Diretrizes de segurança em IoT para o Ecossistema de Serviços de IoT

Diretrizes de segurança em IoT para o Ecossistema de Serviços de IoT

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

Aviso de direitos autorais

Aviso legal

Aviso antitruste

Sumário

1 Introdução

1.1 Introdução ao conjunto de documentos de diretrizes de segurança para IoT da GSMA

Este documento é 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 o documento de visão geral 'CLP.11 IoT Security Guidelines Overview Document' [1] seja lido como base antes da leitura dos demais documentos.

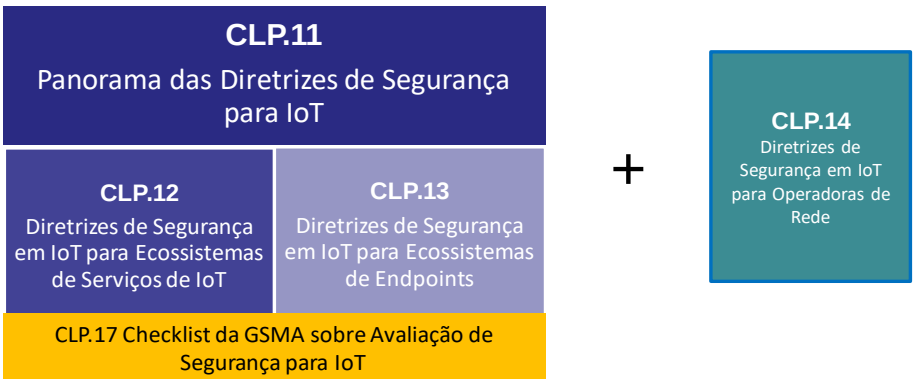

Figura 1 - Estrutura do documento da GSMA “Diretrizes de segurança para IoT”

As operadoras de rede, os provedores de serviços de IoT e outros parceiros no ecossistema de IoT são aconselhados a ler o documento CLP.14 da GSMA "Diretrizes de segurança em IoT para operadoras de rede" [13], 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 de IoT

Um checklist para avaliação é fornecido no documento CLP.17 [16]. Este documento permite aos fornecedores de produtos, serviços e componentes de IoT auto-avaliar a conformidade de seus produtos, serviços e componentes com as diretrizes da GSMA de segurança para IoT.

Completar um checklist da GSMA para avaliar a segurança de IoT [16] permitirá que uma entidade demonstre as medidas de segurança que tomou 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 Propósito do documento

Este guia deve ser usado para avaliar os componentes em um produto ou serviço de IoT a partir da perspectiva do ecossistema do serviço. O ecossistema de serviço inclui todos os componentes que compõem o núcleo da infraestrutura de IoT. Os componentes deste ecossistema são, por exemplo, serviços, servidores, clusters de bancos de dados, elementos de rede e outras tecnologias usadas para direcionar os componentes internos de qualquer produto ou serviço.

O escopo deste documento está limitado às Recomendações relativas ao desenvolvimento e à implementação de serviços de IoT e, também, aos elementos de rede.

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 estes forem considerados adequadamente protegidos.

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

1.3 Público-alvo

Os principais públicos deste documento são:

- Provedores de serviços de IoT - empresas ou organizações que procuram desenvolver produtos e serviços conectados inovadores. Alguns dos muitos setores em que os provedores de serviços de IoT operam são, por exemplo, casas inteligentes, cidades inteligentes, automotivo, transporte, saúde, utilities e eletrônicos de consumo.
- Fabricantes de dispositivos IoT para consumidores finais - fabricantes de dispositivos IoT para provedores de serviços de IoT.
- Desenvolvedores de IoT que criam serviços de IoT em nome dos provedores de serviços de IoT.
- Operadoras de rede que fornecem serviços aos provedores de serviços de IoT.

1.4 Definições

Termo	Descrição
Lista de Controle de Acesso	Uma lista de permissões anexadas a um objeto computacional
Nome do ponto de acesso	Identificador de um ponto de conexão de rede ao qual um endpoint se conecta. Eles estão associados a diferentes tipos de serviços e, em muitos casos, são configurados pela operadora de rede.

Termo	Descrição
Hacker	Definido, para os propósitos deste documento, como agente de ameaças, ator de ameaças, fraudador ou fonte de outra ameaça a um serviço de IoT. Essa ameaça poderia ser oriunda de criminoso isolado, crime organizado, terrorismo, governos hostis e suas agências, espionagem industrial, grupos de hackers, ativistas políticos, hackers por hobby, pesquisadores, bem como falhas involuntárias de segurança e privacidade.
Nuvem	Uma rede de servidores remotos na internet que hospedam, armazenam, gerenciam e processam aplicações e seus dados.
Container	Uma tecnologia que permite executar vários sistemas isolados, ou containers, em um host.
Embedded UICC (eUICC)	Um UICC que suporta o aprovisionamento remoto da rede ou das assinaturas de serviço que ele autentica, conforme especificado pela GSMA.
Cliente final	Consumidor do serviço de IoT fornecido pelo provedor de serviços de IoT. É possível que o cliente final e o provedor de serviços de IoT possam ser o mesmo ator como, por exemplo, no caso de uma empresa de utilities.
Ecossistema Endpoint	Qualquer configuração de dispositivos de baixa complexidade, alta complexidade e gateways que conectam o mundo físico ao mundo digital de novas maneiras. Veja o CLP.11 [1] para mais informações.
Forward Secrecy	Uma propriedade de protocolos de comunicação seguros: considera-se que um protocolo de comunicação seguro tenha “sigilo daqui para frente” se o comprometimento de chaves de longo prazo não comprometer as chaves de sessão anteriores.
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 recursos de comunicação máquina a máquina (M2M) que lhes permitem enviar e receber dados.
Endpoint IoT	Termo genérico para um endpoint IoT de alta complexidade ou dispositivo de gateway IoT.
Serviço de IoT	Qualquer programa de computador que aproveite dados de 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. Veja CLP.11 [1] para mais informações.
Provedor de serviços de IoT	Empresas ou organizações que desejam desenvolver produtos e serviços de IoT conectados e inovadores.
Operadora de rede	Operadora e proprietária da rede de comunicação que conecta o dispositivo endpoint IoT ao ecossistema de serviços de IoT.
Raiz Organizacional da Confiança	Conjunto de políticas e procedimentos de criptografia que determinam como identidades, aplicações e comunicações podem e devem ser protegidas criptograficamente.
Grupo de Segurança	Atua como um firewall virtual que controla o tráfego de uma ou mais instâncias do servidor virtual.

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 variar 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.
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 integrada em formatos incorporados e especificados no ETSI TS 102 671.
Rede Privada Virtual	Partição segura e logicamente separada de uma rede para permitir uso dedicado por um determinado conjunto de clientes. Assim chamada porque a VPN é privada do resto da rede e, portanto, atua como uma rede virtualizada por si só.

1.5 Abreviaturas

Termo	Descrição
3GPP	3rd Generation Project Partnership
ACL	Lista de Controle de Acesso
API	Interface do Programa de Aplicações
APN	Nome do Ponto de Acesso
CERTS	Grupos de Respostas a Incidentes de Segurança em Computadores
CLP	Programa Connected Living da GSMA
DDoS	Ataque Distribuído de Negação de Serviço (Distributed Denial of Service)
GSMA	GSM Association
HSM	Módulo de Segurança de Hardware
IoT	Internet das Coisas
IP	Protocolo de Internet
SQL	Linguagem de Consulta Estruturada
TCB	Base de Computação Confiável
VM	Máquina Virtual
VPN	Rede Privada Virtual
WAF	Firewall de Aplicação Web

1.6 Referências

Ref	Nº do Documeto	Título
[1]	CLP.11	Panorama das Diretrizes de Segurança para IoT

Ref	Nº do Documeto	Título
[2]	CLP.12	Diretrizes de Segurança em IoT para Ecosssistemas de Serviços de IoT
[3]	CLP.13	Diretrizes de Segurança em IoT para Ecossistemas de Endpoints IoT
[4]	CLP.14	Diretrizes de Segurança em IoT para Operadoras de Rede
[5]	n/a	OWASP Projeto de Design de Aplicativo Seguro https://www.owasp.org
[6]	n/a	TCG Módulo de Plataforma Confiável http://www.trustedcomputinggroup.org
[7]	n/a	TCG Orientação de Segurança para IoT http://www.trustedcomputinggroup.org
[8]	n/a	OAuth 2.0 http://oauth.net/2/
[9]		OpenID Foundation http://openid.net/foundation/
[10]	n/a	GSMA Mobile Connect https://mobileconnect.io/
[11]		Especificação da Placa GlobalPlatform www.globalplatform.org/specificationscard.asp
[12]		Especificação da API Core GlobalPlatform TEE www.globalplatform.org/specificationsdevice.asp
[13]		Checklist de Avaliação de Segurança da GSMA https://www.gsma.com/iot/iot-security-assessment/
[14]		Especificações ETSI TC SmartM2M www.etsi.org
[15]		Especificações do oneM2M www.onem2m.org
[16]		Arquitetura de Autenticação Genérica (GAA); Arquitetura de Bootstrapping Genérica (GBA) www.3gpp.org

2 O modelo de serviço

Os modernos produtos e serviços de IoT exigem um ecossistema de serviços que ofereça significado, funcionalidade e valor a endpoints, parceiros e usuários. Dependendo da complexidade das aplicações disponibilizadas pela oferta de IoT, a infraestrutura pode ser vasta e composta por muitos tipos diferentes de serviços e pontos de acesso de serviço. Alternativamente, a infraestrutura pode ser rudimentar para aplicações mais diretas.

Independentemente do formato, o ecossistema de serviço atua como o nexo de funcionalidade e comunicação para cada braço da tecnologia de IoT. Todos os outros

ecossistemas dependem do ecossistema de serviços para autenticação hierárquica, conectividade com usuários, disponibilidade, gerenciamento e outras tarefas essenciais para a operação diária de IoT. Para realizar essas tarefas, o ecossistema de serviços é composto por camadas necessárias para cumprir as metas da infraestrutura. Clusters de bases de dados, servidores de aplicações, servidores proxy de aplicações e outros tipos de infraestrutura são exemplo de camadas que seriam encontradas em diferentes projetos. Conforme descrito no diagrama abaixo, os ecossistemas de rede e de endpoint dependem da funcionalidade central do ecossistema de serviços.

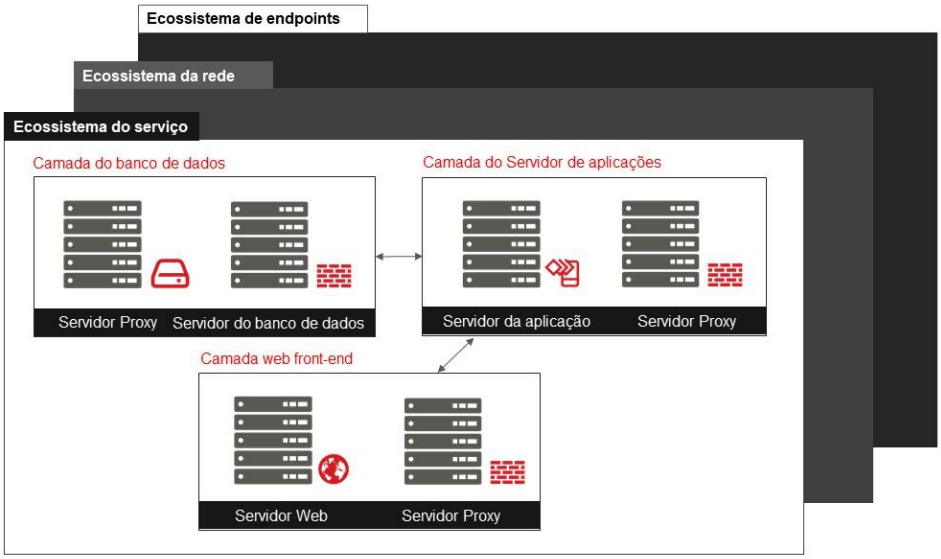

Figura 2 – Dependências vinculadas ao ecossistema de serviços

Alguns exemplos de ecossistemas de serviços modernos incluem, mas não estão limitados a:

- Soluções baseadas em infraestrutura de nuvem
- Aplicações baseadas em conteiner
- Ambientes de servidores de datacenter tradicionais
- Clusters de banco de dados
- Clusters de serviços de estrutura de aplicações Web

Embora cada um desses ambientes mencionados possa aparentemente variar consideravelmente em seu design, topologia e implementação, eles são baseados nas mesmas teorias com relação à maneira como a informação entra e sai de uma aplicação.

Todos os sistemas de computação modernos exigem um ponto de entrada, conhecido como ponto de acesso de serviço, na infraestrutura de uma aplicação. Os subsistemas internos que criam conteúdo e contexto para essa aplicação devem ser capazes de processar dados em ambientes e redes seguras e confiáveis. Os dados devem ser armazenados em algum lugar, depois devolvidos à camada de serviço que responde ou envia comandos autorizados

para vários componentes do mesmo ecossistema, ou de outros ecossistemas e suas redes associadas.

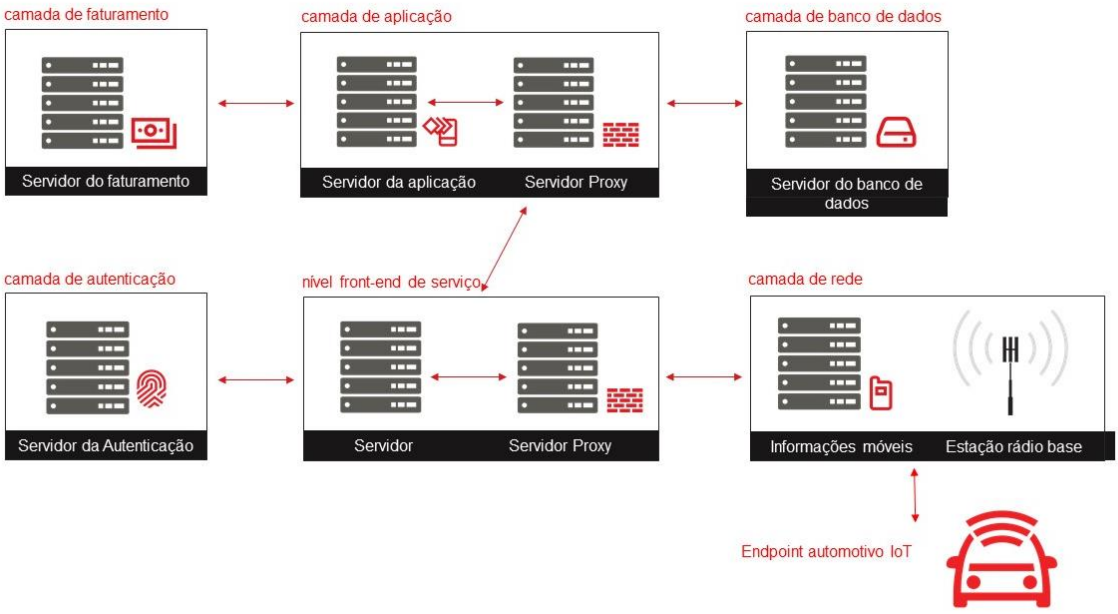

Figura 3 – Exemplo de ecossistema de serviços

Independentemente das tecnologias usadas para implementar essa estrutura padrão - sejam elas modernas ou tradicionais -, as informações serão processadas, utilizadas e autenticadas por meio de protocolos e tecnologias comprovados. Enquanto topologias e abstrações referentes a ambientes de processamento têm sido sutilmente alteradas para se adequarem aos requisitos modernos de velocidade, poder computacional e armazenamento, as tecnologias usadas para implementar essas inovações são, em sua essência, as mesmas. Por exemplo, cada camada geralmente contém um sistema de proxy ou firewall que gerencia a conectividade de e para um conjunto de servidores de um tipo específico. Serviços de faturamento residirão em uma camada específica para o faturamento. Servidores de aplicações residem em uma camada específica para aplicações. Os serviços de banco de dados devem ser gerenciados em uma camada de banco de dados. Todos esses sistemas trabalham juntos com base nas regras de entrada e saída aplicadas nos servidores proxy.

Como resultado, o modelo de segurança para o ecossistema de serviços pode ser facilmente dividido em um conjunto de componentes. Esses componentes serão discutidos neste documento.

3 O modelo de segurança

A segurança em ambientes de serviços para endpoints pode ser desenvolvida usando infraestrutura, estratégias e políticas já conhecidas, independentemente da topologia ou das inovações usadas para construir uma arquitetura de aplicação. Cada aspecto do ecossistema de serviço pode ser dividido em componentes. Esses componentes devem ser protegidos individualmente, mas usando metodologias semelhantes.

Considere, por exemplo os componentes comumente utilizados no desenvolvimento de um serviço simples que seja capaz de colocar em campo as consultas e enviar respostas de e para pontos de extremidade, parceiros e usuários. Este modelo deve conter, mas não se limitar a, os seguintes níveis:

- Uma camada de serviço web
- Uma camada de servidor de aplicações
- Uma camada de banco de dados
- Uma camada de autenticação
- Uma camada de rede
- Camadas de aplicações de terceiros, como uma camada específica para faturamento

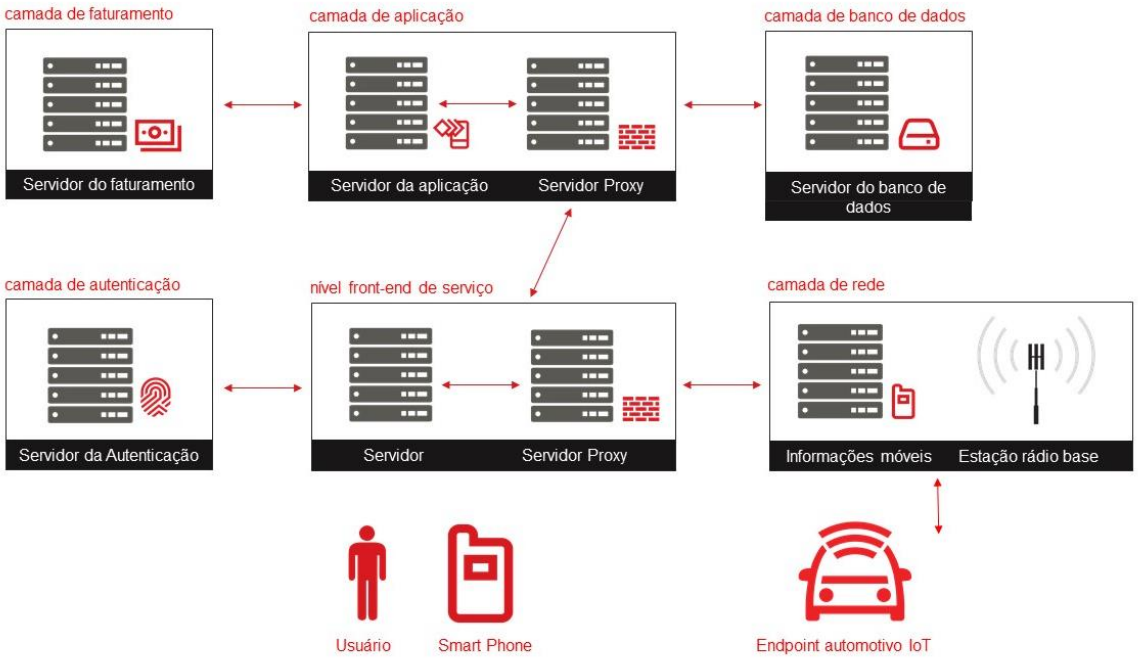

Figura 4 – Um exemplo de ecossistema de serviços com níveis separados

Mesmo que haja apenas um servidor em cada camada, é mais eficaz, em termos de arquitetura, separar cada conceito lógico em sua própria camada. Isso também ajuda a isolar uma camada de tecnologia de outras camadas caso haja comprometimento, ou caso o sistema precise aumentar de escala para atender a mais solicitações.

Se um tipo de sistema for considerado da perspectiva de um tipo de camada, ele pode ser mais facilmente protegido, dimensionado para a demanda, descomissionado e desativado. O único requisito é que haja uma API versátil o suficiente para ser aumentada ou ajustada durante toda a vida útil da camada. Definir essa API está fora do escopo deste documento. No entanto, recomendações relacionadas a atributos de segurança de alto nível da API que a organização escolher ou definir serão discutidas aqui.

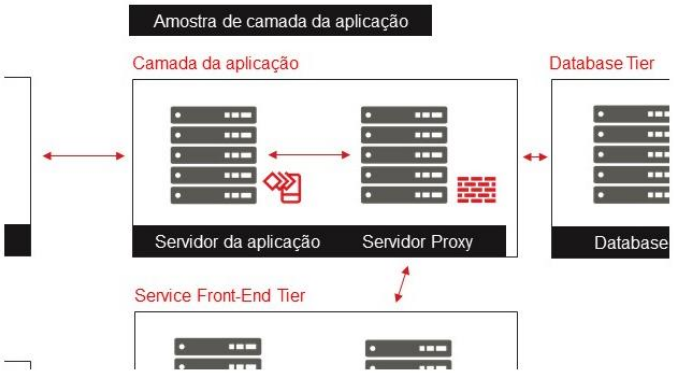

Figura 5 – Uma camada de aplicação protegida pela tecnologia de firewall

O exemplo acima descreve uma camada um pouco mais completa. A única ampliação necessária para representar a camada é um servidor proxy. Esse servidor proxy é apenas um descritor que representa a tecnologia de segurança que será empregada na camada. Independentemente de o controle real ser um firewall de hardware, firewall de software, grupos de segurança, Listas de Controle de Acesso (ACL) ou outra tecnologia, haverá um componente que exige controles de entrada e saída em nome da camada.

Ao escolher ou definir uma API, a organização deve considerar as especificações existentes que podem resolver as preocupações da equipe de engenharia, considerando, em particular, as seguintes especificações:

- ETSI SmartM2M TS 102 690, ETSI SmartM2M TS 102 921 [14]
- oneM2M TS-0001, oneM2M TS-0003 [15]
- 3GPP TS 33.220 [16]

Para componentes acessíveis publicamente, como a camada de serviço front-end, a única ampliação necessária ao modelo é de um componente de segurança adicional para viabilizar:

- Proteção contra o ataque distribuído de negação de serviço (DDoS)
- Balanceamento de carga
- Redundância
- Recurso opcional do Web Application Firewall (WAF)

As tecnologias acima devem ser implementadas para que qualquer serviço funcione adequadamente e para garantir que o serviço que eles protegem seja disponibilizado até mesmo em ambientes com recursos mais limitados. A definição desses componentes está fora do escopo deste documento, mas pode ser investigada mais detalhadamente por meio de consulta às seguintes entidades:

- The Cloud Security Alliance
- NIST Cloud Computing Standards
- FedRAMP
- Cisco Network Management Guidelines

Outro atributo necessário para que a camada funcione com segurança é a própria definição do servidor. Isso é definido pelos controles administrativos, de aplicação e do sistema operacional internos à plataforma escolhida pela equipe de engenharia.

A lista não exaustiva de problemas internos ao ambiente da plataforma inclui:

- Registro em um serviço de log centralizado
- Autenticação e autorização administrativa
- Execução de segurança nas comunicações
- Backup, restauração e duplicação de dados
- Separação de deveres de aplicação
- Monitoramento e integridade do Sistema

3.1 Ataques à infraestrutura de rede

Hackers que tentem comprometer o serviço do endpoint da perspectiva da rede presumirão que existem pontos fracos na maneira como as entidades se comunicam e vulnerabilidades nos serviços expostos por meio de pontos de acesso de serviço. Esses ataques presumem que uma posição privilegiada na rede equivale a uma posição de poder sobre o canal de comunicação.

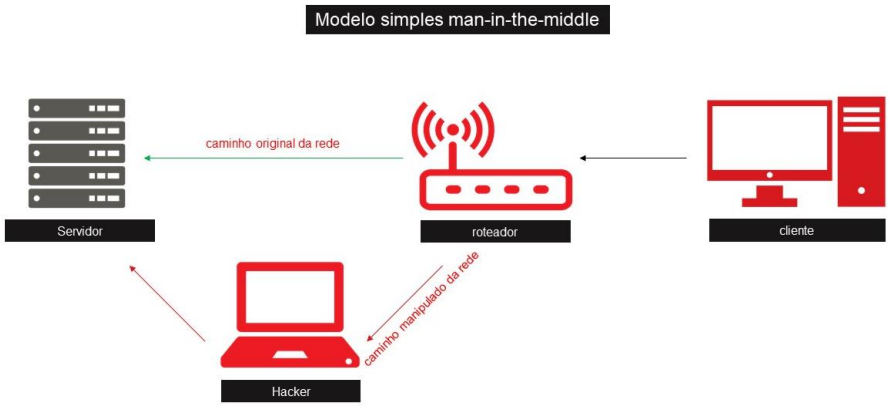

Figura 6 - Um exemplo de modelo de ataque “Man In The Middle”

A forma mais comum de ataque neste modelo é o ataque MITM (Man-In-The-Middle). Esse ataque presume ou que não há autenticação dos pares ou autenticação unilateral, ou que há falha na autenticação mútua no canal de comunicação. O objetivo de um hacker é representar

um lado do diálogo para forçar o outro lado (par) a realizar ações em nome do hacker. Esse ataque pode ser atenuado pela imposição da autenticação mútua, que exige uma raiz organizacional de confiança bem definida, uma base de computação confiável (TCB) e um modelo de comunicação.
Outros exemplos incluem ataques contra o Forward Secrecy, análise de criptografia das comunicações e ataques de canal lateral. Estes devem ser mitigados pelo uso de protocolos, algoritmos e padrões de criptografia apropriados.

Esses ataques são difíceis e exigem acesso à infraestrutura de rede, seja internamente a uma organização, na infraestrutura core da Internet entre uma organização e seus parceiros ou no ecossistema do endpoint, ou na infraestrutura próxima a endpoints. O ataque mais simples e mais comum consiste em tentar manipular a infraestrutura de rede do endpoint, como a rede Wi-Fi, ethernet ou celular, para obter uma posição de privilégio entre o serviço e seu par.

Ataques contra a infraestrutura de um único endpoint são restritos a ele ou ao grupo de endpoints disponíveis nesse local físico. Ataques contra a infraestrutura core da Internet geralmente envolvem sequestro no Border Gateway Protocol (BGP), atacando um roteador central ou abusando da infraestrutura DNS (Domain Name Service). Esses ataques viabilizariam uma posição de privilégio mais dissociada de um alvo em particular, potencialmente permitindo que o hacker tivesse acesso para atacar vários sistemas de uma só vez. Os ataques contra a infraestrutura interna de rede exigem acesso à rede interna, o que implica um ataque interno ou uma posição de privilégio existente dentro do ambiente de uma corporação, o que pode implicar em comprometimento de sistema mais profundo.

Independentemente do tipo de ataque utilizado, esse modelo é fácil de mitigar usando autenticação mútua, sigilo antecipado e algoritmos e protocolos criptográficos apropriados. Essas ações impedirão que o hacker tenha possibilidade de abusar dessa infraestrutura ou aumentará os custos desse tipo de ataque, tornando sua implementação inviável para o invasor.

3.2 Ataques à infraestrutura de nuvem ou de conteiner

Esses ataques pressupõem uma posição de privilégio no ambiente de infraestrutura da nuvem ou do conteiner. Se um hacker é capaz de comprometer uma rede de serviços em nuvem, por exemplo, ele pode ter acesso a hosts que executam sistemas de máquinas virtuais convidados. Isso permitiria que o hacker inspecionasse e modificasse os sistemas de máquinas virtuais em execução. O hacker pode ter objetivos específicos em mente, ou pode ter tido sorte e comprometido um provedor de serviços em nuvem apenas pelo acesso a muitos tipos diferentes de sistemas com dados valiosos.

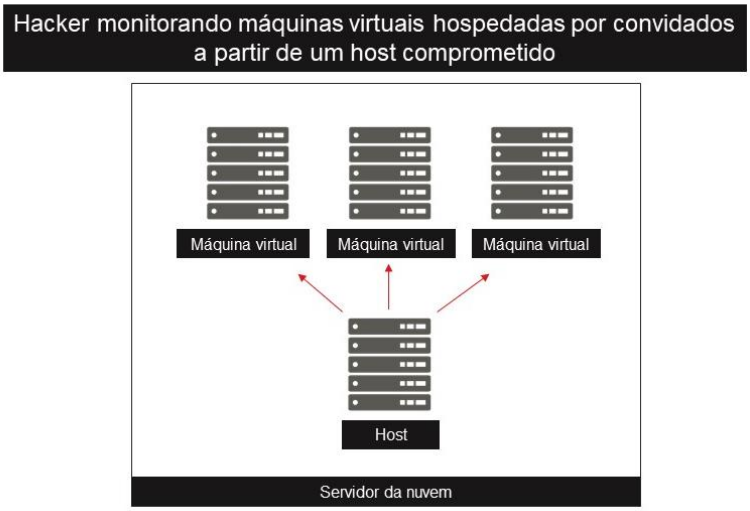

Figura 7 – Exemplo de modelo de ataque VM (virtual machine, ou máquina virtual)

Outro ataque à infraestrutura da nuvem ou conteiner pressupõe que o hacker tenha controle sobre uma máquina virtual no mesmo servidor físico que a máquina virtual de destino. O hacker pode usar vários métodos para comprometer outras máquinas virtuais em um servidor físico. Ele poderia:

- Usar uma vulnerabilidade na infraestrutura da máquina virtual para escapar do guest para um host
- Usar um ataque de canal lateral para inferir chaves secretas de outra máquina virtual convidada
- Consumir recursos excessivos no servidor físico para forçar a máquina virtual alvo a migrar para um servidor físico sobre o qual o hacker tenha mais controle

Independentemente do modelo de ataque utilizado, há pouco que uma empresa possa fazer para se proteger contra esse risco. Em vez disso, o provedor de serviços em nuvem deve implementar a funcionalidade adequada para reduzir a probabilidade de um hacker subverter a infraestrutura da nuvem ou do conteiner.

Uma maneira de reduzir esse risco é implementar uma arquitetura baseada em conteiner que limita cada conteiner a um usuário específico e uma identidade criptográfica exclusiva. Embora essa atividade exija muitos recursos e possa gerar custos adicionais, ela diminuirá a possibilidade de um hacker de abusar da infraestrutura da máquina virtual para obter acesso a vários usuários ou a vários serviços de uma só vez.

Embora uma posição de privilégio em um ambiente de nuvem ou conteiner seja uma ameaça crítica para aplicações executadas em máquinas virtuais convidadas, um alto grau de habilidade, tempo e recursos é necessário para obter acesso a essa posição. Uma vez que o acesso é adquirido, o hacker deve mantê-lo por tempo suficiente para identificar qual sistema contém a máquina virtual que é relevante para seus interesses. Além disso, ele deve ser capaz de monitorar ou alterar essa máquina virtual sem ser detectado pelo subsistema de incidentes do provedor de serviços de nuvem. Isso pode representar um desafio significativo e diminuir a probabilidade de um comprometimento.

No entanto, é importante notar que esse tipo de comprometimento normalmente é indetectável pela máquina virtual convidada ou por uma aplicação executada nela. Assim, é possível reunir métricas que revelam anomalias no comportamento de uma determinada máquina virtual na nuvem ou conteiner, mas pode ser extremamente difícil identificar se um comprometimento realmente ocorreu ou não. Isso acontece porque qualquer hacker com privilégios suficientes na camada de host da infraestrutura da máquina virtual seria capaz de manipular o convidado para dificultar a detecção de manipulação.

Ataques de convidado em outro convidado são excepcionalmente difíceis de detectar, até mesmo pelo provedor de serviços em nuvem. É importante notar, no entanto, que esses ataques são, em grande parte, teóricos por natureza. Embora os ataques de canal lateral sejam possíveis, sua praticidade ainda é debatida, pois esses ataques exigem um nível de consistência na plataforma de execução subjacente que não é garantida em um ambiente do mundo real. Além disso, ataques de escalação de convidados a hosts em um ambiente de máquina virtual, conteiner ou hipervisor são difíceis de encontrar e ainda mais difíceis de explorar. Isso torna muito menos provável que uma vulnerabili7dade resulte na exploração de uma quantidade massiva de convidados ou de um alvo específico.

Portanto, embora seja uma posição significativa de privilégio para os invasores, a probabilidade de um ataque bem-sucedido é baixa, pois a dificuldade, o custo e a oportunidade tornam a exploração quase impraticável.

3.3 Ataques a serviços de aplicações

Embora discussões sobre arquitetura de execução de aplicações estejam amplamente fora do escopo deste documento, é importante observar que essa camada representa o maior risco de um ataque. Se o ecossistema de serviços tiver sido configurado corretamente, conforme recomendado neste guia, os invasores migrarão dos ataques de infraestrutura de rede para a própria aplicação.

A aplicação apresenta a maior camada de complexidade em qualquer produto ou serviço e sempre apresentará a possibilidade de um hacker aumentar seu nível de privilégio por meio de várias camadas de tecnologia. Portanto, embora o objetivo deste documento seja direcionar o foco de um ataque para longe da infraestrutura de rede, ele está direcionando o foco para o único lugar em que o sucesso é muito mais provável.

Para diminuir a possibilidade de ataque, é preciso revisar muitos dos bem elaborados documentos sobre segurança de aplicações (por exemplo, o OWASP Secure Application Design Project [5]), para implementar a arquitetura de execução de aplicações da forma mais segura possível.

3.4 Privacidade

Embora parcerias sejam desenvolvidas para utilizar dados, métricas ou outros componentes centrados no usuário para agregar valor ao sistema como um todo, nunca existe garantia plena quanto ao nível de segurança implementado pelo parceiro. Em vez de simplesmente passar informações a terceiros, é necessário avaliar que tipos de dados devem ser entregues, qual deve ser o retorno tangível e como essas informações devem ser protegidas.

A responsabilidade legal pode ser diminuída por meio de contratos e cláusulas de seguro; no entanto, a perda de clientes pode ocorrer devido a uma falha de terceiros. Em vez de arriscar essa perda de negócios, uma organização deve avaliar as equipes de engenharia de terceiros para determinar o nível de segurança que elas aplicam à sua infraestrutura, aplicações e APIs. Se o nível de segurança não for suficiente, é recomendável procurar parceiros alternativos.

3.5 Objetos mal-intencionados

Sistemas de terceiros são desenvolvidos para apresentar informações ou multimídia aos consumidores. Uma maneira óbvia de atingir esse objetivo é por meio da publicidade. Vários tipos de arquivos são complexos em sua estrutura e são difíceis de serem analisados corretamente pelo software. As redes de publicidade são um canal interessante para a distribuição de malware. Redes de distribuição de conteúdo (CDNs) também representam canais potenciais para distribuição de malware. Qualquer sistema que ofereça tipos complexos de multimídia, ou pacotes de código (seja da web ou executável) para fins de renderização de informações dinâmicas, pode trafegar malwares.

Portanto, é imperativo que a organização avalie os diferentes tipos de ofertas tecnológicas que passarão por um determinado canal. A empresa deve decidir o que permitir entregar e o que é excessivo para os clientes. Por exemplo, uma empresa de publicidade pode querer trafegar código Java para sistemas dos clientes por meio de uma aplicação de serviço proxy oferecida a parceiros por provedores de IoT. A empresa precisará decidir se os sistemas dos clientes em execução em determinados ambientes são mais suscetíveis a ataques da tecnologia Java. Se isso for verdade, a empresa pode querer desautorizar o Java, mas permitir que outras tecnologias, como o Hypertext Mark-up Language (HTML), passem.

Como o malware funciona de várias formas, desde tipos de arquivos polimórficos até explorações de Adobe Flash, Java e multimídia, não existe uma maneira única e uniforme de garantir a segurança do usuário final. Uma solução simples seria a equipe de engenharia impor uma política sobre quais tecnologias são usadas em seus canais e como seus usuários serão afetados. Os subsistemas de monitoramento podem ser colocados em funcionamento, assim como os sandboxes, para garantir que qualquer objeto renderizado em um sistema de cliente esteja menos sujeito a abusos.

3.6 Autenticação e autorização

Parceiros geralmente oferecem serviços específicos apenas para um subconjunto de usuários. Isso pode incluir serviços pagos que um usuário pode assinar opcionalmente. E, também, pode representar uma maneira de um usuário autenticar-se no sistema utilizando credenciais compartilhadas com uma tecnologia separada e conhecida, como APIs de autenticação existentes de provedores de serviços de rede, infraestrutura de rede social e entidades de gerenciamento M2M ou IoT existentes.

Embora sejam excelentes maneiras de compartilhar tecnologia entre plataformas, os engenheiros devem garantir que a tecnologia não consuma inadvertidamente credenciais que possam ser usadas para abusar de permissões não concedidas expressamente a um serviço de terceiros. Por exemplo, certas APIs de plataforma permitem a restrição de permissões a uma classe que seja aceita ou negada pelo usuário. Isso permite que o usuário ajuste a experiência para uma que seja adequada às suas necessidades

específicas de privacidade. Se a plataforma não puder oferecer permissões granulares de segurança, deverá listar as tecnologias às quais deseja acesso.

É necessário que a equipe de engenharia solicite de seus parceiros que a oferta ative permissões granulares para garantir que a revogação de um serviço não permita inadvertidamente uma janela de exposição dos dados dos usuários que continua mesmo após a assinatura ser revogada.

3.7 Falsos positivos e falsos negativos

Embora os serviços de monitoramento e registro sejam formas excepcionais de incrementar uma Infraestrutura de segurança existente, eles devem ser cuidadosamente avaliados quanto a falsos positivos e falsos negativos. Como esses sistemas só interpretam dados originados de vários ecossistemas em um produto ou serviço da IoT, e esses sistemas não são desenvolvidos pela equipe de engenharia interna, eles só podem oferecer informações artificiais sobre um evento. Eles podem, contudo, não ser capazes de distinguir com precisão se um evento adverso está realmente ocorrendo.

Como resultado, é importante que as equipes de TI e engenharia busquem determinar se um evento suspeito é, de fato, atribuível a um comportamento mal-intencionado. Isso ajudará a reduzir a possibilidade de a equipe de monitoramento não permitir o acesso de um usuário legítimo ao sistema. Se esse processo for automatizado e o processo estiver incorreto, muitos usuários poderão ser excluídos de seu serviço legítimo devido a um falso positivo que pode ser atribuído a uma anomalia na aplicação ou na infraestrutura do cliente. Quando um evento crítico está ocorrendo e é questionável, as equipes de TI e engenharia devem examinar os dados para avaliar se realmente há um ataque.

Além disso, os engenheiros devem ter o cuidado de modelar as informações adquiridas por meio de canais analógicos. Falsos positivos e falsos negativos, especialmente em ecossistemas em que os dados precisam ser processados a taxas excepcionalmente altas, podem ter consequências significativas se a aplicação não avaliar corretamente o curso de ação mais seguro, caso não se possa confiar plenamente nos dados adquiridos. É importante notar que, com tempo, tecnologia e experiência suficientes, todo dado analógico pode ser simulado em um sistema digital.

4 Perguntas frequentes sobre segurança

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

4.1 Como combatemos a clonagem?

A diferenciação entre dispositivos válidos fabricados pelo provedor de serviços de IoT e dispositivos que são reproduções ou “rip offs” (clones) é um desafio. Nenhum provedor de serviços de IoT deseja fornecer serviços para endpoints não autorizados, pois os provedores de serviços precisam pagar pelo tempo de CPU, largura de banda,

armazenamento em disco e outros recursos. A organização deve pagar independentemente de o dispositivo ter sido fabricado pelo provedor de serviços de IoT ou não.

Além disso, a organização deve ser capaz de discernir se sua arquitetura endpoint está sendo subvertida. Isso permite que a organização reaja a um dispositivo que foi clonado em várias instâncias do mesmo dispositivo, o que pode ser feito por um fabricante inescrupuloso ou por um hacker tentando se passar por um usuário em particular.

Revise as recomendações a seguir para obter ajuda sobre como usar o Serviço para combater a clonagem:

- Definir uma raiz organizacional de confiança
- Usar os serviços de autenticação de rede
- Forçar autenticação por meio do ecossistema de serviços
- Definir autenticação e autorização da camada de aplicação

4.2 Como os usuários são autenticados por meio do endpoint?

Um dos conceitos mais importantes da IoT é a separação da autenticação do endpoint da autenticação do usuário. Um endpoint pode ser autenticado por sua base de computação confiável (TCB), mas a forma de autenticar o usuário é um processo separado que depende da TCB do endpoint para a segurança das comunicações. O que é mais importante sobre essa abstração é avaliar o quão confiável é o canal de comunicação para a autenticação do usuário.

Por exemplo, se a confiabilidade de um endpoint for baixa porque não há TCB do endpoint ou se for usada uma implementação frágil da TCB do endpoint, o mecanismo de autenticação de usuário que depende do software / firmware do endpoint para executar não pode ser confiável. Isso significa que qualquer usuário autenticado por meio de um endpoint não poderia ser considerado autenticado.

Por outro lado, se o esquema de autenticação for facilmente burlado, uma TCB do endpoint bem arquitetada pode prover uma autenticação fraca do usuário final. Assim, o ecossistema de serviços deve contar com a confiabilidade do endpoint, bem como com a implementação do mecanismo de autenticação, para garantir que o ecossistema do serviço possa garantir que o usuário correto esteja conectado ao sistema.

Considere as seguintes recomendações para ajudar a lidar com essas complexidades:

- Implementar uma base de computação confiável para o serviço
- Definir uma raiz organizacional de confiança
- Definir um modelo de autorização claro
- Usar os serviços de autenticação de rede
- Forçar autenticação por meio do ecossistema de serviços
- Aplicar política de senha forte
- Definir autenticação e autorização da camada de aplicação

4.3 Como o serviço pode identificar o comportamento anômalo do endpoint?

Um dos aspectos mais desafiadores do gerenciamento de endpoints em uma rede distribuída de IoT é determinar se um endpoint está ou não se comportando de maneira anormal. Isso não é importante apenas partindo de uma perspectiva de segurança, mas também de uma perspectiva de confiabilidade. Muitas vezes, um comportamento anormal pode indicar um problema com o firmware ou hardware e pode ser um sinal de que a organização deve se preparar para corrigir um problema inesperado. No entanto, se o comportamento for isolado em uma parte da rede que não pode ser analisada pelo provedor de serviços de IoT, essas métricas serão perdidas, deixando a organização em relativa desvantagem.

Resolver esse problema requer a capacidade de inspecionar o comportamento no endpoint, na camada de rede e no ecossistema do serviço. No entanto, se a infraestrutura, os serviços e as parcerias certas não forem criados para reunir esses dados, a organização não terá as informações necessárias para determinar se há um problema ou se um problema está relacionado à segurança ou confiabilidade.

Avalie as seguintes recomendações da perspectiva do ecossistema de serviço:

- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir uma abordagem de registro e monitoramento de sistemas
- Definir um modelo de comunicação
- Usar serviços de autenticação de rede
- Implementar validação de entrada
- Implementar filtro de saída
- Usar serviços aprimorados de monitoramento de parceiros
- Usar um APN privado para conectividade sem fio
- Definir um modelo de avaliação de falso negativo e falso positivo

4.4 Como o serviço pode restringir um endpoint com comportamento anormal?

Quando se identifica um endpoint de comportamento anormal, o serviço deve tomar decisões sobre quais recursos devem ser limitados ou restritos. Essa questão é relevante para todas as camadas da infraestrutura de serviços.

Um endpoint com capacidade celular que se conecta e desconecta constantemente da rede móvel em um loop frenético, por exemplo, deve ser desativado à força até que o comportamento errático seja resolvido. Outro exemplo útil é um endpoint comprometido que um hacker está usando para tentar atacar serviços de back-end. Nesse cenário, os serviços de back-end devem impedir que o endpoint comprometido atinja os serviços.

Como lidar com cada cenário depende do provedor de serviços de IoT e depende dos objetivos do negócio e, também, de como os incidentes devem ser tratados. Para ajudar no desenvolvimento dessas diretrizes, considere as seguintes recomendações:

- Definir uma raiz organizacional de confiança
- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir um modelo de Respostas a Incidentes de Segurança em Computadores
- Definir um modelo de recuperação

- Definir um modelo de desligamento (sunsetting)
- Definir um modelo de comunicação
- Definir uma política de violação para dados expostos
- Forçar autenticação por meio do ecossistema de serviços
- Usar um APN privado para conectividade sem fio
- Definir um modelo de avaliação de falso negativo e falso positivo

4.5 Como posso determinar se um servidor foi invadido?

Embora anomalias nos endpoints sejam mais enigmáticas e requeiram uma grande quantidade de análises comportamentais para que a maioria dos ataques seja detectado, o ecossistema do serviço é mais direto. Serviços e servidores são implantados em um ambiente rigidamente controlado pelo provedor de serviços de IoT ou por parceiros que gerenciam a infraestrutura de nuvem ou de servidor. Sendo assim, a organização e seus parceiros podem usar sistemas de monitoramento e diagnóstico prontamente disponíveis para identificar e conter possíveis problemas.

Revise as seguintes recomendações para ajudar nesse processo:

- Definir um modelo de administração
- Definir uma abordagem de registro e monitoramento de sistemas
- Definir um modelo de Respostas a Incidentes de Segurança em Computadores
- Implementar validação de entrada
- Implementar filtro de saída

4.6 O que posso fazer quando um servidor é invadido?

Quando um servidor é identificado como comprometido, a equipe de administração precisa resolver o problema da maneira mais rápida e eficiente possível. A complexidade em fazê-lo muitas vezes resulta da determinação dos recursos, informações e contas que foram colocadas em risco. Em alguns ambientes mal planejados, os efeitos de um comprometimento geralmente não são quantificáveis. Portanto, a organização deve implementar um plano para concomitantemente resolver a vulnerabilidade de segurança e proteger os ativos em risco. Uma vez que o ecossistema tenha sido protegido e a vulnerabilidade, eliminada, a organização pode prosseguir com um plano para reconstruir a tecnologia afetada.

Revise as seguintes recomendações para mais informações:

- Definir um modelo de Respostas a Incidentes de Segurança em Computadores
- Definir um modelo de recuperação
- Definir um modelo de sunsetting
- Definir um conjunto de classificações de segurança
- Definir classificações para conjuntos de tipos de dados

4.7 Como os administradores devem interagir com servidores e serviços

O desenvolvimento de um modelo administrativo que não coloca em risco o ecossistema de serviços é uma parte importante da arquitetura de um serviço de IoT. Existem várias camadas de administração e cada camada deve ser considerada pelas equipes de engenharia e segurança. Por exemplo, os administradores que controlam o servidor

(independentemente de ser usada uma arquitetura virtual, de micro-serviço ou uni-kernel) devem interagir com os servidores ativos por meio de um canal de comunicação confiável e seguro. Os administradores que controlam a aplicação eb geralmente interagem com a aplicação na mesma camada de comunicação eb, mas por meio de uma aplicação especializada incorporada no código.

Independentemente da necessidade administrativa, a interface deve ter acesso restrito para limitar a possibilidade de hackers interagirem ou abusarem da tecnologia. Considere os seguintes recursos:

- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir um modelo de administração
- Definir um modelo claro de autorização
- Definir um modelo de comunicação
- Usar um APN privado para conectividade sem fio

4.8 Como a arquitetura de serviços pode limitar o impacto de um comprometimento?

Um atributo fascinante de uma rede de IoT é sua capacidade única de unir serviços a consumidores específicos. Nos serviços web, cada usuário deve ter a capacidade de interagir com o serviço de qualquer tipo de dispositivo ou, potencialmente, de qualquer lugar do mundo. Isso não se aplica à tecnologia da IoT, que normalmente requer um endpoint específico para interagir com os serviços de IoT. Devido a essa diferença, os arquitetos de ecossistema de servidores podem aproveitar o relacionamento personalizado entre os endpoints e os consumidores para restringir o acesso de um endpoint aos dados de back-end.

Considere o cenário em que um endpoint está enviando métricas do sensor para um serviço de back-end. Em uma arquitetura de microsserviço, o ecossistema de serviços pode implantar um micro-serviço ou uni-kernel específico para lidar com um consumidor específico. Usando essa arquitetura, a equipe de engenharia pode garantir que o micro-serviço seja provisionado apenas com recursos e funcionalidades de acesso necessários para fornecer dados e serviços específicos para o consumidor individual.

Isso significa que, se um serviço for comprometido e o endpoint for a única tecnologia que pode se comunicar com esse serviço específico, não há nenhum benefício adicional em comprometer esse serviço, pois o acesso obtido por meio do comprometimento será limitado aos recursos que já estariam disponíveis para o endpoint. Em essência, não há ganhos no ataque.

Revise as seguintes recomendações para assistência:

- Implementar uma base de computação confiável do serviço
- Definir um método de bootstrap
- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir um modelo de armazenamento persistente
- Definir um modelo de administração
- Definir um modelo de sunsetting
- Definir um modelo claro de autorização

- Provisionar servidores quando possível
- Definir um ambiente de execução de aplicações
- Comprometimentos com máquinas virtuais

4.9 Como a arquitetura de serviços pode reduzir a perda de dados durante um comprometimento?

Outro atributo interessante da arquitetura de IoT é a redução da perda de dados. Isso é semelhante a como os serviços podem ser isolados para um usuário específico. Os dados também podem ser isolados por usuário específico depois que o usuário for autenticado. No entanto, o armazenamento de dados não pode ser facilmente implementado por usuário devido aos custos da infraestrutura de base de dados e armazenamento.

Em vez disso, tokens exclusivos devem ser provisionados para serviços que atuam em nome de um usuário específico dentro da infraestrutura de armazenamento. Dessa forma, um hackercom acesso ao ambiente de armazenamento de dados poderá se conectar ao serviço, mas não poderá interagir, recuperar ou alterar dados de usuários que não estejam comprometidos.

Do ponto de vista da camada de rede, reduzir o fluxo de tráfego do ecossistema do servidor para a internet também é um requisito. Os controles de saída forçam um hacker a trafegar dados de propriedade intelectual ou de clientes por meio de canais específicos. Isso pode aumentar a dificuldade de mover grandes quantidades de dados ou forçá-lo a utilizar camadas de comunicação que podem detectar e interromper a comunicação durante incidentes.

Para mais informações, considere as seguintes recomendações:

- Definir um método de bootstrap
- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir um modelo de armazenamento persistente
- Definir um conjunto de classificações de segurança
- Definir classificações para conjuntos de tipos de dados
- Provisionar Servidores quando possível
- Definir um ambiente de execução de aplicações
- Regras de firewall “Default-Open” ou falha aberta

4.10 Como a arquitetura de serviço pode limitar a conectividade de usuários não autorizados?

Um benefício de aproveitar as arquiteturas comuns da IoT é reduzir a possibilidade de usuários da internet não autorizados de se conectarem diretamente aos serviços de back-end. A maioria das aplicações web não tem esse luxo e deve estar disponível para uso público. Na IoT, entretanto, como o endpoint é a entidade que deve se conectar a um serviço específico, uma Virtual Private Network (VPN) pode ser usada para restringir quem tem acesso a serviços de back-end. Isso pode ser implementado em protocolos padrão da internet ou pode ser implementado usando serviços móveis, como um APN privado. Revise as seguintes recomendações para mais informações:

- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Usar um APN privado para conectividade sem fio

4.11 Como reduzir a probabilidade de exploração remota?

A exploração remota de aplicações e serviços da web é uma preocupação constante dos administradores de infraestrutura. Garantir que hackers não tenham uma rota para a rede interna ou, simplesmente, para acessar recursos valiosos, é uma batalha diária. A única maneira de reduzir a possibilidade de hackers comprometerem o ecossistema de serviços é reduzir os potenciais alvos a um conjunto gerenciável de serviços que podem ser rápida e facilmente mantidos. O segundo aprimoramento mais importante para a arquitetura é o design da arquitetura subjacente: a arquitetura de execução, a configuração do sistema operacional, a cadeia de ferramentas de implantação, a segurança da linguagem de programação e outras opções que definem a segurança que uma aplicação pode executar. Essas opções podem ser a diferença entre uma falha de aplicação e um comprometimento de infraestrutura.

Para mais informações sobre como reduzir o potencial de exploração remota, consulte a lista abaixo:

- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir um modelo de atualização
- Implementar validação de entrada
- Implementar filtragem de saída
- Regras de firewall “Default-Open” ou falha aberta
- Definir um ambiente de execução de aplicações
- Rowhammer e ataques similares
- Comprometimentos com máquinas virtuais

4.12 Como o serviço pode gerenciar a privacidade do usuário?

Na medida em que os provedores de serviços de IoT crescem, eles invariavelmente criam parcerias com organizações que utilizarão os dados do consumidor de maneiras inovadoras. No entanto, esses dados trazem um custo para a privacidade do consumidor. Os consumidores devem ter o direito de determinar quais dados são compartilhados com os parceiros e como serão usados. Além disso, os parceiros devem ser obrigados a usar os dados de maneiras específicas. Os modelos de autorização podem ajudar nisso, mas isso implica uma discussão muito maior sobre privacidade, repercussões legais, seguro comercial e muito mais.

Para iniciar a discussão na sua organização, analise as seguintes recomendações:

- Definir um conjunto de classificações de segurança
- Definir classificações para conjuntos de tipos de dados
- Definir um modelo claro de autorização
- Definir uma política de vazamento para dados expostos
- Avaliar o modelo de privacidade das comunicações
- Definir uma política de distribuição de dados de terceiros
- Criar um filtro de dados de terceiros
- Criar uma API para os usuários controlarem os atributos de privacidade

4.13 Como um serviço pode melhorar sua disponibilidade?

Ataques de negação de serviço (DoS) ou ataques de negação de serviço distribuído (DDoS) são tão comuns na internet moderna que toda empresa deve estar preparada para enfrentar um grande ataque dessa classe e deve ser capaz de permanecer on-line mesmo sob ataques prolongados. A razão pela qual esses ataques se tornaram tão comuns é que eles exigem muito pouca habilidade para ser executados e as ferramentas para implementá-los estão amplamente disponíveis online. Na verdade, há serviços on-line nos quais é possível pagar a um hackerpara lançar um ataque DDoS contra um determinado alvo.

Como resultado, modelos inteiramente novos para a disponibilidade de serviços foram construídos para combater essa ameaça. Considere as seguintes recomendações ao criar o ecossistema de serviços:

- Definir uma infraestrutura de segurança para sistemas expostos à internet pública
- Definir uma abordagem de registro e monitoramento de sistemas
- Definir um modelo de Respostas a Incidentes de Segurança em Computadores
- Definir um modelo de recuperação
- Definir um modelo de comunicação
- Regras de firewall “Default-Open” ou falha aberta

5 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 hacker terceiro.

5.1 Implemente uma base de computação confiável do serviço

Uma Base de Computação Confiável (TCB) é um conjunto de hardware, software, protocolos e políticas. Uma TCB deve ser a base de qualquer plataforma de computação e deve definir o ambiente no qual uma aplicação pode ser executada de forma confiável, segura e com alta qualidade.

Uma TCB pode ser construída e implantada para qualquer classe de sistema, como equipamentos móveis (smartphones), endpoints IoT e até mesmo servidores dentro de um ecossistema de serviço. Todas as TCBs são compostas por tecnologias similares. No entanto, dependendo da classe de sistema, essas tecnologias podem ter características muito diferentes. Por exemplo, a inicialização de uma TCB em um servidor de nuvem parecerá muito diferente da inicialização de um endpoint.

Construir uma TCB em um ecossistema de serviços significa definir a maneira como a imagem de uma aplicação deve ser implementada. Uma imagem nesse contexto representa os dados binários brutos que compreendem o executável da aplicação, seus arquivos de configuração e seus metadados. Esses itens juntos são comumente chamados de imagem da aplicação ou simplesmente imagem. Na maioria dos ecossistemas de serviços modernos, os sistemas serão replicados, acionados ou reduzidos sob demanda para viabilizar a escalabilidade de forma reativa com as mudanças no ambiente computacional. Isso significa que uma TCB deve definir uma maneira de permitir que os sistemas sejam dimensionados de maneira eficaz, mantendo um modelo de segurança persistente.

Para fazer isso corretamente, a equipe deve:

- Padronizar a plataforma de computação:
- Escolher um conjunto de modelos de servidores físicos
- Selecionar um conjunto de plataformas de nuvem ou imagens de máquina virtual (VM)
- Definir o conjunto de aplicações, bibliotecas e arquivos de configuração a serem executados na plataforma de computação:
- Definir um ambiente de conteiner, se aplicável
- Gerar uma imagem de aplicação, composta pelo conjunto definido acima
- Assinar criptograficamente um arquivo da imagem usando a chave de assinatura da camada da TCB
- Armazenar com segurança o arquivo e a assinatura

A execução desse conjunto de tarefas resultará em uma imagem de aplicação aprovada que pode ser implantada em uma camada específica. Cada camada terá um hardware e um modelo de aplicação diferentes que funcionam melhor para essa camada específica. Por exemplo, o hardware de base de dados tem necessidades de desempenho e armazenamento muito diferentes das de um nível de aplicação. Uma camada de

armazenamento terá requisitos semelhantes em termos de hardware de armazenamento se comparada aos de uma camada de base de dados, mas terá requisitos de desempenho diferentes. Depois de padronizar a definição de cada camada, o resultado é uma imagem que pode ser implantada e verificada em cada plataforma de hardware.

A dificuldade em implantar uma TCB vem de:

- Configurar uma raiz de confiança organizacional para gerenciar a assinatura criptográfica de imagens
- Configurar um procedimento para assinar cada imagem
- Configurar um procedimento para verificar cada imagem
- Configurar um procedimento para gerar imagens de maneira automatizada, mas com verificação de imagem

Considere usar o material das seguintes organizações para ajudar com esta recomendação:

- eGlobalPlatform Card Specification [11]
- Trusted Computing Group's TPM Specification [6]
- GlobalPlatform TEE Internal Core API Specification [12]

5.1.1 Risco

Sem uma base de computação confiável bem definida, as plataformas de computação não podem verificar se estão sendo executadas em uma configuração aprovada pela equipe de engenharia. Isso é importante, pois o subsistema da aplicação deve ser capaz de determinar se ele foi comprometido por um hacker. Uma TCB pode ser usada para remediar esse risco, bem como fornecer uma camada de segurança para todas as comunicações da rede.

5.2 Definina uma raiz organizacional de confiança

Uma raiz organizacional de confiança é um certificado ou sistema baseado em chave pública para autenticar entidades de plataforma de computação em uma organização. Cada plataforma de computação em um ecossistema de serviços deve ser autenticada criptograficamente durante as comunicações de rede. Isso diminui a possibilidade de um insider ou alguém dentro de uma posição de rede privilegiada personificar ou abusar da confiança de um sistema privilegiado.

Para criar uma raiz organizacional de confiança, basta executar as seguintes ações:

- Construir ou adquirir, por exemplo, um Módulo de Segurança de Hardware (HSM) para armazenar o segredo da raiz organizacional
- Gerar um segredo de raiz e/ou certificado
- Assegurar-se de que a faceta privada do segredo esteja armazenada com segurança
- Gerar um conjunto de uma ou mais chaves de assinatura a serem usadas para a chave de assinatura da camada da TCB
- Assinar a faceta pública da chave de assinatura com a raiz organizacional
- Garantir que essas chaves não possam ser usadas sem autenticação e autorização dos líderes do negócio e da engenharia

Nesse caso, toda vez que um novo sistema de camadas é definido, sua chave ou certificado criptográfico exclusivo pode ser assinado pela chave de assinatura. Se outro sistema se

conectar a esse novo sistema, ele poderá validar a identidade do sistema, verificando a cadeia de confiança definida pela raiz organizacional.

Ele irá validar criptograficamente que as mensagens foram assinadas pela chave pública que representa o sistema. Em seguida, ele verificará a assinatura da chave de assinatura gerada pela chave pública exclusiva desse sistema. Posteriormente, o cliente deve verificar se a chave de assinatura é, de fato, a chave de assinatura autenticada pela raiz organizacional.

Considerando que cada conjunto de certificados ou segredos é restrito a menos indivíduos na organização, e as políticas e procedimentos definidos devem restringir quem pode usar esses segredos e quando, cada nível de confiança deve aumentar à medida que o cliente avança pela cadeia da raiz.

Um serviço deve ser definido de maneira que apresente recursos de autenticação para pares autorizados dentro do ecossistema de serviços. Por exemplo, a autenticação usando o certificado ou a cadeia secreta não pode ser usada por conta própria para garantir a segurança. Um serviço deve ser disponibilizado para verificar se os certificados foram revogados ou se estão válidos no momento. Outro serviço pode precisar ser usado para autenticar as identidades de servidores ou serviços com uma vida útil curta, dependendo dos requisitos da infraestrutura subjacente.

Durante a definição da raiz da confiança, considere os seguintes pontos:

- Cada segredo deve ser protegido contra abuso
- O uso interno de cada segredo deve ser rastreado e monitorado de forma verificável
- Cada indivíduo aprovado para utilizar um segredo deve usar autenticação multifator ao acessar o (s) segredo (s)
- Definir um conjunto de políticas e procedimentos que imponham o uso consistente e seguro pode ser um desafio
- Construir um processo para sunset ou revogar um certificado pode ser um desafio
- Identificar se uma chave foi abusada pode ser um desafio
- A escolha do conjunto correto de algoritmos criptográficos pode não ser intuitiva

Para mais informações sobre o conceito de raiz de confiança, considere as seguintes fontes de informação:

- Trusted Computing Group
- TPM Specification [6]
- TCG Guidance for Securing IoT [7]
- ISO 11889
- PKI Specifications
- RFC 2510
- RFC 3647

5.2.1 Risco

O risco de não usar uma raiz organizacional de confiança é que qualquer comprometimento de uma única chave pode resultar no comprometimento de todo o ecossistema. Ao separar

a organização em uma hierarquia e implantar chaves separadas para a hierarquia, 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.

5.3 Definia um método de bootstrap

Para que uma aplicação seja executada corretamente, ela deve ser carregada e executada de forma consistente em uma plataforma confiável, de alta qualidade e segura. A TCB define como formular essa plataforma, mas o modelo de bootstrap define como a aplicação deve ser executada sobre ele.

Para definir um modelo de bootstrap efetivamente, os seguintes pontos devem ser considerados:

- Definir uma API que permita à aplicação identificar-se criptograficamente com seus pares
- Considerar a utilização de uma API existente definida por um líder confiável do setor
- Definir como a aplicação autenticará endpoints, serviços pareados e parceiros
- Definir como deve ser a configuração da aplicação
- Exigir que cada aplicação diferente tenha uma identidade exclusiva, especialmente aplicações executados em camadas separadas

Embora pareça intuitivo que uma aplicação deva se identificar criptograficamente a seus pares, e talvez não seja necessária uma API para fazer isso, o processo de produção não é inteiramente intuitivo. Isso porque, no modelo de bootstrap, é necessário considerar como a identidade criptográfica é provisionada para a aplicação. Como a aplicação adquire sua identidade? A identidade é adquirida com segurança? Qual é o processo para revogar segredos que a identidade usará no caso de os segredos precisarem ser atualizados ou alterados?

Em tempo de execução, as aplicações exigem determinados recursos para serem executadas de maneira eficaz. A aplicação deve ser capaz de se comunicar e executar autenticação mútua com todos os serviços externos, endpoints e parceiros envolvidos nesse processo.

A configuração de uma aplicação geralmente determina a segurança dela na produção. Uma configuração deve ser imposta e apresentada somente como leitura para uma aplicação. A aplicação, ou alguém que esteja abusando da infraestrutura da aplicação, não deve ser capaz de alterar a configuração de uma aplicação.

Use a raiz organizacional de confiança para definir modelos de confiança para cada camada implantada no ecossistema como um todo. Isso permitirá que cada aplicação separada tenha uma identidade criptográfica exclusiva, o que fornecerá aos pares a capacidade de diferenciar um serviço de base de dados de um serviço de aplicação, por exemplo.

5.3.1 Risco

Sem um modelo de bootstrap bem definido, o sistema não será capaz de verificar cada camada necessária para operar. Em essência, não há camadas de confiança em cada faceta da tecnologia como um todo. Essa falta de camadas de confiança representa uma complexidade que pode resultar em brechas passíveis de abuso por hackers.

5.4 Defina uma infraestrutura de segurança para sistemas expostos à internet pública

Diversas tecnologias de segurança e confiabilidade são necessárias para manter a disponibilidade, confidencialidade e integridade de serviços acessíveis ao público:

- Infraestrutura resistente a DDoS
- Infraestrutura de balanceamento de carga
- Sistemas de redundância
- Firewalls de aplicações web (opcional)
- Firewalls tradicionais

Essas tecnologias adicionais devem ser colocadas à frente do nível da aplicação para garantir que ela não possa ser manipulada por invasores públicos. Enquanto o modelo de segurança de comunicação corrigirá ou mitigará o potencial de um terceiro anônimo acessar o sistema, essas tecnologias diminuirão a possibilidade de um hacker tornar o sistema indisponível.

A segurança de front-end deve ser aplicada a todos os protocolos implementados pelos serviços. Por exemplo, se o serviço estiver disponível em IPv4 e IPv6, as mesmas restrições de segurança devem ser aplicadas ao serviço nos dois protocolos. Se um serviço for acessível pelo TCP, bem como pelo protocolo SCTP, as restrições de segurança também devem ser aplicadas a esses dois protocolos. Portas que não oferecem serviços públicos fixados ao produto ou serviço da IoT não devem estar acessíveis.

É preciso garantir que a filtragem de entrada e saída seja gerenciada sempre que possível. A filtragem de entrada interrompe uma série de ataques, ao passo em que qualquer ataque contra um serviço publicamente acessível ainda pode resultar em um comprometimento do ecossistema de serviços. A filtragem de saída é imperativa, neste ponto, para garantir que um componente comprometido do ecossistema de serviço não possa ser usado por um hacker para se mover lateralmente dentro do ecossistema. Além disso, a filtragem de saída ajuda a dificultar a possibilidade dos invasores de exfiltrar dados críticos do ecossistema para servidores controlados pelo hacke, deixando mais tempo para os administradores identificarem e isolarem o invasor.

Várias organizações oferecem esses serviços em um modelo de API simples que pode ser introduzido em uma determinada tecnologia. Isso permite que a tecnologia seja usada com pouco esforço de engenharia, bastando se inscrever e configurar a aplicação no sistema do provedor de serviços. Consulte seu provedor de serviços para determinar a melhor maneira de implementar sua tecnologia de segurança em seu ambiente.

Considere usar o material das seguintes organizações para ajudar com esta recomendação:

- Amazon Best Practices for DDoS Resiliency:
  - https://d0.awsstatic.com/whitepapers/DDoS_White_Paper_June2015.pdf
- Arbor Networks DDoS Mitigation Best Practices:
  - https://www.arbornetworks.com/images/documents/Arbor%20Insights/AI_DDoSMitigation_EN2013.pdf
- Cisco DDoS Defence Guide:
  - http://www.cisco.com/web/about/security/intelligence/guide_ddos_defense.html

5.4.1 Risco

A infraestrutura segura para serviços e aplicações voltados para o público é imperativa devido à natureza volátil da internet. Ataques aleatórios de DDoS ocorrem com frequência. Os “serviços DDoS” podem ser adquiridos no “mercado underground” por centenas de dólares. Assim, hackers da empresa ou de seu cliente não serão os únicos autores de tais ataques. Ataques aleatórios podem ocorrer apenas para se averiguar se é possível derrubar o sistema. É melhor estar preparado contra esses ataques para garantir que os serviços críticos de IoT não sejam inesperadamente derrubados. A disponibilidade é um componente crítico de um produto ou serviço da IoT.

5.5 Defina um modelo de armazenamento persistente

Os ambientes de aplicação na computação moderna costumam ser efêmeros, como sistemas baseados em conteiners ou ambientes de nuvem. Como resultado, o armazenamento atribuído a esses sistemas não é grande o suficiente, nem desenvolvido para ser disponibilizado a longo prazo, para que a aplicação use essas tecnologias como armazenamento persistente. Além disso, esses sistemas podem ser definidos como entidades sob demanda e podem não ter qualquer semelhança de centralização. Em outras palavras, não há como os outros sistemas definirem qual sistema possui armazenamento suficiente para uso persistente.

Por esse motivo, os sistemas de armazenamento central são imperativos e devem ser cuidadosamente protegidos. Como os sistemas de armazenamento devem ser disponibilizados para qualquer sistema temporário nesse tipo de ambiente, qualquer servidor ou serviço de vida curta que seja comprometido terá acesso a uma entidade de armazenamento persistente (ou camada) usada por muitos outros servidores ou serviços. Geralmente, isso é uma maneira eficaz de os hackers comprometerem lateralmente (ou possivelmente, verticalmente) qualquer rede específica.

Para restringir isso, cada servidor ou serviço deve ter acesso ao armazenamento persistente, mas deve armazenar informações com base na aplicação que representa e, mais importante, no endpoint, no parceiro ou no usuário único em nome do qual a aplicação está agindo. A última parte deste ponto é o ponto mais essencial, pois o requerimento de acesso de armazenamento persistente em nome de uma determinada identidade limita o acesso aos dados por parte do servidor ou serviço de curta duração.

Em outras palavras, um hacker que comprometeu o sistema de curta duração pode afetar apenas os dados armazenados em nome da identidade vinculada ao mesmo sistema de vida curta. Se esse sistema tiver acesso somente a uma única identidade, o hacker não poderá usar o comprometimento desse sistema para se mover lateralmente para outras contas. Eles ficarão restritos a acessar informações para essa única identidade. Esse é o limite da possibilidade do hacker de aproveitar a vulnerabilidade para uma exploração significativa do sistema.

5.5.1 Risk

Caso um modelo seguro de armazenamento persistente não seja definido, não haverá arquitetura que imponha atributos únicos por usuário para que sejam separados com segurança de outros ativos. Como possível resultado, qualquer comprometimento de um token que conceda a um hacker acesso a um dispositivo de armazenamento pode resultar

no comprometimento dos dados de vários usuários. No entanto, um modelo de armazenamento persistente pode isolar o comprometimento de um único usuário ou de uma única tecnologia de armazenamento com dados criptografados. Em ambos os casos, o escopo do comprometimento é significativamente reduzido, concedendo à organização mais tempo para reagir e combater a ameaça tanto para os usuários quanto para a empresa.

5.6 Defina um modelo de administração

Cada sistema deve ser acessível pela administração para solucionar problemas e diagnosticar falhas das aplicações. Isso pode ser desafiador em ambientes nos quais os serviços ou servidores são de curta duração, se um modelo administrativo não for suficientemente desenvolvido.

Para atingir esse objetivo, identifique como a equipe administrativa se comunicará com cada sistema em cada camada. Deve haver limites de autenticação, como VPNs, que separam sistemas diferentes uns dos outros. É necessário garantir que a equipe administrativa precise se autenticar em cada camada.

Além disso, é preciso identificar como o administrador irá interagir com o sistema. O sistema pode ser instantâneo, semelhante a uma máquina virtual? Um determinado terminal é usado? Um Secure Shell (SSH) remoto é usado para interagir com o sistema? Existem APIs para monitoramento e análise de métricas do sistema, como uso da CPU, uso do disco e uso da rede? Essas métricas podem ser usadas para solucionar problemas ou identificar anomalias?

Independentemente do modelo, há certas coisas que devem ser definidas:

- Como os administradores serão autenticados no ambiente
- Como administradores autenticados podem ser atribuídos a identidades físicas:
  - Uso da autenticação de dois fatores (2FA)
- Como os snapshots de sistemas podem ser obtidos
- Como as mudanças podem ser feitas e devem ser rastreadas

5.6.1 Risco

Ambientes que não possuem um caminho para acesso administrativo bem arquitetado geralmente acabam usando meios ad-hoc para acessar sistemas em produção. Isso geralmente leva a portas administrativas abertas à conectividade pública ou a serviços que oferecem diagnósticos, mas não estão restritas a serem usadas por terceiros. Um modelo administrativo claro reduz as vias potenciais que os invasores podem adotar para obter acesso privilegiado a recursos críticos de IoT.

5.7 Defina uma abordagem de registro e monitoramento de sistemas

Cada sistema deve ser monitorado para permitir que os administradores e a Tecnologia da Informação (TI) trabalhem para detectar e diagnosticar anomalias. O monitoramento deve ser realizado em várias dimensões. O monitoramento de rede no nível da infraestrutura, por exemplo, ajuda a diagnosticar ataques de aplicações ou DDoS contra componentes de rede. O monitoramento de camadas identifica se as aplicações específicas ou partes de infraestrutura podem ter sido violados. Finalmente, o monitoramento no nível do sistema

define se as aplicações individuais ou plataformas de aplicações estão sendo atacadas ou foram comprometidas.

Obviamente, isso exige vários níveis de monitoramento e consolida as informações em um recurso que pode ser transmitido para uma equipe de supervisão. Existem várias aplicações profissionais que fornecem essa tecnologia e convertem as métricas em sistemas visuais, utilizáveis por profissionais de TI e engenheiros de sistemas.

Anomalias que indicam comportamento hacker podem incluir, mas não estão limitadas a:

- Maior tráfego de rede
- Maior tráfego de rede em uma direção incomum (especialmente saída)
- Tráfego de rede de saída de um recurso que não deveria precisar de saída
- Utilização anormal da CPU
- Utilização de GPUs para sistemas sem interface visual, mas com uma GPU como parte da CPU
- Utilização de disco ou armazenamento de rede
- Mudanças anormais no tempo do sistema em um host particular

Embora os sistemas de monitoramento para anomalias de captura estejam prontamente disponíveis, o contexto pode ser específico para a aplicação ou a infraestrutura usada pela organização. Consulte o fornecedor do sistema de monitoramento para determinar como capturar e interpretar métricas de uma maneira que seja mais eficaz para a implementação específica.

Níveis separados podem ter diferenciais nas anomalias indicando um ataque ou comprometimento. Avalie quais são esses indicadores para cada nível.

Considere o material das seguintes organizações como suporte a esta recomendação

- Amazon EC2 Monitoring Documentation
  - http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html
- Google Cloud Monitoring
  - https://cloud.google.com/monitoring/
- Microsoft Azure Monitoring
  - https://azure.microsoft.com/en-us/documentation/articles/best-practices-monitoring/
- DigitalOcean Monitoring Tutorials (General)
  - https://www.digitalocean.com/community/tags/monitoring?type=tutorials

5.7.1 Risco

A tecnologia de monitoramento de sistemas é um atributo-chave do modelo de segurança da IoT. Sem monitoramento, não há como determinar se uma vulnerabilidade foi encontrada em componentes críticos do serviço. O monitoramento permite aos administradores diagnosticar rapidamente os pontos problemáticos no serviço e na infraestrutura e podem ajudar a diferenciar incidentes de segurança de bugs de software.

5.8 Defina um modelo de resposta a incidentes de segurança em computadores

Não é suficiente detectar um comprometimento potencial ou um ataque contínuo. A organização deve ser capaz de reagir e combater o ataque. A limpeza ou o desligamento não será suficiente para o sistema comprometido. A organização deve ser capaz de diagnosticar a origem do comprometimento, corrigir o sistema e implantar o patch em toda a infraestrutura existente.

Isso pode ser difícil se for usado um ambiente baseado em conteiner no qual aplicações clonadas estão sendo executadas com uma configuração vulnerável. O sistema de aplicações deve ser capaz de detectar um evento de “reinicialização” ou “atualização”, em que a conexão da aplicação é transmitida de forma eloquente para outro sistema na nuvem ou o usuário é desconectado à força para permitir a atualização.

Entretanto, independentemente do modelo de execução, a equipe de engenharia deve ser capaz de capturar métricas de uma maneira que permita a análise forense. Essas políticas e procedimentos devem ser estabelecidos e aprovados pela equipe legal (e possivelmente pela equipe de seguros) para validar se as informações estão contidas de forma adequada para os responsáveis pela aplicação da lei. A conformidade ajudará a garantir que a empresa não esteja apenas cumprindo as leis locais e federais, mas esteja demonstrando um comprometimento que pode ser usado em juízo.

Depois que amostras são capturadas, cada aspecto do sistema como um todo deve ser avaliado quanto a logs, métricas e outros dados que possam corroborar com o evento em questão. Todos esses dados devem ser capturados e armazenados em um sistema seguro para revisão legal.

Considere o uso do material das seguintes organizações como suporte a esta recomendação:

- CERT Recommendations for Creating a CSIRT
    - http://www.cert.org/incident-management/products-services/creating-a-csirt.cfm

5.8.1 Risco

Organizações que não têm um modelo de resposta a incidentes de segurança levarão muito mais tempo para organizar seus recursos, identificar sistemas comprometidos, colocar esses sistemas em quarentena e revisá-los para obter informações. Isso também diminui significativamente os esforços para corrigir e restaurar um determinado sistema. Esse despreparo confere aos hackers uma grande oportunidade para buscar um comprometimento lateral ou vertical dentro de um determinado ambiente. Como resultado, um comprometimento significativamente maior pode ocorrer devido ao aumento do tempo de resposta. As organizações devem estar preparadas para responder a um incidente quase que imediatamente para reduzir a quantidade de tempo que um hacker tem para controlar partes críticas do serviço.

5.9 Defina um modelo de restauração

A recuperação deve ocorrer independentemente de um usuário ou aplicação ser afetado devido a um comprometimento de segurança ou uma falha de hardware. Um procedimento

deve ser implementado para recuperação de informações e capacidade na camada da aplicação. O procedimento deve ser adaptado ao contexto de cada aplicação e camada.

Se uma aplicação reúne informações sobre a saída (output) de uma ação específica de um endpoint e há uma falha de armazenamento que proíba a aplicação de solidificar a saída desses dados no armazenamento persistente, a aplicação poderá:

- Tentar armazenar novamente até conseguir (pode ser infinito)
- Realizar um número limitado de tentativas de armazenamento até o limiar de sucesso ou falha
- Falha imediata, potencialmente perdendo as métricas
- Solicitar os mesmos dados novamente ao endpoint (poderão nunca estar disponíveis)

O método mais adequado para a aplicação e requisito negocial deve ser escolhido. A escolha, novamente, dependerá do contexto da aplicação, e pode não ser fácil modelar fora de um determinado sistema.

Envolva a liderança da engenharia e do negócio para determinar como uma aplicação comprometida ou com falha deve ser restaurado, especialmente no contexto da atividade do usuário.

Para sistemas que foram comprovadamente comprometidos por um hacker, deve existir um modelo para validar se a aplicação ou sistema foi suficientemente corrigido antes da restauração. Sem essa política e conjunto de procedimentos definidos, um sistema vulnerável pode simplesmente ser reimplantado no ecossistema de serviços, facilitando comprometimentos futuros.

5.9.1 Risco

Os modelos de restauração garantem que as informações, aplicações e configurações sejam restaurados corretamente. Sem um modelo de restauração, a equipe pode redirecionar inadvertidamente subsistemas vulneráveis para servidores ou infraestrutura. Além disso, os dados corrompidos, e possivelmente manipulados por um hacker em uma base de dados ou ambiente de armazenamento, poderiam ser replicados em vários sistemas, propagando malware involuntariamente ou simplesmente alterando dados. Processos de restauração reduzem a possibilidade de os hackers abusarem de falhas na restauração de um incidente, o que encarece um evento já caro.

5.10 Defina um modelo de sunsetting

Todo sistema implantado por uma organização e toda camada usado tem uma vida útil. Ainda que o mesmo produto ou serviço seja implantado pela organização há décadas, as tecnologias usadas para rodar esse produto ou serviço serão alteradas. Portanto, não deve haver apenas um plano para desenvolver e implementar o produto ou serviço, deve haver um plano para descontinuar (ou sunset) esse produto ou serviço.

Este processo ajuda a garantir que todas as tecnologias sejam invalidadas e descontinuadas de tal maneira que um hacker não possa assumir a identidade ou usar as capacidades da tecnologia dada. Como exemplo, pode-se considerar um simples caso do domínio para um produto específico após a aquisição de uma empresa por uma controladora. Se o produto for renomeado e o domínio for migrado para o domínio da

empresa controladora, um hacker poderá se apropriar do domínio agora extinto. Se o hacker puder emitir certificados criptográficos para esse domínio e ainda interagir com a tecnologia implantada nesse domínio antigo, haverá uma falha significativa na segurança causada pela falta de procedimento no sunsetting desse produto ou serviço.

Cada tecnologia usada na arquitetura, implementação e gerenciamento de um determinado produto ou serviço deve, então, ser catalogada e avaliada quanto à sua usabilidade. Uma vez que a tecnologia não seja mais utilizável, ela pode ser desativada de acordo com seu modelo. Isso permite que engenheiros e líderes de empresas migrem a tecnologia para um conjunto de inovações mais adequado, sem brechas nas plataformas subjacentes. Também garante que um produto não mais oferecido a parceiros e usuários terminará sua vida útil sem possibilidade de exploração pelos hackers após o fechamento do negócio.

5.10.1 Risco

A falta de um processo de sunsetting pode resultar no comprometimento de endpoints e serviços por concorrentes ou hackers. Isso é possível legalmente porque, se uma organização liberar acesso a certos objetos, como nomes de domínio, números de telefone e outros serviços renováveis, um hacker ou concorrente tem o direito de adquirir esses objetos, mesmo que pareça ser antiético. Isso pode expor dispositivos ou serviços a abusos inescrupulosos ou comportamento mal-intencionado.

5.11 Defina um conjunto de classificações de segurança

Para gerenciar adequadamente as interações com organizações parceiras de forma eficaz, as classificações de segurança devem ser definidas. Isso definirá o tom não apenas da política organizacional interna de segurança de dados, mas ajudará a definir o nível de segurança que as organizações parceiras aplicam aos dados da empresa, seus próprios dados e dados do cliente.

Embora esse processo deva ser investigado e adaptado à organização, a maioria das políticas de classificação de segurança de dados deve começar com as seguintes classes:

- Público - Qualquer entidade que tenha acesso concedido
- Classificado - o usuário deve autorizar a liberação
- Confidencial - dados específicos do usuário
- Top secret- Dados específicos da organização; nunca serão liberados

Depois de definir as classes básicas, a organização deve avaliar como cada classe de segurança deve ser atribuída a uma classe de dados. Em outras palavras, avaliar como a classificação deve ser usada na prática, não apenas na teoria. Determine quais políticas e procedimentos devem ser implantados a partir de uma perspectiva de negócios e de engenharia.
Isso permitirá que a organização não apenas crie uma política técnica, mas coloque em vigor uma política comercial que suporte os requisitos técnicos. Dessa forma, a equipe de engenharia pode entregar esses requisitos a parceiros e organizações internas que tentam quebrar a política, intencionalmente ou não, com mais facilidade.

Depois que as classificações de segurança forem padronizadas, é importante avaliar como o modelo de classificação de segurança pode ser afetado pelos requisitos de privacidade da empresa e de seus usuários. A organização deve reservar um tempo para aplicar um

modelo de privacidade às classificações de segurança, para dar sentido aos dados dos usuários e ajudar a proteger sua privacidade, caso um parceiro deseje acessar recursos específicos que possam colocar os usuários em risco de exposição. Ao considerar a privacidade no contexto das classificações de segurança, os parceiros precisarão buscar a aprovação da liderança da empresa e dos usuários quando quiserem adquirir certos tipos de dados relevantes para a privacidade. Os usuários devem ter a opção de proteger seus dados e devem poder limitar a exposição de seus dados a terceiros.

5.11.1 Risco

A classificação de modelos de segurança é imperativa para arquitetar soluções que usem a segurança de maneira efetiva. Para que as informações sejam protegidas, elas devem ser quantificadas para que os controles apropriados possam ser formulados com base nas políticas e procedimentos correspondentes. Sem esses modelos, os engenheiros tendem a implementar a segurança ou de forma intensa ou sequer implementar segurança, dependendo da percepção dos riscos envolvidos. Toda a equipe, incluindo engenheiros e líderes da empresa, deve identificar o que os dados significam para a empresa e como esses devem ser protegidos dentro de um conjunto apropriado de controles com boa relação custo-benefício.

5.12 Defina classificações para conjuntos de tipos de dados

Depois de definir as classificações de segurança, a organização deve definir os tipos de dados a serem usados pelo produto ou serviço IoT. Isso permitirá que a organização defina claramente quais tipos de informações são adquiridos, gerados e disseminados para os pares no sistema de IoT e como a organização deve tratar esses tipos de dados. Esses dados fornecerão contexto e valor aos componentes gerais usados em todo o ambiente de IoT.

Embora este documento não tente modelar todas as variações de dados que possam ser relevantes para uma organização específica, alguns tipos incluem:

- Usuários
- Ações
- Imagens
- Documentos editáveis
- Informação pessoalmente identificável
- Informação de saúde protegida

Uma informação pode ser atribuída a um ou mais tipos. Mas os dados em si devem ser atribuídos apenas a uma classe de segurança. Enquanto o tipo identifica o que os dados representam e como eles devem ser processados, a classe de segurança representará como, onde e quando as informações podem ser usadas e com quem elas podem ser compartilhadas.

Definir os vários tipos de dados e atribuir classificações a eles é um processo longo. Isso define um padrão organizacional para a empresa e permite que a equipe de engenharia execute controles técnicos em torno dos dados e de suas classificações, o que ajuda muito as equipes de engenharia e liderança da empresa mais tarde, ao negociar com parceiros sobre como os dados podem ser compartilhados e processados.

5.12.1 Risco

Assim como acontece com as classificações de segurança, os controles não podem ser implementados em torno dos dados sem quantificar quais são esses dados e qual o relacionamento desses dados com o negócio. Essas classes definem como as informações devem ser usadas no sistema e quais proteções devem ser aplicadas aos dados para manter uma postura de segurança apropriada. Sem essas classes, os engenheiros tendem a aplicar medidas de segurança muito rigorosas ou muito fracas. As medidas de segurança devem ser acordadas entre a equipe de engenharia e a liderança, para equilibrar os controles com a importância dos dados para o negócio.

6 Recomendações de alta prioridade

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

6.1 Defina um modelo de autorização transparente

Embora o modelo de privacidade lide com a maneira como as informações do usuário são oferecidas aos parceiros, o modelo de autorização define como a empresa ou os parceiros agirão em nome de um usuário. Isso, por exemplo, seria útil para um sistema de automação residencial em que as métricas de um parceiro poderiam otimizar o uso de aquecimento ou refrigeração em uma determinada casa. O modelo de autorização concederia ao parceiro a capacidade de alterar os controles de aquecimento ou refrigeração da casa desse usuário quando determinadas métricas fossem por ele detectadas.

Para isso, tenha uma interface gráfica similar que descreva recursos granulares de autorização e como eles serão distribuídos aos parceiros. Permita que o usuário conceda ou revogue o acesso a determinados recursos sob demanda. Certifique-se de que a revogação de recursos entre em vigor imediatamente para diminuir a possibilidade de comprometimento.

O sistema deve ser fortemente monitorado para garantir que os parceiros não realizem ações que não têm permissão de executar. O controle granular do modelo de autorização deve permitir aos usuários configurar quando os parceiros terão acesso a determinados recursos e com que frequência. Atributos como esse devem melhorar o processo ao permitir que o usuário assuma o controle de seu sistema de um parceiro potencialmente abusivo ou comprometido (hackeado).

6.1.1 Risk

Sem um modelo de autorização, terceiros não terão acesso reservado aos recursos de um usuário. Isso pode permitir que um hacker ou um parceiro hackeado adquira acesso total à tecnologia ou aos dados de um usuário. Ao criar um modelo de autorização, o acesso é restrito apenas aos atributos permitidos pelo usuário. Isso permite que este tenha maior controle sobre quais recursos e dados são disponibilizados a terceiros e reduz o risco do provedor de serviços de IoT, diminuindo a possibilidade generalizada de comprometimento.

6.2 Gerencie a arquitetura criptográfica

Toda a tecnologia implantada em um ambiente de IoT deve usar criptografia, independentemente de a tecnologia ser um endpoint rudimentar de baixa potência ou um serviço de nuvem robusto. Para implementar corretamente a segurança em um produto ou serviço de IoT, a criptografia usada deve ser desenvolvida, gerenciada e ajustada para atender às mudanças de especificações ao longo do tempo.

A equipe de engenharia deve identificar se:

- Seus algoritmos criptográficos foram depreciados aprovados
- Está utilizando chaves criptográficas com comprimentos adequados

- Algoritmos de hash estão sujeitos a ataques de colisão
- Um gerador forte de números aleatórios é usado
- As mensagens são suficientemente preenchidas com dados aleatórios
- Protocolos criptográficos, como o TLS, estão atualizados com as melhores práticas
- Conceitos centrados em privacidade, como sigilo de encaminhamento, são usados
- Senhas de texto simples ou números de pins são enviados pela rede
- Um algoritmo criptográfico customizado foi usado

Cada um dos pontos acima, além de outras ações, são importantes para manter uma arquitetura criptográfica de alta qualidade no produto ou serviço de IoT. O sucesso na implantação de uma solução criptográfica está intimamente ligado à capacidade da equipe de engenharia de aproveitar as soluções de criptografia mais resilientes para implantar atualizações em tecnologias que usam soluções menos resilientes.

Por exemplo, recentemente foi descoberto que o algoritmo RC4 possui falhas significativas de segurança. Se um patch puder ser distribuído com segurança para os clientes configurados para usar o RC4, que substitui o RC4 pelo AES-256, o RC4 se torna uma preocupação menor. Se a autenticação mútua for executada usando uma tecnologia mais resiliente, como a troca de chaves Ephemeral Diffie Hellman e chaves assimétricas, ou um token de segurança UICC, a correção poderá ser verificada sem o uso do algoritmo criptográfico vulnerável.

As senhas e os pins usados por um usuário ou endpoint nunca devem ser passados pela rede em texto sem formatação, mesmo se o canal de comunicação for protegido por meio de criptografia. Em vez disso, o hash criptográfico da senha ou pin deve ser usado para garantir que qualquer configuração incorreta no túnel criptográfico não exponha a própria senha. O hash deve ser gerado pela senha e pelo menos um token único e descartável. Embora seja comum o token ser retirado da sessão de rede, é mais seguro obter o valor de um rolling code (código evolutivo) armazenado no endpoint e na infraestrutura de serviço. Dessa forma, um hacker com uma posição de privilégio na rede não pode propagar o hash com valores benéficos, o que pode resultar em um ataque de assinatura forçada.

Algoritmos criptográficos customizados (algoritmos criados internamente) nunca devem ser usados. Sempre use algoritmos desenvolvidos por criptógrafos e recomendados por organizações de supervisão especializadas em segurança criptográfica. Sempre evite o uso de algoritmos mal desenvolvidos, algoritmos reprovados ou compactação, binário-para-texto ou outros algoritmos comumente confundidos com algoritmos criptográficos, como LZO, base64, ROT13 e XOR.

Revise os seguintes guias e referências para obter mais informações sobre este tópico:

- ISO 18033-1: 2015 - Algoritmos de Criptografia
- ISO 18033-2: 2015 - Cifras Assimétricas
- ISO 18033-3: 2015 - Cifras de bloco
- www.owasp.org/index.php/Guide_to_Cryptography
- csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf
- csrc.nist.gov/groups/ST/toolkit/key_management.html

6.2.1 Risco

A implantação adequada de uma solução com uma arquitetura criptográfica garante que os algoritmos, protocolos e segredos utilizados estejam dentro das recomendações atuais. Além disso, as recomendações mudam com o tempo. Sem uma arquitetura criptográfica, será mais difícil identificar todas as tecnologias que foram reprovadas, o que cria uma oportunidade para falhas na segurança.

6.3 Defina um modelo de comunicação

Cada sistema no ecossistema de serviços deve ser capaz de autenticação mútua. Nenhuma plataforma de computação dentro desse ecossistema deve ser acessível a usuários públicos anônimos. Cada endpoint, parceiro ou usuário se comunicará com o ecossistema de serviços por meio de tecnologias que exigem autenticação mútua. Como os serviços que compõem a interface do usuário geralmente são implantados e gerenciados em um ambiente separado, a interface publicamente acessível deve estar confinada a esse espaço. O ecossistema de serviços, no entanto, compreende o conjunto de todo o sistema usado para implantar o serviço em todos os recursos autenticados.

Isso inclui endpoints que ainda não foram provisionados pelo sistema, já que o processo de fabricação e customização de hardware deve configurá-lo de forma adequada o suficiente para que possa ser autenticado como um recurso implementado comercialmente.

Portanto, o modelo de comunicação deve fornecer:

- Autenticação mútua
- Confidencialidade
- Integridade

Para atingir esse objetivo, o modelo de comunicação também deve fornecer:

- Uma raiz de confiança centralizada ou, alternativamente, uma raiz de confiança descentralizada
- Provisionamento de identidade e revogação
- Perfect Forward Secrecy

Uma raiz de confiança deve ser usada para garantir que cada entidade no modelo de comunicação seja autorizada pela mesma organização que o par. Isso ajuda a garantir que todas as entidades tenham sido provisionadas e autorizadas por uma organização central. A tecnologia usada para garantir essa raiz de confiança pode ser centralizada (semelhante aos certificados TLS) ou descentralizada (semelhante aos modelos de IoT baseados em blockchain Bitcoin, por exemplo, projeto ADEPT da IBM / Samsung, Tilepay e outros). Independentemente disso, uma empresa central deve ser a proprietária do modelo e proteger o sistema de provisionamento.

Provisionamento e revogação devem fazer parte do modelo de comunicação para ajudar a garantir que qualquer segredo ou identidade comprometida possa ser removido do sistema com o mínimo de esforço. Tecnologias como o Online Certificate Security Protocol (OCSP) ajudam nesse processo.

O protocolo de comunicações deve empregar uma tecnologia que mitigue a possibilidade de comprometer as comunicações do passado. Isso é feito criando-se chaves criptográficas

assimétricas e efêmeras que são usadas para trocar um segredo de comunicação. Se um certificado for comprometido, o segredo efêmero não será. Isso garante que o armazenamento de mensagens criptografadas por um longo período de tempo não resultará, no futuro, em um hacker sendo capaz de descriptografá-las se o segredo do certificado estiver comprometido ou exposto.

O desafio da segurança das comunicações está na implementação e longevidade da tecnologia. Algoritmos de criptografia podem ser selecionados se são mantidos em um alto grau de confiança por entidades autoritativas, diminuindo a possibilidade de falha.

Bibliotecas e implementações de algoritmos desenvolvidas ou aprovadas por autoridades de engenharia devem ser usadas. Implementações customizadas de algoritmos não devem ser usadas. Isso diminui não apenas o trabalho da equipe de engenharia, mas também o potencial de um algoritmo ser enfraquecido criptograficamente por um sistema mal desenvolvido ou incorretamente implementado.

Considere o uso de material das seguintes organizações para ajudar com esta recomendação:

- Guia prático do CafeSoft Apache Mutual Authentication:
  - http://www.cafesoft.com/products/cams/ps/docs32/admin/ConfiguringApache2ForSSLTLSMutualAuthentication.html

6.3.1 Risco

A segurança das comunicações é a base da IoT. Sem ela não há garantia de que os dispositivos embutidos estejam se comunicando com os serviços de back-end corretos. Isso é imprescindível para serviços críticos que orientam, configuram e enviam comandos para dispositivos como telemática, dispositivos médicos e sistemas de controle industrial. Sem segurança das comunicações, não há garantias de que os comandos estão sendo enviados para o endpoint correto. Reforce a segurança das comunicações para garantir que as mensagens estão sendo enviadas e recebidas do par pretendido.

6.4 Use serviços de autenticação

Operadoras de rede, quando usadas como parceiros, permitem que os usuários sejam autenticados usando tokens específicos para a operadora de rede. Embora esses tokens, presentes no UICC da operadora de rede, autentiquem um usuário na camada de rede, eles não necessariamente autenticam o usuário na camada da aplicação. O uso das seguintes tecnologias pode facilitar a autenticação na rede:

- Arquitetura Genérica de Inicialização (3GPP TS 33.220)
- M2M SM (ETSI TS 102 921)

Avalie se a tecnologia de autenticação será significativa para fins de autenticação na camada da aplicação . Se o token puder ser usado como um armazenamento de segurança, determine se o dispositivo pode ser usado como uma camada de autenticação para o endpoint físico para construir uma TCB usando o token.

Embora muitas operadoras de rede imponham a autenticação baseada em rede, a concessão de acesso a essa API para autenticar os usuários ou os endpoints é uma

tecnologia relativamente nova. Avalie se a operadora de rede com a qual você está trabalhando possui uma experiência significativa nesse âmbito. Nesse caso, considere usar essa tecnologia como mais do que um token de autenticação de camada de rede, pois pode ser mais fácil utilizar uma única tecnologia de armazenamento de segurança em vez de várias tecnologias.

6.4.1 Risco

Quando os serviços de autenticação de rede incorporam âncoras de confiança, como o UICC, a não utilização desses serviços para proteger a camada de aplicações limitará a capacidade da aplicação de autenticar usuários com segurança e aumentará a despesa da plataforma endpoint subjacente. Isso aumenta o custo de implantação e também diminui as informações disponíveis para a organização da operadora de rede.

6.5 Disponibilize servidores quando possível

A disponibilização de servidor envolve definir, configurar, personalizar e implementar um servidor em um ambiente de produção. O processo de provisionamento, de uma perspectiva de serviço, garante que um servidor esteja com segurança reforçada e pronto para implementação em um ambiente potencialmente hostil.

Independentemente de o servidor estar implantado em uma infraestrutura de nuvem, em um provedor de hospedagem dedicado ou em um rack exclusivo da empresa, um servidor ficará vulnerável a ameaças internas e externas. O servidor deve ser protegido contra ataques antes de ser implantado na infraestrutura de serviço.

Para isso, identifique os serviços que devem estar acessíveis ao ambiente circundante. Defina se o ambiente em que o servidor viverá será público ou privado e o que isso significa no contexto de segurança. Determine se cada serviço em execução no servidor deve estar acessível ao público ou se somente os clientes autenticados devem se conectar ao serviço.

Avalie o ciclo de vida do sistema operacional que será executado no servidor. Determine como gerenciar adequadamente as atualizações de software para garantir que os patches de segurança sejam implantados rapidamente e comprometidos com os servidores que estão em produção. Avalie um modelo de reversão caso as atualizações falhem ou causem problemas inesperados nos serviços em produção, pois algumas atualizações de biblioteca ou aplicação podem resultar em efeitos colaterais indesejados.

Finalmente, avalie o modelo de sunsetting do servidor provisionado para determinar a maneira mais segura de remover ativos do sistema. Isso inclui logs do sistema que podem ser necessários para avaliar o comportamento do cliente ou anomalias no serviço.

Essa recomendação implica que um processo de gerenciamento de patches deve ser implementado pela organização para identificar serviços vulneráveis, implantar e monitorar o sucesso da implementação desses patches.

Revise o seguinte material sobre o gerenciamento de patches:

- http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

6.5.1 Risco

O provisionamento de servidores é uma parte imperativa da segurança de um ambiente de IoT. Sem isso, o controle da organização sobre a arquitetura do servidor será substancialmente enfraquecido. Isso pode resultar em falhas de segurança devido à falta de especificação de arquitetura. Sem uma especificação, a organização não pode avaliar se as tecnologias implantadas aderem às práticas atuais recomendadas. Além disso, o aprimoramento dessas tecnologias exigirá a investigação de cada sistema implantado para avaliar os deltas entre ativos implantados. Isso é ineficiente e uma grande preocupação no caso de uma atualização de segurança crítica precisar ser implantada. Se não houver consistência e nenhuma arquitetura para definir os serviços, não haverá maneira de rastrear facilmente quais sistemas requerem atenção imediata sem verificar manualmente cada um deles.

6.6 Defina um modelo de atualização

Atualizar um ambiente de execução, imagem de aplicação ou TCB é um processo desafiador. Considere o seguinte modelo de exemplo que simplifica o processo como um todo:

- Para cada camada da plataforma de execução, defina um recurso de rede, como uma URL única para a nova imagem da aplicação
- Gere uma chave de assinatura para cada camada específica
- Para todas as novas versões autorizadas em cada camada, gere uma imagem dessa camada
- Incluir metadados descrevendo a imagem (versão, timestamp, identidade etc.) na camada desta
- Assine a imagem da camada com a chave de assinatura
- Disponibilize a imagem, a assinatura e a chave pública, possivelmente por meio de recurso de rede único ou por meio de um serviço de atualização

Quando um novo sistema é implantado, ele deve:

- Para cada camada:
  - Recuperar a (s) versão (ões) a ser implantada(s)
  - Verificar criptograficamente a imagem
  - Implantar a camada de imagem no sistema

Nenhum segredo privado deve ser armazenado em qualquer camada da aplicação. Em vez disso, os segredos devem ser provisionados dinamicamente à medida que cada sistema é implantado, para personalizar cada sistema. Essas identidades devem ser revogadas quando o sistema é desativado, independentemente de sua duração.

Essa recomendação indica que um processo de gerenciamento de atualizações deve ser usado para manter serviços e tecnologias dentro da infraestrutura.

Verifique a seguinte documentação para mais informações:

- http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

6.6.1 Risco

Sem um modelo de atualizações bem definido, os serviços e aplicações correm risco de comprometer o uso desse procedimento. Hackers podem inserir aplicações customizadas no processo de atualização e implantar seu próprio software nos sistemas em nuvem e em outros servidores. Se a infraestrutura da segurança de comunicações não estiver protegida, isso pode ser facilmente executado simplesmente pela manipulação de serviços de rede, como o DNS (Domain Name Service, serviço de nomes de domínio). Ataques mais avançados contra roteamento, como ataques Border Gateway Protocol (BGP), foram implementados muitas vezes no passado para comprometer serviços inseguros.

6.7 Defina uma política de violação para dados expostos

Definir políticas e procedimentos para a classificação de dados não é suficiente. Também deve haver um modelo para detectar se os dados foram expostos por um parceiro. A organização deve ter um plano para avaliar se um parceiro esteve envolvido em práticas de negócios que violam os controles tecnológicos ou as políticas estabelecidas para proteger os dados e a privacidade do usuário.

Para conseguir isso, a equipe de engenharia deve definir tecnologias de monitoramento e registro que se apliquem às classificações de segurança, e não simplesmente aos dados do usuário. Isso permitirá que uma auditoria seja aplicável não apenas às informações, mas à classificação destas. Isso ajudará a organização a se defender no caso de informações do usuário serem expostas. A organização poderá mostrar que suas classificações de segurança e os controles técnicos implementados para gerenciar essas classes, gerenciaram, armazenaram e disseminaram os dados de acordo com a política.

É benéfico que a organização use a tecnologia de monitoramento e registro para comprovar quando um parceiro violou as regras das classificações de segurança. A liderança deve, nesse ponto, decidir se ele deve ou não estar sujeito a multas, dispensa ou outra consequência.

6.7.1 Risco

Sem uma política de violação, há poucas proteções legais para proteger a organização contra a responsabilidade legal por dados que foram expostos por terceiros. Se a empresa é a fonte dos dados expostos, ainda que um terceiro tenha perdido estes dados, mas a empresa é responsável pelos dados que entrega aos seus parceiros.

As políticas de vazamento garantem que os parceiros devem manter um nível de segurança adequado aos dados que estão sendo fornecidos. Qualquer violação dessa segurança ajuda a reduzir a responsabilidade legal do provedor de serviços de IoT, desde que este siga seus próprios requisitos de segurança. Então, cabe ao parceiro aderir à política.

Essas políticas devem ser revisadas por equipes legais e de seguros para garantir que os modelos reduzam, de fato, a responsabilidade legal da organização, aderindo a políticas e padrões de segurança rigorosos. Algumas empresas, devido à natureza dos produtos ou serviços que oferecem, não podem ficar isentas de regulamentos, estatutos legais ou outras questões.

6.8 Force a autenticação por meio do ecossistema de serviços

Uma interface de usuário nunca deve autenticar um usuário diretamente. O sistema deve sempre ser capaz de autenticar o usuário usando o serviço central disponível. A única exceção a essa regra é se uma aplicação em execução em um dispositivo móvel for protegida por uma senha local. Essa senha pode ser usada para acessar a aplicação local. No entanto, o acesso a serviços e recursos remotos deve ser verificado por um token de autenticação separado.

Embora, por questões de usabilidade, a equipe de engenharia possa optar por reduzir esses dois esquemas de autenticação a apenas um, caso o usuário receba informações suficientes que descrevam os riscos de utilizar esse método de autenticação. Esse método permitiria à senha da aplicação local de um usuário autenticado desencriptar um banco de dados local contendo um token de autenticação que funcionasse no serviço remoto. Esse modelo de autenticação em várias etapas pode ser suficiente para a maioria dos usuários.

Independentemente disso, o serviço de autenticação central deve primeiro autenticar o usuário para a aplicação local, depois aplicar políticas e procedimentos que determinem como esse token de autenticação pode ser usado e por qual período de tempo. As métricas também devem ser reunidas para determinar se o usuário migrou para uma plataforma de computação alternativa, mas está usando o mesmo token. Ou se o usuário migrou para outro local em um curto período de tempo, mas está usando o mesmo token. Dependendo do tipo e da velocidade do movimento, essas métricas podem indicar um possível comprometimento do token. Nesse ponto, o token deve ser invalidado e o usuário deve ser forçado a efetuar login novamente, possivelmente usando a autenticação multifator, quando aplicável.

6.8.1 Risco

Devido aos possíveis abusos em sistemas de endpoints, independentemente de quão segura a arquitetura possa ser, a autenticação de um usuário sem a confirmação de um sistema de back-end é sempre insegura. Isso pressupõe que o usuário não atualizou suas credenciais ou pode desmembrar credenciais em vários tipos de dispositivos. Isso é

ineficiente e pode abrir uma lacuna onde um dispositivo comprometido está usando uma versão mais antiga das credenciais do usuário.

6.9 Implemente validação de entrada

Todos os dados adquiridos de um endpoint, usuário, ou usuário presumido devem ser analisados quanto a anomalias. A rota de ataque mais fácil para um hacker é sempre abusar da entrada da aplicação web nos serviços que compõem a interface com o usuário. Isso ocorre porque essa tecnologia deve processar informações dinamicamente, com base nas variações de localidade, codificações e outros parâmetros que mudam de usuário para usuário. Usuários qualificados podem manipular certos atributos de codificações para causar efeitos colaterais inesperados e benéficos para um hacker em diferentes camadas dos subsistemas de processamento.

Por exemplo, um ataque fascinante é o que inclui a codificação de um byte nulo em mensagens processadas como strings por linguagens de alto nível. Algumas linguagens de alto nível aceitam bytes nulos como parte de uma cadeia binária, em vez de vê-la como um delimitador. Quando esta cadeia binária é transmitida para bibliotecas em nível mais baixo, o byte nulo incorporado é interpretado como um delimitador de cadeia, truncando a cadeia para significar algo totalmente diferente do que a aplicação interpretou sobre a cadeia. No passado, essa era uma maneira inteligente de acessar recursos do sistema de arquivos que, de outra forma, estariam indisponíveis para um usuário específico.

Embora exista um número infinito de variações para um input mal-intencionado, os engenheiros não precisam testar todos os casos possíveis. Em vez disso, o processo é relativamente simples:

- Identificar como os dados devem ser usados internamente
- Aplicar uma política em torno de quais tipos de codificações e caracteres aderem ao modelo de uso interno
- Desenvolver uma API que analise os dados de acordo com essa política
- Elencar uma exceção quando forem identificados dados que quebrem o modelo
- Registrar o evento internamente com metadados sobre a sessão para ajudar a detectar o comportamento do hacker

Todos os dados armazenados no sistema devem primeiro ser processados e integrados em um modelo estático. Uma técnica eficaz para isso é simplesmente codificar todos os dados com o algoritmo base64 e, em seguida, colocá-lo em um banco de dados. Isso garante que os dados não possam manipular o banco de dados.

6.9.1 Risco

Os sistemas que não empregam a validação de entrada estão sujeitos a uma série de possíveis ataques, incluindo problemas mencionados no OWASP Top Ten, como o SQL Injection (SQLi), e até mesmo ataques de execução remota de código. Como o conjunto de possíveis abusos é muito grande, o risco não pode ser totalmente quantificado aqui. A validação de entrada é um atributo crítico de qualquer aplicação segura, seja um serviço de nuvem ou uma aplicação em execução em um endpoint.

6.10 Implemente filtragem de saída

A filtragem de saída é o complemento para a validação de entrada. Esse processo não apenas protege a camada de apresentação da manipulação de hackers, mas também impede que o sistema forneça informações para um usuário que deve ser considerado privilegiado.

No primeiro caso, todos os dados a serem renderizados pela camada de apresentação devem ser avaliados antes de saírem da camada de serviço. Isso garantirá que os dados codificados na camada de apresentação, por exemplo, em mensagens JSON ou JavaScript codificado, não contenham formatação que possa interromper ou invalidar a apresentação dos dados. Isso significa que qualquer caractere armazenado no sistema que, se renderizado, pode quebrar o modelo de apresentação, deve ser filtrado ou codificado de tal forma que não altere essa apresentação de maneiras inesperadas.

Uma metodologia para remediar esse problema é filtrar caracteres restritos, impondo uma codificação em todos os caracteres para que a apresentação desses caracteres não altere a GUI (os caracteres não são interpretados por um mecanismo de renderização, como códigos de controle) ou simplesmente não exibir a mensagem. Embora qualquer um desses métodos funcione, alguns são mais apropriados em determinadas aplicações. Revendo o exemplo do fórum de mensagens, seria igualmente ruim se um hacker conseguisse colocar scripts que outros usuários possam copiar e executar sem saber o que estão fazendo. Portanto, em vez de apenas renderizar as informações de maneira que não insiram HTML ou outros scripts na camada de apresentação, as informações devem ser removidas para que outros usuários não sejam afetados.

No caso em que os dados não devem ser devolvidos ao usuário, isso não está relacionado aos dados armazenados e processados por um hacker. Em vez disso, esse problema está relacionado à manifestação de dados impróprios para consumo público e deve ser reservado para administradores e engenheiros. Por exemplo, se um erro interno for gerado no processamento de informações, esse erro não deverá ser renderizado, com seus dados de depuração completos, para o usuário. Isso pode permitir que um usuário identifique e instrua um bug para fins de exploração de pontos fracos na aplicação. Esta informação deve ser internamente registrada, e um erro genérico deve ser levantado para o usuário que não fornece contexto suficiente para o usuário abusar do bug. Mesmo que o usuário possa reproduzir o bug, o usuário não deve ser capaz de avaliar um diferencial na saída da aplicação que indica melhorias na metodologia de violação.

6.10.1 Risco

A validação de saída é um atributo crítico da segurança da IoT. Sistemas que não executam validação de saída arriscam a exposição de dados críticos do usuário, dados relacionados à privacidade, dados de diagnóstico, mensagens de erro detalhadas e muito mais. Essas mensagens podem ser usadas para expor informações do usuário ou podem ser usadas para criar uma violação sólida contra um serviço em rede.

6.11 Reforce a política de senhas fortes

É imperativo que todos os sistemas de autenticação apliquem senhas fortes, necessárias para a autenticação do usuário. A complexidade das senhas tem sido uma batalha constante entre pesquisadores de segurança da informação, engenheiros e executivos de

negócios. Os executivos geralmente querem que os usuários consigam lembrar facilmente suas senhas. Os engenheiros precisam reduzir a complexidade das interfaces, especialmente para desenvolvedores da camada de apresentação. Os pesquisadores de segurança da informação frequentemente superestimam a habilidade de um hacker e pressionam para uma complexidade desnecessária em uma determinada tecnologia.

A resposta, no entanto, está em algum lugar entre os requisitos de cada grupo. As senhas devem ser forçosamente longas, mas não devem ser complexas. Enquanto senhas de oito caracteres costumavam ser a norma, e alguns sistemas ainda permitam 6 caracteres, mesmo no momento em que este documento está sendo escrito, o comprimento da senha deve ser retirado do último padrão de melhores práticas, mas provavelmente excederá os 8 caracteres. Ao impor um comprimento de senha mais longo, o requisito de complexidade é reduzido. Em vez de impor sortimentos bizarros de vários conjuntos de caracteres, o usuário pode simplesmente lembrar uma frase. Como eles podem optar por utilizar espaços em branco, letras maiúsculas, números e pontuação, a complexidade automaticamente cresce para qualquer hacker que aplique força-bruta.

Não esqueçamos que, normalmente, há quatro maneiras pelas quais um hacker compromete uma senha:

- Invadindo o banco de dados de senhas e quebrar senhas individuais
- Atacando com força bruta o serviço de autenticação de aplicações
- Instalando malwares
- Usando senhas codificadas ou padrão

Impor senhas longas ajuda a diminuir o primeiro risco. Mas a segurança na camada do ecossistema de serviços é muito mais benéfica. O hacker não conseguirá capturar o banco de dados de senhas, o que nos leva ao segundo ponto.

O uso de força bruta sobre as senhas da aplicação é, então, a maneira mais eficaz para um hacker comprometer senhas. Essa possibilidade é significativamente reduzida por um serviço de autenticação adequadamente desenvolvido. Se uma senha incorreta foi percebida, o sistema deve começar automaticamente a aumentar o delay necessário entre as tentativas. Em seguida, deve ser definido um limite ao número total de tentativas. Se o hacker atingir esse limite, a conta deverá ser bloqueada e a autenticação de dois fatores, ou outro modelo, deverá ser usada para que o usuário desbloqueie e verifique sua conta. Esse tipo de segurança reduz substancialmente a vantagem de um ataque baseado em rede, o que nos leva ao ponto final.

O malware no sistema do cliente é algo que deve ser tratado pela plataforma de computação ou pelo usuário que instala a apropriada tecnologia de combate. Isso normalmente não é algo que pode ser protegido pela própria aplicação. Como há pouco ou nada que a aplicação possa fazer para combater esse risco, além de impor a 2FA, o engenheiro de aplicações terá reduzido satisfatoriamente a área de ameaça dos ataques de senha contra o sistema de autenticação se esse for o *único caminho viável de ação* de um invasor.

Deve-se notar, no entanto, que a recompensa pela implementação desta recomendação *não é alta*. Isso ocorre porque não importa quais tecnologias são usadas para reduzir a possibilidade de ataques à autenticação de senhas, pois elas são, essencialmente, um

recurso intangível. Elas não são tokens físicos que só podem ser capturados por um único indivíduo. Pelo contrário, é um objeto abstrato que pode ser copiado infinitamente por meio de sistemas de computação e observação visual. Portanto, são uma fonte significativamente fraca de autenticação que, de maneira alguma, identificam adequadamente um usuário em particular. Portanto, as próprias senhas são uma fragilidade e qualquer tecnologia que use senhas está sujeita aos riscos inerentes a elas.

As senhas nunca devem ser codificadas no sistema. Para endpoints, chaves criptográficas únicas devem ser geradas. Consulte o documento sobre endpoints para obter mais informações sobre seu provisionamento. Para serviços e interfaces de usuário, a senha deve ser definida pelo usuário quando se registra. A senha, nesse momento, deve obedecer a requisitos fortes de segurança. Nunca permita que um usuário utilize uma senha padrão, fraca ou mal desenvolvida.

Certifique-se de que o usuário sempre tenha a capacidade de alterar sua senha a qualquer momento. Imponha requisitos fortes de autenticação e segurança de comunicações para que o usuário altere sua senha. Sempre que possível, ative a autenticação de dois fatores (2FA) para verificar a identidade do usuário antes de permitir uma alteração de senha. Sempre forçar um usuário a digitar novamente sua senha original ao enviar uma nova senha ao sistema. Isso garante que outro usuário não tenha usurpado uma aplicação web aberta aproveitando um notebook desbloqueado ou roubado um token de sessão da aplicação web.

6.11.1 Risco

Sistemas que não impõem controles adequados de senha arriscam a possibilidade de os hackers adivinharem facilmente as senhas dos usuários.

6.12 Defina autenticação e autorização na camada da aplicação

Enquanto a raiz organizacional de confiança e seus serviços definirão as tecnologias de autenticação que protegem a camada de comunicação da rede, as tecnologias de autorização de usuário, administração e parceiros devem ser configuradas separadamente. Enquanto os canais de comunicação dessas entidades são protegidos pela raiz organizacional de confiança, suas ações e identidades devem ser autenticadas usando um sistema separado.

Geralmente, essa autenticação da camada de aplicação será facilitada pelo mesmo serviço. No entanto, as informações serão coletadas de um recurso separado. Por exemplo, é melhor colocar dados de autenticação do usuário e administrativos em bancos de dados separados. Isso garante que, se houver uma maneira de manipular o banco de dados por meio da camada da aplicação (por exemplo, usando uma inserção de SQL), os hackers só poderão se mover lateralmente pelo banco de dados do usuário. Eles não podem se mover verticalmente, elevando seus privilégios ao de administrador, sem comprometer o próprio banco de dados. Esta é uma melhoria significativa na segurança organizacional.

Se possível, defina sistemas de armazenamento separados para:

- Identidades de endpoint
- Usuários
- Credenciais de administrador
- Parceiros

Isso criará uma separação lógica de tarefas para aplicações e infraestrutura, mas dentro da mesma API de autenticação gerenciada pelo serviço da raiz organizacional de confiança.

Considere o uso de material das seguintes organizações para auxiliar nesta recomendação:

- OAuth 2.0 [8]
- OpenID Foundation [9]
- GSMA Mobile Connect [10]

6.12.1 Risco

Sem uma metodologia para impor a autenticação e a autorização na camada da aplicação, não há como o sistema confirmar se as ações supostamente de um usuário estão realmente autorizadas para ele. A implementação desta recomendação garante que cada ação seja rastreável para um usuário autenticado e uma autorização. Essas métricas podem ser armazenadas e posteriormente revisadas no caso de suspeita de comprometimento. Sem essas etapas, não haverá salvaguardas que minimizem o risco de abuso.

6.13 Regras de firewall “default-open” ou falha aberta e fortalecimento do sistema

Em alguns ambientes de infraestrutura de serviço, os mecanismos de proteção de entrada e saída não são configurados por padrão. Isso significa que os engenheiros devem empregar firewalls ou conjuntos de regras de tráfego de rede. Essas regras devem ser definidas na infraestrutura antes que qualquer serviço seja implantado para o público.

No entanto, há momentos em que essas tecnologias podem não ser suficientes para proteger a infraestrutura de serviços. Às vezes, firewalls e outros sistemas de proteção de tráfego de rede falham. Quando esses sistemas falham, eles geralmente não conseguem iniciar. A razão para isso é que, se o sistema falhar, o tráfego ainda poderá funcionar, pois o tráfego para outros ambientes de computação será roteado pela infraestrutura, juntamente com o tráfego do provedor de serviços da IoT. Assim, o tráfego não pode repentinamente parar. Como resultado, muitas vezes, o sistema falha ao iniciar para permitir que o maior número possível de serviços continue funcionando.

A equipe de engenharia deve empregar o fortalecimento do sistema operacional para garantir que os efeitos da infraestrutura com falha não resultem em um evento de segurança catastrófico. Em vez disso, pode significar simplesmente que mais conexões podem ser feitas à infraestrutura de serviço existente.

Por exemplo, serviços ocultos não devem ser colocados atrás de tecnologias como firewalls. Em vez disso, as Redes Privadas Virtuais (VPNs) ou outras proteções de alta segurança podem ser usadas para resguardar os serviços dos hackers.

Observe que os firewalls de software têm um risco adicional, pois podem ser manipulados por um hacker experiente. Se um firewall de software for usado, qualquer infraestrutura de servidor que esteja incorretamente protegida pode ser manipulada por um hacker. Em outras palavras, se um serviço público em execução em um servidor tiver privilégios desnecessários (como privilégios de superusuário) e estiver comprometido, o hacker provavelmente será capaz de desabilitar o firewall de software. Assim, a equipe de engenharia deve avaliar se um firewall de software é muito poderoso para a arquitetura

6.13.1 Risco

Sem empregar estratégias para compensar falhas nos sistemas de segurança de tráfego de rede, o ambiente estará sujeito a ataques desnecessários que poderiam ser facilmente evitados com estratégias padrão de fortalecimento de serviços.

6.14 Avalie o modelo de privacidade das comunicações

A privacidade das comunicações é um tópico ligeiramente diferente da privacidade da aplicação (descrita acima), ou da segurança de comunicação das informações. Embora a privacidade seja amplamente avaliada a partir da possibilidade de terceiros de ler ou interceptar dados com eficiência, a confidencialidade e a integridade não representam o escopo total da privacidade das comunicações.

Outros problemas que afetam a privacidade das comunicações incluem:

- Exclusividade criptográfica de cada mensagem
- Padrões de transmissão
- Metadados em texto simples
- Endereços de hardware ou números de série relativos

Embora cada mensagem deva ser confidencial e ter integridade verificável, ela também deve ser criptograficamente única. Se determinadas mensagens forem enviadas em resposta a eventos previsíveis por um hacker, todas as respostas que não forem

criptograficamente exclusivas poderão ser repetidas pelo invasor. Cada mensagem deve ser exclusiva para proibir a possibilidade do hacker de capturar e reproduzir mensagens que sejam genuínas.

Os padrões de transmissão podem permitir que um hacker identifique um usuário em particular ou equacione o comportamento com uma determinada ação atribuída. Por exemplo, a tecnologia que emite uma mensagem quando um usuário entra em uma determinada zona física pode ser identificada por “sniffers” que são capazes de receber essas mensagens à medida que são transmitidas pelo ar. Embora possa não ser intuitivo, isso é possivelmente uma fonte de risco legal se um hacker puder identificar quem está em um local físico e onde está. Os padrões de rede devem ser avaliados para determinar se existe uma maneira simples de os hackers transformarem os padrões de transmissão em dados acionáveis.

Os metadados são usados há muito tempo pelos serviços de inteligência para avaliar o contexto dos sistemas de mensagens sem exigir uma garantia ou outro acesso legal aos dados criptografados. Geralmente, os metadados são informações suficientes para uma organização criar inteligência acionável. No entanto, agora, amadores, organizações criminosas e usuários curiosos são capazes de usar metadados para rastreamento e outros fins possivelmente nefastos. Como resultado, é mais importante do que nunca diminuir a quantidade de metadados disponíveis para terceiros. Sempre que possível, limite esta quantidade apenas às informações necessárias para que um ponto de comunicação avalie se a mensagem é destinada a eles.

Ao longo dessa linha de pensamento, o endereço de hardware do módulo de comunicação e quaisquer números de série únicos devem ser protegidos ou randomizados, se possível. Por exemplo, a Apple mudou o modelo do iOS para sondagem de pontos de acesso Wi-Fi. Em vez de usar o endereço de hardware estático, eles mudaram sua tecnologia para usar um endereço de hardware aleatório, o que diminui a possibilidade de alguém rastrear a localização de um usuário com base em varreduras ativas de Wi-Fi. A tecnologia da IoT funcionará de forma semelhante, mas terá um conjunto maior de tecnologias de comunicação afetadas por esse problema. Algumas tecnologias não serão capazes de gerar aleatoriamente endereços de hardware, como o celular. Mas outros, como 802.15.4, Wi-Fi e Bluetooth, podem ser capazes disso, dependendo da funcionalidade do firmware.

6.14.1 Risco

Embora não seja necessário dizer que a segurança das comunicações é um requisito, às vezes não é clara a razão de ser um requerimento. A segurança das comunicações não garante apenas que um hacker não possa ler os dados. Também garante que:

- Um endpoint não possa ser simulado
- Um serviço crítico não possa ser simulado
- Mensagens violadas possam ser detectadas
- Alterações no software ou configurações de segurança possam ser executadas com confiança

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

7 Recomendações de média prioridade

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

7.1 Defina um ambiente de execução de aplicações

Vários pontos devem ser observados sobre os ambientes de execução de aplicações:

- A linguagem de programação usada pode ter um relacionamento direto com a segurança:
  - Linguagens como PHP e Ruby podem apresentar problemas de segurança
  - Linguagens como GoLang e Erlang podem diminuir o risco
- Bibliotecas de terceiros devem ser monitoradas, gerenciadas e auditadas quanto ao risco:
  - Algumas bibliotecas não são bem mantidas
  - Algumas bibliotecas nunca foram auditadas por falhas de segurança
  - Algumas bibliotecas exigem dependências desatualizadas que possuem falhas de segurança conhecidas
- Sempre execute uma aplicação como um usuário não privilegiado:
  - Se a aplicação exigir um recurso privilegiado, use um wrapper para provisionar esse recurso antes de eliminar privilégios e executar a aplicação completa
- Use um modelo TCB e Bootstrap bem definido:
  - Aplicações que possuem ambientes bem definidos são mais *confiáveis* e *seguras*

Considere o uso de material da seguinte organização para auxiliar nesta recomendação:

- OWASP [5]

7.1.1 Risco

Aplicações que são implantadas com uma arquitetura segura podem estar sujeitas a comprometimentos que não podem ser facilmente rastreados até uma origem específica. Ferramentas e técnicas para comprometer serviços e aplicações se tornaram avançadas na última década. Algumas tecnologias de software livre, como o Metasploit, permitem o desenvolvimento e a integração de explorações personalizadas em uma plataforma de ataque que pode fornecer tecnologias para aumentar a furtividade de um ataque.

Um ambiente seguro de execução de aplicações pode combater esse risco protegendo a maneira como eles são executados, interagem entre si e os tipos de tecnologias usadas durante o tempo de execução. Esses atributos podem não apenas diminuir a probabilidade de ocorrer um comprometimento, como também podem adicionar rastreabilidade e recursos críticos de registro para detectar e diagnosticar a vulnerabilidade explorada.

7.2 Use os serviços de monitoramento aprimorado dos parceiros

Se o parceiro que está sendo utilizado for uma operadora de rede móvel, identifique se ela é capaz de oferecer serviços de monitoramento. Algumas operadoras de rede são capazes de analisar o comportamento dos endpoints que se comunicam através de sua rede. Operadoras com esse tipo de capacidade têm experiência em avaliar quais métricas equivalem a comportamentos anômalos e hackers.

Isso permitirá que os negócios de IoT identifiquem mais rapidamente se um determinado usuário ou endpoint é uma ameaça ou se foi comprometido por um hacker. Como resultado, as empresas podem reagir com mais eficácia para evitar ataques contra outras áreas da infraestrutura do negócio.

A complexidade com esse serviço vem da capacidade da operadora de rede fornecer inteligência em uma janela de tempo significativa. Se a operadora de rede puder fornecer a inteligência somente quando o hacker atacar o negócio de IoT, os sistemas de monitoramento e registro localizados na infraestrutura da empresa de IoT poderão detectar seu comportamento. No entanto, se a operadora de rede for capaz de notificar a empresa sobre o comportamento agressor na camada de rede e puder identificar qual assinante individual está emitindo tráfego de rede anômalo, a empresa poderá limitar a exposição do ecossistema de IoT ao isolar esse tráfego individual do usuário.

7.2.1 Risco

Há determinadas tecnologias nas quais o provedor de serviços de IoT dependerá e que não podem ser monitoradas por ele. Uma dessas tecnologias é a rede de comunicações que conecta um endpoint ao Ecossistema de Rede e Serviços. Sem serviços de monitoramento, o provedor de serviços de IoT não terá uma janela para os eventos que ocorrem dentro da própria rede. Portanto, se uma identidade A no nível da aplicação estiver tentando comprometer um serviço, a organização não conseguirá identificar se o endpoint B é realmente a unidade que se conectou à rede de comunicações. Essa lacuna na informação é crítica, pois a organização pode atribuir o ataque à identidade A, em vez do endpoint B comprometido.

7.3 Use um APN privado para a conectividade celular

Um APN é um componente de comunicação celular que conecta a rede sem fio à Internet. Este ponto atua como, essencialmente, uma Rede Privada Virtual (VPN) entre equipamentos celulares endpoints e a infraestrutura de serviços com os quais o endpoint deve interagir. Um APN privado (às vezes chamado de APN seguro) é uma versão que foi protegida para implementar vários controles desejados:

- Acesso limitado e restrito a clientes autenticados
- Regras de firewall
- Comunicação endpoint-endpoint é desativada à força
- Monitoramento de serviços para detecção de anomalias
- Segurança opcional ou serviços de monitoramento

Ao restringir o acesso ao APN, uma organização pode garantir que apenas endpoints autenticados tenham permissão para se conectar à infraestrutura de serviços disponibilizada por meio deste. Isso diminui a possibilidade de clientes sem fio ou aleatórios

se conectarem ao APN e terem acesso a seus serviços restritos. Além disso, permite que a organização identifique quais clientes específicos estão se comportando de maneira anormal, o que permite à organização vincular o comportamento negativo a um componente de hardware ou a um usuário específico.

O firewall garante que as entidades conectadas ao APN do lado cliente (Endpoint Ecosystem) e do lado serviço (Service Ecosystem) sejam impedidas de se comunicar usando canais não aprovados. Isso também restringe a possibilidade de um endpoint abusar do APN como um canal para a Internet aberta e isolar o tráfego de um conjunto específico de serviços aprovados.

As restrições de comunicação do endpoint garantem que não autorizados não consigam atacar outros endpoints usando o APN como uma rede de banda larga. Em vez disso, toda a comunicação deve orbitar por serviços aprovados pela organização. Se preferir, a organização pode proibir completamente a comunicação ponto-a-ponto.

Os serviços de monitoramento aprimoram as melhorias de segurança que serão feitas pela organização no monitoramento da infraestrutura de nuvem ou serviço existente. Ao parear os serviços de monitoramento existentes com as tecnologias de monitoramento de rede e o APN oferecido pela operadora de rede, a organização pode rastrear mais facilmente a fonte do comportamento anômalo. Isso permite que a organização inspecione mais profundamente os incidentes que ocorrem em sua infraestrutura de endpoints ou de serviços. Por exemplo, se a camada de aplicação indicar que o usuário A pode estar comprometido, mas o equipamento do usuário B estiver fazendo a conexão autenticada com o APN, a organização poderá usar os serviços de monitoramento APN para identificar se o usuário B possivelmente comprometeu o usuário A, ou se um hacker comprometeu tanto o usuário A quanto o B.

Operadoras de rede têm serviços adicionais que podem ser colocados em camadas sobre os serviços descritos acima. Esses serviços poderão ajudar a monitorar agentes mal-intencionados e colocá-los em uma lista negra na rede, monitorar usuários específicos ou conjuntos de usuários e redirecionar determinados tipos de tráfego que podem indicar anomalias. Outras opções podem estar disponíveis. Envolva a operadora de rede para determinar quais serviços são adequados para sua organização.

Embora a utilização de todos esses serviços em conjunto possa parecer desafiadora, trabalhar com a operadora de rede simplificará o processo e a integração dessas ofertas na infraestrutura existente da empresa. A complexidade virá da utilização efetiva dos dados e requer uma equipe de engenharia que seja capaz de processar e gerenciar os dados de maneira razoável. Alguns serviços podem incorrer em um custo adicional. Determine qual modelo de preço e serviços funcionará melhor para sua organização.

7.3.1 Risco

Sem um APN privado, um endpoint pode se conectar a praticamente qualquer serviço ou tecnologia, inclusive fazer conexões diretas com outros endpoints no APN ou serviço arbitrário na Internet. Como isso permitiria que um endpoint comprometido interaja com praticamente qualquer serviço na Internet e possa transformá-lo em um alvo prático para agir como um proxy para atacar redes ou serviços mais seguros, essa recomendação deve ser imposta para restringir a possibilidade de endpoints de fazer conexões arbitrárias e não

autorizadas. É muito melhor para o negócio e para a segurança de todo o ecossistema de IoT quando os endpoints são forçados a se conectar apenas a serviços aprovados.

7.4 Defina uma política de distribuição de dados de terceiros

Após as classificações de segurança terem sido definidas, e os tipos de dados terem sido atribuídos a uma classificação válida, e uma política de violação tiver sido promulgada, uma política de distribuição de dados deverá ser criada. Uma política de distribuição de dados descreve como as informações devem ser processadas por meio de controles técnicos e aplicações de serviço aos quais foi concedida permissão de acesso. O modelo de permissões faz parte da política de distribuição de dados e é combinado com a capacidade do usuário de criar permissões de dados granulares.

Embora uma política de distribuição de dados possa ser muito descritiva, existem vários elementos-chave que ajudarão a definir uma política bem-sucedida:

- Qual nível de autenticação mútua é necessário para trafegar esses dados
- Que confidencialidade e integridade dos dados são exigidas
- Que capacidade a empresa tem para reter os dados
- Que capacidade o parceiro tem para reter os dados
- Se a retenção for permitida, por qual período de tempo os dados podem ser retidos
- Qual nível de segurança de armazenamento deve ser aplicado aos dados
- Qual classificação de segurança de acesso deve ser aplicada aos dados

7.4.1 Risco

As políticas de distribuição de dados impõem requisitos de segurança a parceiros que podem não aderir ao mesmo nível de segurança internamente, como faz o provedor de serviços IoT. Como o provedor de serviços de IoT não pode controlar a segurança que um parceiro implementou em seus serviços internos e na rede, este só pode garantir que os dados entregues a um parceiro sejam trafegados de maneira segura. Sem essa definição, o parceiro pode utilizar configurações inseguras que podem expor os dados do usuário a invasores enquanto os dados ainda estiverem sob o controle do provedor de serviços de IoT. Ao impor controles de segurança rigorosos para o canal de comunicações, o provedor de serviços de IoT prova que está fazendo tudo o que pode para impor a segurança até que os dados estejam fora de seu controle.

7.5 Desenvolva um filtro de dados para terceiros

Aceitar dados gerados dinamicamente, tais como anúncios, de um parceiro requer um certo nível de presunção em relação à qualidade e segurança dos dados. Em vez de fazer presunções e aplicar os dados à camada de entrada, resso, a equipe de engenharia deve tomar medidas para garantir que os dados distribuídos da aplicação de serviço para ou de um parceiro sejam bem formados e não contenham conteúdo possivelmente mal-intencionado.

Para fazer isso, a equipe de engenharia deve considerar o seguinte modelo:

- Os dados se ajustam ao formato que o parceiro descreveu para o modelo de dados
- Os dados estão bem formatados

- Os dados representam um objeto polimórfico que poderia ser mal interpretado pelo cliente
- Os dados afetarão a maneira como o cliente renderiza a camada de comunicação
- Os dados afetarão como o cliente interpreta a camada de comunicação
- Os dados atraem ou solicitam que o usuário execute um comportamento que enfraqueceria a segurança
- Os dados falsificam ou representam um componente (campo de entrada de senha) da GUI cliente

Rejeite todos os dados que não se encaixem em um modelo aprovado. Notifique a administração imediatamente sobre a detecção de tais dados e inclua o maior número possível de métricas em relação à origem e ao formato dos dados. Registre uma amostra, se possível, em um banco de dados seguro.

Aceitar dados gerados dinamicamente, tais como anúncios, de um parceiro requer um certo nível de presunção em relação à qualidade e segurança dos dados. Em vez de fazer presunções e aplicar os dados à camada de entrada, resso, a equipe de engenharia deve tomar medidas para garantir que os dados distribuídos da aplicação de serviço para ou de um parceiro sejam bem formados e não contenham conteúdo possivelmente mal-intencionado.

Para fazer isso, a equipe de engenharia deve considerar o seguinte modelo:

- Os dados se ajustam ao formato que o parceiro descreveu para o modelo de dados
- Os dados estão bem formatados
- Os dados representam um objeto polimórfico que poderia ser mal interpretado pelo cliente
- Os dados afetarão a maneira como o cliente renderiza a camada de comunicação
- Os dados afetarão como o cliente interpreta a camada de comunicação
- Os dados atraem ou solicitam que o usuário execute um comportamento que enfraqueceria a segurança
- Os dados falsificam ou representam um componente (campo de entrada de senha) da GUI cliente

Rejeite todos os dados que não se encaixem em um modelo aprovado. Notifique a administração imediatamente sobre a detecção de tais dados e inclua o maior número possível de métricas em relação à origem e ao formato dos dados. Registre uma amostra, se possível, em um banco de dados seguro.

7.5.1 Risco

Dados gerados dinamicamente por terceiros podem conter malware, conteúdo impróprio ou outros dados indesejáveis, intencionalmente ou não. Sem um filtro de acesso voltado para a definição do serviço de terceiros, a organização pode se arriscar acidentalmente a permitir que malwares ou outros conteúdos maliciosos cheguem ao usuário final. Isso pode resultar em comprometimento do sistema ou simplesmente perda de clientes, devido aos efeitos colaterais de tais dados.

8 Recomendações de baixa prioridade

Recomendações de baixa prioridade englobam o conjunto de recomendações que se aplica a riscos extremamente dispendiosos para combater ou que provavelmente não afetam o design do endpoint. Embora essas recomendações sejam valiosas e as informações detalhadas destas sejam importantes, as estratégias de mitigação ou reparação 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.

8.1 Rowhammer e ataques similares

Algumas implementações da tecnologia RAM moderna, como DRAM (Dynamic Random Access Memory) e SRAM (Static Random Access Memory), são vulneráveis a erros que podem ser induzidos por certas sequências de acesso à memória. Abusar desse tipo de erro pode resultar na alteração de um bit, ou bits específicos, em áreas previsíveis da memória. Uma exploração bem-sucedida dessa condição pode alterar os bits na memória que representam os tipos de privilégios denotados pelo software.

Em outras palavras, se explorar corretamente, um hacker pode passar os privilégios de um usuário para outro, manipulando uma falha de hardware em implementações modernas de DRAM ou SRAM. Muitas implementações modernas de DRAM e SRAM foram encontradas como comprovadamente exploráveis por meio dessa vulnerabilidade. No entanto, isso requer a capacidade de executar códigos no sistema local para criar as sequências de acesso à memória capazes de acionar esse bug.

E, no entanto, pode ser possível acionar esse tipo de comportamento remotamente por meio de linguagens interpretadas, como GoLang Sandboxed, Python, Erlang e outras mais. No entanto, a precisão desses tipos de ataques ainda não foi documentada e é altamente improvável que funcione efetivamente como um abuso.

Esse ataque deve ser resolvido no nível do hardware. No entanto, os engenheiros podem atenuar o risco de abuso impedindo que os clientes executem código, mesmo que por meio de uma máquina virtual ou tempo de execução, em um determinado serviço. Ao restringir essa capacidade, os engenheiros poderão impedir que os hackers criem as sequências de acesso à memória necessárias nesse ataque.

8.1.1 Risco

Sem proteções suficientes contra esse tipo de ataque, os hackers podem aumentar remotamente privilégios ou executar código arbitrário contra um host de destino. Deve-se notar, no entanto, que um ataque bem-sucedido requer um conhecimento extremamente profundo do hardware, sistema operacional, vetor de ataque e outros fatores que tornam esse ataque improvável e raro.

8.2 Comprometimento de máquinas virtuais

A infraestrutura moderna de serviços geralmente utiliza máquinas virtuais para implantar serviços sob demanda. Embora esse modelo tenha se mostrado extremamente conveniente e fácil de implantar, o problema com essa metodologia é a segurança da infraestrutura como um todo. Embora a equipe de engenharia possa ter sucesso na implantação de uma

arquitetura bem projetada, a organização que gerencia e implanta a infraestrutura virtual pode não ser tão bem-sucedida.

Uma das principais preocupações da implantação em ambientes de servidores virtuais é a capacidade de comprometimento dos hosts ou dos servidores (virtual guests) interceptarem os dados de outros convidados em execução na mesma infraestrutura.

Embora esses ataques sejam preocupações válidas que devem ser avaliadas pelo provedor de serviços de IoT, eles geralmente precisam de muita habilidade e tempo para serem aperfeiçoados. Assim, é possível que um ataque ocorra, mas provavelmente será um evento raro. No entanto, se a infraestrutura de serviço não for bem protegida, será possível que os hackers consigam comprometer o acesso administrativo às máquinas virtuais. Esse tipo de violação pode não exigir uma grande habilidade para ser bem-sucedido.

Uma maneira de combater esse problema é com a provisão de servidores. Esse processo garantirá que cada servidor seja codificado com um conjunto exclusivo de chaves criptográficas. Se esse processo for seguido, qualquer comprometimento em um único servidor pode ser limitado a este.

8.2.1 Risco

O risco em não se preparar para esse tipo de ataque pode deixar a infraestrutura de serviços vulnerável a muitos outros tipos de ataques. A configuração do servidor usando chaves acessíveis a partir da infraestrutura de serviço, o vazamento de dados, o comprometimento da privacidade e da identidade do usuário podem se tornar viáveis.

8.3 Desenvolva uma API para os usuários controlarem os atributos de privacidade

Todos os usuários devem poder controlar quais informações podem oferecer a terceiros por meio de APIs de serviço. As informações devem ser classificadas em classes de dados e atribuídas a classificações de segurança. Os usuários devem conseguir recuperar os tipos de dados e classificações usados na modelagem de suas contas. O usuário deve poder aplicar restrições aos tipos de dados, para permitir que eles concedam ou revoguem o acesso a esses dados a terceiros.

Isso pode vir na forma de uma API autenticada ou de uma GUI que permite controles simples de “Sim” ou “Não” de modo geral, ou por parceiro.

8.3.1 Risco

Sem que os usuários sejam capazes de controlar com quais dados irão contribuir para um provedor de serviços de IoT, eles correm o risco de que seus dados sejam expostos no caso de uma violação de segurança ao provedor ou a um de seus parceiros. Como certos usuários correm um risco muito maior do que outros, cada um deles deve poder ajustar suas restrições de privacidade de acordo com suas necessidades pessoais. Disponibilizar essa interface ajuda a confirmar que a garantia esteja presente. O usuário deve assumir a responsabilidade de ajustar os controles de acordo com suas necessidades. Por exemplo, o oneM2M (por meio do TS-0003) permite que o usuário defina as preferências de privacidade a um provedor de serviços.

8.4 Defina um modelo de avaliação falso positivo/negativo

Embora a análise de falsos positivos seja um tópico extremamente complexo, existe uma maneira simples de identificar se uma tecnologia tem maior probabilidade de apresentá-los. Isso é feito avaliando os seguintes itens:

- A fonte de dados é *confiável*
- A fonte de dados pode ser adulterada ou falsificada
- É a fonte de dados do próprio domínio
- Os dados podem ser confirmados a partir de várias fontes da mesma origem
- As fontes de dados corroboradas estão no mesmo sistema do endpoint
- São confirmadas as fontes de dados fáceis de falsificar ou usurpar
- As ferramentas estão prontamente disponíveis para manipular a fonte de dados
- Qual nível de conhecimento ou custo é necessário para manipular a fonte de dados
- O dispositivo conectado à fonte de dados é *confiável*

Todos esses atributos, e outros mais, podem ser usados para avaliar se os dados são confiáveis. Isso é extremamente importante, pois decisões críticas que afetam o mundo físico podem resultar em efeitos potencialmente prejudiciais. É imperativo que a equipe de engenharia crie um modelo de confiabilidade e aplique-o a cada fonte de dados envolvida na tomada de decisões críticas. Se o peso da fonte de dados é tal que não pode ser confiável, a ação mais racional e mais segura deve ser tomada.

É importante anotar que a equipe de engenharia não é a única entidade que deve tomar esse tipo de decisão. A liderança empresarial, a equipe de advogados e a de seguros devem estar envolvidas na determinação da reação correta em cenários potencialmente perigosos. Os engenheiros devem então codificar o processo correto de tomada de decisão na tecnologia, de uma *maneira verificável e reproduzível*.

Esse processo é altamente desafiador, pois exige a atenção de toda a organização sobre como a tecnologia deve reagir em cenários críticos. Confiabilidade é um atributo desafiador para se aplicar a um componente da tecnologia, especialmente em uma tecnologia embutida.

8.4.1 Risco

Sem um modelo de avaliação para falsos positivos, os engenheiros podem gastar muito tempo analisando eventos benignos enquanto outros mais importantes estão ocorrendo. Isso pode resultar em aumento do risco de que as métricas analisadas pela organização não fornecerão orientações claras sobre quais os tipos de eventos que estão ocorrendo na produção. Isso desvaloriza a infraestrutura de registro e monitoramento e reduz a capacidade da organização de usar esses recursos preciosos em seu benefício.

9 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 riscos antes e durante eventos relacionados à segurança e políticas e procedimentos para manipular esses eventos. Ao analisar quais conceitos de segurança de alto nível são importantes para o provedor de serviços de IoT, as perguntas frequentes de segurança podem ser revisadas. Isso deve orientar a equipe de engenharia em relação a quais recomendações são mais relevantes para resolver as lacunas em sua arquitetura de segurança.

Na medida que a equipe avança em sua definição de arquitetura, ela pode revisar recomendações particulares conforme suas questões de segurança e preocupações se tornem mais exclusivas de 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 seus parceiros para construir uma base de conhecimento comum para os riscos e as estratégias de reparação. Juntas, nossas organizações podem construir tecnologia e conhecimento para ajudar mas às outras na construção da segurança para o futuro da IoT.

Anexo A Gestão do Documento

A.1 Histórico do documento

Versão	Data	Breve Descrição de Alterações	Responsável pela aprovação	Editor / Company
1.0	08-Fev-2016	Novo PRD CLP.12	PSMC	Ian Smith GSMA & Don A. Bailey Lab Mouse Security
1.1	07-Nov-2016	Adicionadas referências ao esquema de avaliação de segurança em IoT da GSMA. Correções editoriais menores.	PSMC	Ian Smith GSMA
2.0	29-Set-2017	Referências adicionais de M2M adicionadas	Grupo de segurança em IoT	Rob Childs GSMA

A.2 Outras Informações

Tipo	Descrição
Proprietário do documento	Programa de IoT da GSMA
Contato	Rob Childs - GSMA

É nossa intenção proporcionar um produto de qualidade para o seu uso. Se você encontrar quaisquer erros ou omissões, por favor, nos contacte com seus comentários. Você pode nos notificar em prd@gsma.com

Seus comentários, ou sugestões e dúvidas, são sempre bem-vindos.