o provedor de serviços de IoT não avaliar sua tecnologia na perspectiva de possíveis exposições à privacidade e riscos de segurança, talvez seja necessário reformular substancialmente a arquitetura a fim de compensar os riscos que devem ser enfrentados. Em vez de tentar fazer ajustes dispendiosos para a arquitetura em um momento posterior, projete essas soluções no produto no início da fase de engenharia ou o mais cedo possível.

9.6 Combata feixes concentrados de íons e raios X

Um Feixe Concentrado de Íons (FIB, da sigla em inglês) é um instrumento de fabricação comumente usado na avaliação de semicondutores. A tecnologia é capaz de inspecionar e alterar circuitos no nível nanométrico, o que permite aos analistas identificar falhas de fabricação e testar as atualizações do circuito antes de alterar o processo de fabricação.

Em segurança de informação, um FIB pode ser usado para acessar os canais de comunicação com a finalidade de interceptar dados trafegados sobre componentes internos. Além disso, um FIB pode ser usado para alterar os circuitos internos, o que altera a forma como o componente interno irá operar, permitindo que um invasor contorne uma restrição de segurança.

Quase todos os dispositivos estão sujeitos a ataques por um FIB. No entanto, apenas determinados dispositivos serão executados por meio de um processo FIB. Isso ocorre porque o próprio FIB é uma tecnologia extremamente cara, de aproximadamente 1 milhão de dólares por unidade. Devido ao alto custo da tecnologia, poucas organizações têm esse dispositivo em seu kit de ferramentas. Além disso, o dispositivo não é automatizado. Isso requer um alto grau de habilidade para manipular, bem como um alto grau de especialização em análise de semicondutores, para ser utilizável. Assim, o custo real de um FIB está muito além de um simples valor de um milhão de dólares, e se estende aos milhões de dólares para a própria concessionária e para a educação, o salário e a experiência do usuário.

As organizações estão disponíveis para terceirização, no entanto. Como a engenharia reversa é amplamente legal, as organizações fornecem serviços de ataque a semicondutores para clientes interessados em engenharia reversa de um dispositivo. Esses

contratos custam entre 10 mil e 1 milhão de dólares, dependendo do nível de customização e conhecimento necessários para atacar um determinado componente. Por exemplo, uma empresa terceirizada teria um manual para contornar as proteções em um chip comum. Mas, uma solução FPGA customizada com tecnologia inovadora de bloqueio de segurança custaria muito mais, já que nenhum manual existente seria adotado. Um novo processo seria necessário para usar o FIB com sucesso, custando tempo e dinheiro substanciais.

Algumas novas tecnologias, como modernas variantes de âncoras de confiança, alegam resistência às sondas FIB. Embora haja alguma validade para essas alegações, qualquer proteção de hardware que não seja dinâmica (e a maioria não é) resultará em um *manual* após um período suficiente para analisar as técnicas de desvio. Portanto, essas novas afirmações podem ser válidas, mas só podem ser válidas por um intervalo de tempo.

Portanto, para compensar tecnologias invasivas, mas quase sempre bem-sucedidas, como essas, é essencial que a organização de engenharia projete uma estratégia de segurança que seu sucesso não se atrele apenas à âncora de confiança. Em vez disso, um protocolo eficiente deve ser desenvolvido para usar a tecnologia como uma âncora de confiança de base, mas que customize as chaves criptográficas de cada endpoint de modo que nenhum comprometimento de um único dispositivo resulte em um comprometimento de toda a rede de endpoints.

Considere o cenário em que um invasor deve usar um FIB para extrair criptografia de cada endpoint que deseja segmentar. Isso rapidamente se tornaria uma proposta extremamente custosa e estaria fora do objetivo em relação ao orçamento de quase todos os invasores. Uma vez que estas metodologias de ataque não podem ser suficientemente mitigadas, elas devem ser desprezadas, para diminuir o risco através da arquitetura e não através da obscuridade.

9.6.1 Risco

O risco de um FIB é que os segredos criptográficos e outras propriedades intelectuais podem ser roubados de um componente, até mesmo de um com segurança reforçada. Uma vez que derrotar um FIB de uma maneira custo-benefício para o consumidor IoT é impraticável, a organização deve alterar sua estratégia para proteger os sistemas endpoint ou arriscar um comprometimento completo do ecossistema.

9.7 Analise a segurança da cadeia e fornecedores

A segurança de qualquer sistema de computação começa com os componentes dos quais a placa de circuito é composta. O silício, os tokens criptográficos, a memória de leitura (ROM), o firmware e outros atributos centrais de um sistema embutido contribuem para a segurança de tal sistema. Se qualquer um desses componentes for adulterado, todo o sistema poderá estar sujeito a um comprometimento de segurança.

Como resultado, os provedores de serviços de IoT que estão conscientes sobre a segurança devem levar em consideração a origem de seus componentes, sua montagem e o processo de fabricação usado para entregar a tecnologia embarcada. Se o processo usado para gerar a tecnologia não for planejado com cuidado, um único ponto de falha no processo pode resultar em uma falha crítica de segurança.

Considere os seguintes questões:

- Onde e por quem foi fabricado o chip?
- O design do chip foi analisado por uma equipe independente e confiável de segurança da informação?
- O chip será fabricado em uma instalação segura?
- Como a EEPROM ou NVRAM irá carregar uma cópia executável, como um gerenciador de inicialização?
- O processo de inicialização da cópia executável é seguro?
- Como a cópia do executável será entregue ao fabricante?
- A cópia executável é verificada depois de ter sido enviada para a EEPROM ou a NVRAM?
- Como os segredos criptográficos são armazenados nos chips?
- Se os segredos são gerados no fabricante, eles estão usando RNG certificado para gerar as chaves?
- Todas as chaves de segurança são exclusivas de acordo com as recomendações da TCB?
- Como os segredos criptográficos são compartilhados com o provedor de serviços de IoT? Com segurança?
- Como os identificadores exclusivos de chip (número de série etc.) estão correlacionados aos segredos criptográficos e compartilhados com o provedor de serviços de IoT?

Embora a escolha de uma instalação segura para construir e montar um produto possa gerar um investimento maior, pode ser um passo decisivo para a organização. Isso depende do caso de uso do produto, do ambiente de implementação pretendido, do cliente pretendido e de outros fatores, como segurança humana, aplicações militares e implantações de infraestrutura crítica. Onde a vida humana pode ser impactada pela tecnologia resultante, a cadeia de fornecimento deve ser avaliada quanto a falhas na segurança.

9.7.1 Risco

Sem a segurança da cadeia de fornecedores, a organização está sujeita a muitos riscos, alguns dos quais podem ser totalmente inesperados e, no entanto, críticos para os negócios:

- Clonagem de endpoint (manufatura ilegal)
- Roubo de tecnologia (concorrentes roubando e subfaturando o provedor de serviços)
- Roubo de credencial (interceptação de dados ou ataques à identidade)

- Injeção de implantes (“back doors” maliciosos que podem ser ativados em um momento

9.8 Interceptação legal

Interceptação legal é o ato de interceptar ou manipular legalmente as comunicações entre um cliente e um provedor de serviços. Isso pode funcionar de duas maneiras. Primeiro, o cenário mais típico é que um órgão competente possa enviar uma solicitação a uma operadora para acesso a metadados ou dados reais de comunicações requisitadas por um usuário específico. Em segundo lugar, o órgão competente solicitará ao provedor de serviços de IoT o acesso a dados e/ou metadados pessoais desse usuário específico. No cenário em que o órgão competente solicita acesso por meio da operadora, o provedor de serviços de IoT talvez nunca seja notificado de que há um problema, dependendo do objetivo da solicitação legal. Assim, o provedor de serviços deve estar pronto para implementar ou cumprir uma solicitação legal feita por tal órgão.

Portanto, o provedor deve identificar quais desafios para a privacidade podem resultar de uma ordem judicial e deve estar pronto para fornecer informações relevantes sobre seu modelo jurídico e a política de privacidade da organização, dentro de sua respectiva responsabilidade legal.

No passado recente, empresas como Google, Apple e outras gigantes adotaram sistemas de notificação (*warrant canaries*) para informar legalmente aos usuários quando uma solicitação sigilosa foi feita à empresa por órgão competente. A empresa pode remover uma frase, imagem ou outro conteúdo que simboliza *não* estar em contato com órgãos competentes. A remoção deste objeto é indicativa, é claro, de uma solicitação requerida.

9.8.1 Risco

Não estar preparada para uma solicitação de impedimento legal coloca a empresa em desvantagem se tal requisito for apresentado a ela. A empresa pode precisar atender à solicitação, mas pode não ter a infraestrutura legal ou as políticas de privacidade devidas, possivelmente as colocando em risco.

Não preparar o protocolo do endpoint e da plataforma IoT para confidencialidade e integridade adequadas permitirá que as comunicações sejam interceptadas na rede sem o conhecimento da empresa. Isso pode colocá-la em risco de vazamento dos dados do usuário ou associado, como no caso dos vazamentos de Snowden (NSA), diminuindo substancialmente a confiança do público na credibilidade da organização em proteger os dados do usuário.

10 Resumo

Em resumo, quase todos os riscos de segurança em um produto ou serviço de IoT podem ser combatidos por uma arquitetura bem definida, inteligência para identificar ameaças antes e durante eventos relacionados à segurança, políticas e procedimentos para lidar com esses eventos. Ao analisar quais conceitos de segurança de alto nível são importantes para o provedor de serviços de IoT, as perguntas frequentes podem ser revisadas. Isso deve orientar a equipe de engenharia em relação às recomendações mais relevantes para resolver as falhas em sua arquitetura de segurança.

À medida que a equipe avança em sua definição de arquitetura, ela pode revisar recomendações individuais conforme seus desafios de segurança, e suas preocupações se tornam mais exclusivas em sua própria implementação.

De modo geral, toda equipe de engenharia enfrentará riscos muito semelhantes. É imperativo que a organização opte por compartilhar suas preocupações com suas parcerias para construir uma base de conhecimento comum para os riscos e as estratégias de remediação. Juntas, as organizações podem construir tecnologia e conhecimento para ajudar umas às outras na criação de segurança para o futuro da IoT.

Annex A Exemplo usando uma arquitetura genérica de inicialização

O nível de segurança de uma grande rede de serviços como um todo é definido pelo elo mais fraco da cadeia. Assim, o link local entre o endpoint IoT e um gateway precisa ser protegido com um nível de segurança comparável ao da rede de banda larga para manter o mesmo nível geral de segurança.

Uma tecnologia candidata para isso é a Generic Bootstrap Architecture (GBA) [17], que pode ser usada tanto para autenticação quanto para integridade de dados. Ela é baseada em chaves pré-compartilhadas, que são usadas para gerar chaves com tempo limitado (tokens) como base de autenticação e criptografia.

Autenticação é o processo de determinar se alguém ou algo é, de fato, quem se declara. No espaço da IoT, onde bilhões de endpoints estarão ativos, determinar qual comportamento de comunicação é genuíno e confiável é primordial. O mecanismo estabelecido para criar essa relação de confiança precisa satisfazer o requisito de ser escalar e sustentável.

Além disso, a variedade de serviços de IoT impõe a exigência de que o mecanismo de autenticação possa ser adaptável para acomodar esses diferentes serviços e ainda manter uma infraestrutura comum. Um mecanismo que se comprovou ao longo do tempo é a autenticação de rede baseada no SIM. Essa infraestrutura de autenticação tem a virtude de fornecer não apenas autenticação, mas também recursos de criptografia baseados em chaves pré-compartilhadas.

A explosão no número de endpoints e o alcance global da IoT torna o uso de SIM limitado por causa do roaming de rede e a fragilidade da segurança de poder remover fisicamente um SIM de um endpoint desativado. A chegada de tecnologias como o embedded SIM fornece uma infraestrutura prática para autenticação baseada em chaves pré-compartilhadas, estendendo a atual autenticação de rede baseada em SIM. Além disso, o crescimento da IoT é mais provável de acontecer na forma de redes capilares (o PAN, conforme mostrado nas configurações de exemplo 2, 3 e 4 na seção anterior deste documento).

Essas redes capilares são colméias de dispositivos endpoints conectados a um gateway. A maioria desses dispositivos são endpoints compactos (ou seja, eles não contêm conectividade SIM nem celular). Esses endpoints compactos, no entanto, exigem recursos de autenticação e criptografia. Em redes capilares, a principal responsabilidade da autenticação está no gateway, reduzindo o número de dispositivos endpoints complexos baseados em SIM na rede como um todo. Essa autenticação e segurança devem ser estendidas do gateway para o endpoint, criando um canal seguro a partir do endpoint fornecido para a plataforma de serviços de IoT.

A autenticação baseada em SIM destina-se a atender a uma única aplicação, ou seja, a autenticação de um endpoint exclusivo para a conexão de rede. Os endpoints terão uma infinidade de serviços, cada um com necessidade diferente e exclusiva de autenticação. Uma estrutura que amplia a autenticação de rede para vários serviços é necessária. Uma estrutura projetada para essa finalidade é o GBA, Generic Bootstrapping Architecture. O GBA aproveita a infraestrutura baseada em SIM para gerar chaves de compartilhamento

temporizadas entre dispositivos e as Network Application Functions (NAFs). O GBA é um método de autenticação padronizado pelo 3GPP na especificação 3GPP TS 33.220 [17].

O método permite a autenticação de um dispositivo com uma assinatura 3GPP para um serviço. As credenciais da assinatura estão no dispositivo, geralmente armazenadas em um SIM card, como UICC (Universal Integrated Circuit Card) ou como credenciais gerenciadas remotamente, armazenadas e gerenciadas em um embedded SIM (eUICC), por exemplo, como especificado no Embedded SIM da GSMA(eUICC) [5].

As vantagens deste framework são:

- Autenticação mútua baseada em um PSK exclusivo entre um dispositivo e uma Network Application Function ou por autenticação UE baseada em chave compartilhada com autenticação NAF, baseada em certificado (TS 33.222) [18].
- Credenciais podem ser protegidas em um ambiente confiável
- Se eUICC for usado, as credenciais podem ser alteradas over-the-air
- Escalabilidade. A complexidade e o custo econômico da manutenção aumentam linearmente com o número de dispositivos, já que a autenticação é “embarcada” no framework.
- Integridade de dados. As chaves geradas por base de tempo durante a autenticação podem ser usadas para estabelecer túneis TLS-PSK, fazendo com que essa conexão forneça integridade e confidencialidade de dados muito fortes.

Annex B Tutorial sobre o uso de cartões UICC em serviços de IoT

O UICC, conforme padronizado no ETSI TS 102 221, é uma plataforma de cartão inteligente (um elemento seguro resistente a violações programáveis) que fornece uma interface de sistema de arquivos segura e interoperável e uma estrutura de aplicações segura para dispositivos de hospedagem UICC. ETSI TS 102 221 fornece uma estrutura para um dispositivo de hospedagem UICC para descobrir aplicações relevantes em um UICC, e cada aplicação UICC corresponde a um conjunto conhecido de informações de configuração e fornecimento, bem como procedimentos operacionais (como autenticação ou derivação de chave) que podem ter suporte pelo dispositivo de hospedagem de acordo com suas necessidades.

No contexto da IoT, o UICC pode estar disponível em vários fatores de formato e intervalos operacionais ambientais, conforme especificado na ETSI TS 102 671. Em sua forma de realização mais simples, o UICC é normalmente de propriedade de um operador de rede e hospeda apenas uma aplicação de acesso à rede (3GPP TS 51.011, USIM conforme 3GPP TS 31.102, CDMA CSIM, como especificado por 3GPP2, WiMAX SIM etc.).

Nesse caso, o UICC fornece um suporte padronizado para hospedar informações de configuração e configuração de segurança, bem como procedimentos criptográficos em um dispositivo móvel para permitir acesso à rede, com mecanismos adicionais para gerenciar remotamente o conteúdo do UICC, usando ETSI TS 102 225 / TS 102 226. O ecossistema de rede móvel tem procedimentos implementados para garantir a customização e a implantação seguras de UICCs sob o controle da operadora de rede, resultando no estabelecimento de chaves simétricas compartilhadas individuais entre os dispositivos de hospedagem UICC e a infraestrutura.

Uma característica importante da plataforma UICC é o suporte de domínios de segurança isolados, que permitem que cada usuário de um ecossistema complexo seja atribuído a cada um em sua própria área em um UICC e gerencie seu conteúdo em confidencialidade com outros interessados. Essa funcionalidade é herdada por meio de ETSI TS 102 226 da Emenda de Especificação de Cartão Global Platform [15]. Portanto, em um contexto da IoT, um único UICC permite que várias partes interessadas armazenem e administrem suas próprias credenciais independentes das outras.

De modo geral, um UICC pode conter várias aplicações de acesso à rede (com apenas um ativo em um determinado momento) e possivelmente outras aplicações protegendo o acesso a serviços mais elaborados, como aplicações ISIM para acesso IMS (conforme especificado em 3GPP TS 31.103) ou, no caso de serviços de IoT, aplicações 1M2M SM especificadas no Anexo D da oneM2M TS-0003. Uma aplicação 1M2MSM pode dar suporte ao fornecimento direto de credenciais de serviços de IoT/aplicações dedicadas, bem como a derivação de credenciais de acesso à rede preexistentes no UICC usando um mecanismo GBA especificado pelo 3GPP. Além disso, permite que um provedor de serviços de IoT customize os procedimentos criptográficos de acordo com suas necessidades específicas, por exemplo, para dar suporte a mecanismos específicos de autenticação do serviço.

Um único UICC também pode conter várias aplicações 1M2MSM, permitindo a instalação confidencial de chaves simétricas dedicadas a cada provedor de serviços de IoT. Uma concessionária UICC (normalmente uma operadora de rede ou fabricante OEM para a IoT) pode compartilhar espaço em seu UICC com provedores de serviços de IoT que solicitem,

para que a cadeia de customização e credenciadas da infraestrutura UICC permitam a implantação segura de credenciais de acesso à rede também possam ser aproveitadas por provedores de serviços de IoT para implantar suas próprias credenciais.

Onde a segurança da aplicação de IoT depende de criptografia assimétrica, as aplicações UICC customizadas podem ser usadas de forma semelhante para facilitar a implantação de pares de chaves pública/privada, conforme necessário para um serviço de IoT específico. Essas aplicações UICC precisam ser especificadas e darem suporte a dispositivos de hospedagem em uma base específica da aplicação de IoT.

Annex C Gerência do documento

C.1 Histórico do documento

Versão	Data	Breve Descrição de Alterações	Autoridade de Aprovação	Editor / Companhia
1.0	08-Feb-2016	New PRD CLP.13	PSMC	Ian Smith GSMA & Don A. Bailey Lab Mouse Security
1.1	07-Nov-2016	Referências adicionadas ao esquema da GSMA de avaliação sobre segurança para IoT. Pequenos esclarecimentos editoriais.	PSMC	Ian Smith GSMA
2.0	29-Sep-2017	Referências incluídas nos recursos de rede LPWA da GSMA, além de atualizações secundárias adicionais.	IoT Security Group	Rob Childs GSMA

C.2 Outras informações

Tipo	Descrição
Proprietário do documento	GSMA IoT Programme
Contato	Rob Childs – GSMA

É nossa intenção fornecer um produto de qualidade para seu uso. Se você encontrar algum erro ou omissão, entre em contato conosco com seus comentários. Você pode nos notificar em prd@gsma.com

Seus comentários ou sugestões e perguntas são sempre bem-vindos.