Problemas de Segurança em Redes IoT

Marcelo Maia Juvencio

13 de outubro de 2024

1 Introdução

Segurança e privacidade são requisitos essenciais em sistemas de Internet das Coisas (IoT) para garantir que o sistema opere de maneira segura e proteja os dados pessoais e sensíveis dos usuários. Com o aumento cada vez maior de dispositivos de IoT, segundo a Statista a previsão é que o número de dispositivos de Internet das Coisas (IoT) no mundo quase dobre, de 15,9 bilhões em 2023 para mais de 32,1 bilhões de dispositivos de IoT em 2030, requisitos de segurança são prioridades na implantação de um serviço de IoT.

Por mais complexo que um sistema possa parecer, um ataque normalmente é feito de forma simples explorando "brechas"de segurança no sistema. Investigação da Kaspersky mostra que o principal método para infectar dispositivos IoT é por meio de ataques que adivinham senhas fracas por força bruta, seguida pela exploração de vulnerabilidades em serviços de rede como o Telnet.

Não é objetivo deste trabalho discutir em detalhes cada tipo de ataque e suas respectivas soluções, assim foram elaborados dois glossários ao final do trabalho, um dedicado aos tipos de ataque e outro às soluções propostas, que servirão como uma referência rápida para cada tipo de ataque e solução mensionados ao longo do trabalho.

2 Ataques por camadas

A arquitetura da Internet das Coisas (IoT) é composta pelas camadas de Percepção, Rede e Aplicação. Cada uma dessas camadas apresenta vulnerabilidades únicas que podem ser exploradas por atacantes, comprometendo a segurança de dispositivos e sistemas IoT. Os tipos de ataques e soluçoes variam de camada para camada e de componente para componente conforme mostrado nas tabela 1, 2 e 3.

A seguir vamos apresentar os principais problemas, desafios e soluções na questão da segurança em cada camada do sistema IoT.

2.1 Camada de Percepção

A camada de percepção, onde ficam concentrados os dispositivos sensores, tem um grande desafio com a segurança, pois tais dispositivos tem características de baixo processamento, baixo consumo de energia e baixa capacidade de armazenamento que tornam os esquemas de autenticação seguros baseados em criptografia de chave pública inviáveis. A tabela 1 mostra os diversos tipos de ataques aos RFID Nodes, Sensor Node e Sensor Gatways e as soluções para tentar mitigar estes ataques.

	Layer/ component	Attacks	Solutions
a.	**Perception Layer**		
	Perception Nodes RFID	Tracking, DoS, repudiation, spoofing, eavesdropping, data newness, accessibility, self-organization, time management, secure localization, tractability, robustness, privacy protection, survivability, and counterfeiting [13].	Access control, data encryption which includes non-linear key algorithms, IPSec protocol utilization, cryptography techniques to protect against side channel attack [9], [14], Hashed-based access control [15], Ciphertext re-encryption to hide communication [16], New lightweight implementation using SHA-3 appointed function Keccak-f (200) and Keccak-f (400) [17]
	Sensor nodes	Node subversion, node failure, node authentication, node outage, passive information gathering, false node message corruption, exhaustion, unfairness, sybil, jamming, tampering, and collisions [18, 19]	Node authentication, Sensor Privacy
	Sensor Gateways	Misconfiguration, hacking, signal lost, DoS, war dialing, protocol tunneling, man-in-the-middle attack, interruption, interception, and modification fabrication [20]	Message Security, Device Onboard Security, Integrations Security [21]

Tabela 1: Ataques e soluções na camada de percepção

Na camada de percepção da arquitetura IoT, um dos ataques mais críticos é o ataque de comprometimento de nós (Node Compromise) onde um dispositivo ou nó dentro de uma rede IoT é acessado de forma não autorizada, permitindo que um invasor obtenha controle total ou parcial sobre o dispositivo.

2.2 Camada de Rede

A camada de rede, diferente da camada de percepção, possui mais recursos de processamento e armazenamento, mas enfrenta desafios próprios com relação a proteção de dados e disponibilidade. Como ela é responsável por transportar os dados coletados pelos sensores, fazer a comunicação entre os nós de percepção e a camada de aplicação, a variedade de sensores e aplicações requer diferentes protocolos de comunicação, como Wi-Fi, Bluetooth, Zigbee, LoRa, NB-IoT, 6LoWPAN, entre outros, que tem, cada um, vulnerabilidades e características de segurança diferentes, o que dificulta a criação de um padrão de soluções de

segurança. A tabela 2 mostra os diversos tipos de ataques ás Comunicações Móveis, Computação em Nuvem e Internet e as soluções para tentar mitigar estes ataques.

	Layer/ component	Attacks	Solutions
b.	**Network layer**		
	Mobile Communi-cation	Tracking, eavesdropping, DoS, bluesnarfing, bluejacking, bluebugging alteration, corruption, and deletion [1], [5], [38]	Developing secure access control mechanisms to mitigate the threats by employing biometrics, public-key crypto primitives and time changing session keys.
	Cloud Computing	Identity management, heterogeneity which is inaccessible to an authentic node, data access controls, system complexity, physical security, encryption, infrastructure security and misconfiguration of software [22]	**Identity privacy** – Pseudonym [23–25], group signature [24], connection anonymization [26, 30] **Location privacy** – Pseudonym [23–25], one-way trapdoor permutation [25, 27] **Node compromise attack** – Secret sharing [27–29], game theory [26], population dynamic model [27] **Layer removing/adding attack** – Packet transmitting witness [25, 27, 30], aggregated transmission evidence [27] **Forward and backward security** – Cryptographic one-way hash chain [23, 24] **Semi-trusted/malicious cloud security** – (Fully) homomorphic encryption [31], zero knowledge proof [32]
	Internet	Confidentiality, encryption, viruses, cyberbullying, hacking, identity theft, reliability, integrity, and consent [33]	Identity Management for confidentiality [34], Encryption schemes for confidentiality of communication channels [35], Cloud based solutions to establish secure channels based on PKI for data and communication confidentiality [35]

Tabela 2: Ataques e soluções na camada de rede

Um dos ataques mais destacados nesta camada é o ataque de Denial of Service (DoS), especialmente sua forma distribuída, o Distributed Denial of Service (DDoS), que visa sobrecarregar um sistema ou rede IoT, tornando-os indisponíveis para os usuários.

2.3 Camada de Aplicação

A segurança na camada de aplicação enfrenta uma série de problemas e desafios de segurança pois é responsável pela interface entre os usuários e os dispositivos IoT. Seja qual for a aplicação, Smart City, Smart Agriculture, Smart Home entre outras, além dos problemas com os dispositivos já mencionados, temos agora a interação humana que faz acessos com senhas fracas a dispositivos com controles de acesso inadequados. A tabela 3 mostra alguns tipos de ataques em aplicações de IoT. Na camada de aplicação da arquitetura IoT, um dos ataques

	Layer/ component	Attacks	Solutions
c.	**Application Layer**	Data privacy, Tampering Privacy, Access control, disclosure of information [18]	Authentication, key agreement and protection of user privacy across heterogeneous networks [1], Datagram Transport Layer Security (DTLS) for end-to-end security [36], Information Flow Control [28]

Tabela 3: Ataques e soluções na camada de aplicação

mais destacados é o ataque Man-in-the-Middle (MitM) que intercepta e possivelmente altera a comunicação entre duas partes sem que elas percebam. Esse ataque compromete a confidencialidade, integridade e autenticidade dos dados, e tem grande impacto especialmente em aplicações críticas, onde a comunicação segura entre dispositivos e sistemas de back-end é essencial.

3 Medidas de Segurança

As medidas de segurança, como vimos, são várias, distribuidas em cada camada da arquitetura IoT, cada uma com suas particularidades e limitações, mas vamos dar destaque a uma das medidas de segurança mais importantes em toda a arquitetura IoT, Autenticação e Autorização.

3.1 Autenticação e Autorização

Autenticação e Autorização são essenciais para proteger a integridade, confidencialidade e autenticidade das interações entre dispositivos, redes e aplicações. Vamos fazer uma analogia para mostrar sua importância com a seguinte pergunta. De que adianta ter um cofre em casa escondido atras de um quadro se qualquer pessoa pode entrar nesta casa? Nos sistemas de IoT a idéia é a mesma quando se trata de segurança. Por exemplo, a camada de rede pode ter recursos de proteção ao fluxo de dados, mas não é qualquer dispositivo que pode ter acesso às informações. A seguir vamos detalhar cada processo.

3.1.1 Autenticação

O processo de autenticação envolve duas entidades, a entidade solicitante e a entidade autenticadora. O objetivo é que a entidade autenticadora reconheça a identidade da entidade solicitante, mas em contra partida, a entidade solicitante também precisa reconhecer a identidade da entidade autenticadora. A entidade solicitante pode ser humana ou máquina, já a entidade autenticadora será sempre máquina.

Em IoT a autenticação mais comum é a Machine-to-Machine (M2M), onde um dispositivo, no papel de entidade autenticadora, precisa reconhecer a identidade do dispositivo entidade solicitante. A identidade de dispositivos são verificadas através de Primitivas criptográficas que são algoritmos de criptografia. O anexo A mostra detalhes das principais Primitivas criptográficas.

A identidade dos dispositivo entidade autenticadora é verificada por certificados digitais, assinaturas criptográficas, ou técnicas de autenticação mútua. O anexo B mostra detalhes dos principais métodos.

E assim, definidadas as identidades de cada entidade, a entidade solicitante (o dispositivo ou usuário) envia suas credenciais ou informações de autenticação para a entidade autenticadora. A entidade autenticadora valida essas credenciais comparando-as com informações pré-registradas ou executando um protocolo criptográfico. Se as credenciais forem válidas, a entidade autenticadora concede acesso ao sistema ou serviço. Caso contrário, o acesso é negado.

3.1.2 Autorização

Após o processo de autenticação a entidade solicitante teve acesso concedido ao sistema pela entidade autenticadora, porém um dispositivo autenticado, por exemplo, não tem permissão ou autorização para executar determinadas ações dentro do sistema, regras e direitos são concedidos a este dispositivo, este é o processo de Autorização. A autorização é baseada em política de controle de acesso diversas, como Controle de Acesso Baseado em Função (RBAC), Controle de Acesso Baseado em Atributos (ABAC) entre outros. O anexo C mostra detalhes destes controles de acesso.

4 Ataques de Autenticação

Como já foi discutido, o processo de autenticação é a primeira medida de segurança que sistemas IoT devem implementar em todas as camadas: percepção, rede e aplicação. Vimos os desafios de implementar algoritmos de segurança robustos principalmente na camada de percepção e a dificuldade de criar uma solução genérica por conta da diversidade de protocolos da camada de rede, e por fim a interação humana descuidada na camada de aplicação. Dada a importância da Autenticação, vamos descrever como são feitos ataques para quebrar este processo e apresentar soluções.

4.1 Ataques de Força Bruta

Neste tipo de ataque o invasor testa, em alta velocidade, todas as combinações de usuário e senha utilizando programas automatizados e se valendo de senhas fracas ou padrão.

4.1.1 Alguns programas automatizados

- **Hydra**: Permite realizar ataques contra vários tipos de serviços de login (HTTP, FTP, SSH).
- **John the Ripper**: Frequentemente usado para quebrar hashes de senhas.
- **Hashcat**: Ferramenta eficiente para ataques de força bruta em senhas hash.

Uma variação do Ataque de Força Bruta é o Ataque de Dicionário, onde em vez de testar todas as combinações de usuário e senha, o atacante utiliza uma lista pré-compilada de senhas comuns. Essas listas podem ser carregadas pelos programas automatizados já citados.

4.2 Soluções para Ataques de Força Bruta/Dicionário

- **Limite de Tentativas de Login**: Implementar um limite de tentativas falhas de login (por exemplo, bloqueio da conta após 3 tentativas) pode impedir ataques de força bruta.
- **Captcha**: A introdução de um CAPTCHA após várias tentativas de login impede que scripts automatizados continuem tentando novas combinações.
- **Autenticação Multifator (MFA)**: Mesmo que o invasor consiga descobrir a senha, o MFA exige uma segunda camada de autenticação (como um código enviado ao celular ou uma chave física), tornando o acesso muito mais difícil.
- **Criptografia e Hashing de Senhas**: Sistemas que armazenam senhas de forma segura, utilizando hashes criptográficos fortes e salt (aleatoriedade adicionada ao processo de hash), tornam difícil para os invasores decifrarem as senhas mesmo que tenham acesso ao banco de dados.
- **Política de Senhas Fortes**: Encorajar os usuários a escolherem senhas longas e complexas, que incluam letras, números e símbolos, dificulta significativamente os ataques de força bruta.

4.3 Ataque de Man-in-the-Middle (MitM)

Nesse tipo de ataque, o invasor intercepta a comunicação entre dois dispositivos durante o processo de autenticação e assim ele pode capturar informações sensíveis, como senhas ou tokens e se autenticar posteriormente de

forma legítima. A intercepção em um ataque MitM é feita por meio de diversas técnicas, que podem envolver a manipulação de protocolos de rede, redirecionamento de tráfego ou exploração de vulnerabilidades.

4.3.1 Métodos de intercepção

- **ARP Spoofing (ARP Poisoning)**: Muitos dispositivos IoT operam em redes locais (LAN), como redes domésticas ou empresariais. Se um invasor ganhar acesso à rede, ele pode realizar ARP spoofing para interceptar o tráfego entre dispositivos IoT e seus servidores de controle, capturando dados sensíveis como credenciais de autenticação ou comandos.

- **DNS Spoofing (DNS Cache Poisoning)**: Dispositivos IoT frequentemente se conectam a servidores ou APIs externas, usando DNS para resolver endereços IP. Um invasor pode alterar a resolução DNS para redirecionar o dispositivo IoT para um servidor malicioso, onde ele pode capturar dados de autenticação ou manipular os comandos recebidos pelo dispositivo.

- **HTTPS Spoofing / SSL Stripping**: Muitos dispositivos IoT ainda não implementam comunicação segura por HTTPS. Um invasor pode usar SSL stripping para forçar esses dispositivos a usar HTTP, expondo o tráfego a intercepções e ataques de MitM. Mesmo dispositivos que suportam HTTPS podem estar vulneráveis se o invasor conseguir desabilitar ou manipular a segurança.

- **Wi-Fi Eavesdropping**: Muitos dispositivos IoT usam conexões Wi-Fi para se comunicar. Se a rede Wi-Fi estiver mal configurada ou não criptografada (como em redes públicas), um invasor pode interceptar o tráfego entre o dispositivo e o servidor, capturando informações sensíveis. Mesmo em redes criptografadas, técnicas como ARP spoofing podem ser usadas para facilitar a escuta.

- **Ataque de Proxy Malicioso**: Dispositivos IoT que dependem de proxies para comunicação com servidores externos podem ser enganados a usar um proxy malicioso controlado pelo invasor. Isso pode permitir que o atacante intercepte, manipule ou bloqueie as comunicações entre o dispositivo e o servidor legítimo.

- **Ataque de Man-in-the-Browser (MitB)**: Embora o ataque de MitB seja mais comum em navegadores web, ele também pode ser relevante para dispositivos IoT que possuem interfaces de gerenciamento acessíveis via web. Se um atacante puder comprometer o navegador usado para acessar a interface de um dispositivo IoT, ele pode capturar credenciais ou comandos enviados ao dispositivo.

- **Session Hijacking (Sequestro de Sessão)**: Dispositivos IoT muitas vezes utilizam sessões para manter a comunicação com servidores ou outros

dispositivos. Se um invasor conseguir capturar um token de sessão válido, ele pode assumir o controle da sessão e enviar comandos maliciosos para o dispositivo ou servidor, comprometendo a segurança do sistema.

4.4 Soluções para Ataque de Man-in-the-Middle

- **Criptografia Forte (TLS/SSL)**:Utilizar protocolos de criptografia robustos, como Transport Layer Security (TLS) ou Secure Sockets Layer (SSL). Quando os dados são criptografados com TLS, mesmo que um invasor intercepte o tráfego, ele não poderá ler ou modificar o conteúdo sem a chave de criptografia correta. O uso de certificados digitais autentica as partes envolvidas na comunicação.

  Todos os dispositivos IoT que se comunicam com a internet ou com outros dispositivos devem usar HTTPS (que é baseado em TLS) para proteger a comunicação. Além disso, a troca de dados entre dispositivos também deve ser criptografada com TLS.

- **Certificados Digitais e PKI (Infraestrutura de Chave Pública)**: Usar certificados digitais emitidos por uma Autoridade Certificadora (CA) confiável para autenticar dispositivos e servidores. Certificados digitais permitem que os dispositivos verifiquem a identidade da outra parte com base em uma cadeia de confiança, prevenindo que um invasor se posicione entre eles sem ser detectado.

  Em dispositivos IoT, a integração de PKI e certificados digitais pode garantir que apenas dispositivos autenticados possam se conectar a redes ou servidores.

5 Conclusão

Neste trabalho foram apresentados os principais tipos de ataques que podem ocorrer em cada camada da arquitetura IoT (percepção, rede e aplicação) e as soluções que visam mitigar esses ataques. Embora existam diversas abordagens para minimizar vulnerabilidades, a natureza da arquitetura IoT impõe desafios para a implementação efetiva dessas soluções. Entre os principais obstáculos estão a limitada capacidade de processamento e armazenamento dos dispositivos, a alta escalabilidade da rede e a necessidade de comunicação em tempo real. Esses fatores impedem a adoção de mecanismos de segurança robustos, comuns em sistemas tradicionais, exigindo o desenvolvimento de soluções mais leves, adaptativas e eficientes para garantir um nível adequado de proteção em ambientes IoT.

A arquitetura IoT como está hoje foi criada utilizando recursos de tecnologias já existentes e que não foram projetadas para este ambiente, é natural que novos protocolos sejam desenvolvidos para atender o universo IoT de forma mais robusta, dado o grande crescimento de dispositivos conectados previstos para o futuro.

A Anexo: Principais Primitivas Criptográficas

Este anexo descreve as principais primitivas criptográficas utilizadas em sistemas de segurança.

Funções Hash

Transformam uma entrada de tamanho variável em uma saída de tamanho fixo, criando uma "impressão digital"dos dados. São amplamente usadas para garantir a integridade das informações. **Exemplo**: SHA-256.

Cifra Simétrica (ou de chave secreta)

Usa a mesma chave tanto para cifrar quanto para decifrar os dados. A segurança depende da manutenção da chave em segredo. **Exemplo**: AES (Advanced Encryption Standard).

Cifra Assimétrica (ou de chave pública)

Utiliza um par de chaves: uma chave pública para cifrar e uma chave privada correspondente para decifrar. Isso permite uma comunicação segura sem a necessidade de compartilhar uma chave secreta. **Exemplo**: RSA, ECC (Criptografia de Curvas Elípticas).

Assinaturas Digitais

Garantem a autenticidade e a integridade dos dados. Elas permitem que o autor de uma mensagem prove sua autoria usando uma chave privada para assinar a mensagem, que pode ser verificada com a chave pública correspondente. **Exemplo**: RSA com SHA-256.

Geradores de Números Aleatórios Criptograficamente Seguros

Geram números ou chaves aleatórias de forma imprevisível, um componente essencial para várias primitivas criptográficas, garantindo que as chaves e outros valores sejam seguros e não previsíveis.

HMAC (Hash-based Message Authentication Code)

Usa uma função hash em combinação com uma chave secreta para garantir a integridade e autenticidade de uma mensagem. **Exemplo**: HMAC-SHA256.

B Anexo: Principais Métodos para Verificar a Identidade da Entidade Autenticadora

Este anexo descreve os principais métodos usados para verificar a identidade da entidade autenticadora em sistemas de autenticação.

1. Certificados Digitais (PKI)

Certificados X.509: Em muitos sistemas, a entidade autenticadora possui um certificado digital, geralmente emitido por uma Autoridade Certificadora (CA) confiável, que atesta sua identidade.

Como funciona: A entidade autenticadora apresenta seu certificado digital para o dispositivo ou cliente. O dispositivo verifica se o certificado é válido e confiável, conferindo se foi emitido por uma CA confiável, se não está expirado e se a cadeia de confiança é válida.

Exemplo: Quando você acessa um site seguro via HTTPS, o servidor (entidade autenticadora) apresenta seu certificado SSL/TLS para que o navegador possa verificar sua autenticidade.

2. Chaves Públicas e Criptografia Assimétrica

A entidade autenticadora pode ser verificada usando pares de chaves públicas e privadas. Durante o processo de autenticação, a entidade autenticadora assina digitalmente um desafio ou uma mensagem com sua chave privada.

Como funciona: O dispositivo que deseja verificar a autenticadora usa a chave pública da entidade (geralmente fornecida anteriormente ou obtida de forma segura) para validar a assinatura. Se a assinatura for verificada, a autenticidade da entidade é confirmada.

Exemplo: Em IoT, um dispositivo pode validar a identidade de um servidor autenticador que assina uma mensagem com sua chave privada, que o dispositivo pode verificar usando a chave pública.

3. Mutual TLS (mTLS)

Em sistemas de **autenticação mútua**, tanto o cliente quanto a entidade autenticadora (servidor) precisam apresentar certificados digitais.

Como funciona: O dispositivo (cliente) e a entidade autenticadora (servidor) realizam uma troca mútua de certificados e, ambos, verificam a autenticidade um do outro. Isso garante que o cliente está se conectando ao servidor certo, e o servidor também valida a identidade do cliente.

Exemplo: Usado em redes IoT de alta segurança, onde tanto os dispositivos quanto os servidores autenticadores devem autenticar-se mutuamente antes de permitir qualquer comunicação.

4. Tokens de Autenticação Assinados (JWT)

Alguns sistemas usam **Tokens de Acesso** assinados, como **JSON Web Tokens (JWT)**, para verificar a autenticidade da entidade autenticadora.

Como funciona: O servidor autenticador assina um token com sua chave privada e envia-o ao dispositivo. O dispositivo verifica a assinatura com a chave pública do servidor, garantindo que o token é legítimo e não foi alterado.

Exemplo: Em APIs seguras ou em sistemas de autenticação M2M, o dispositivo pode verificar a entidade autenticadora que emite um token JWT assinado.

5. HMAC (Hash-based Message Authentication Code)

HMAC é uma técnica de autenticação baseada em uma chave secreta compartilhada entre a entidade autenticadora e o dispositivo.

Como funciona: A entidade autenticadora gera um código HMAC de uma mensagem ou desafio, usando uma chave secreta. O dispositivo, que também conhece essa chave, pode gerar seu próprio HMAC da mensagem recebida e comparar com o HMAC recebido, verificando a autenticidade da entidade autenticadora.

Exemplo: Usado em ambientes onde uma chave compartilhada pode ser mantida em segredo entre duas partes confiáveis.

6. Autenticação com Provas de Conhecimento Zero (Zero-Knowledge Proofs)

Esse método permite que uma entidade autenticadora prove sua identidade sem compartilhar diretamente suas credenciais (como uma senha ou chave secreta).

Como funciona: A entidade autenticadora demonstra que possui conhecimento de uma informação secreta (como uma chave privada), sem jamais divulgar essa informação ao dispositivo que está verificando sua identidade.

Exemplo: Utilizado em sistemas criptográficos avançados e blockchain, permitindo uma verificação altamente segura sem expor dados sensíveis.

C Anexo: Principais Controles de Acesso no Processo de Autorização em IoT

Este anexo descreve os principais modelos de controle de acesso utilizados no processo de autorização em sistemas IoT.

1. Controle de Acesso Baseado em Função (RBAC)

- **Descrição**: As permissões são atribuídas a entidades com base em suas funções dentro do sistema. - **Como funciona**: Cada dispositivo, usuário ou serviço recebe uma função específica (e.g., administrador, usuário comum) que define o conjunto de permissões que ele possui. As políticas de controle de acesso associadas a essas funções determinam o que a entidade pode ou não fazer. - **Exemplo**: Um sensor IoT de um edifício inteligente pode ter permissões diferentes dependendo se está classificado como um dispositivo de monitoramento de temperatura ou um sensor de segurança.

2. Controle de Acesso Baseado em Atributos (ABAC)

- **Descrição**: O acesso é concedido com base em atributos da entidade solicitante, do recurso ou do ambiente. - **Como funciona**: As políticas de controle de acesso consideram atributos como identidade, localização, tempo, estado do dispositivo, entre outros. O acesso é permitido se os atributos da entidade solicitante cumprirem as regras definidas. - **Exemplo**: Um dispositivo IoT pode acessar dados de um sensor apenas se estiver em uma determinada localização geográfica ou se o ambiente estiver em um estado específico.

3. Controle de Acesso Discricionário (DAC)

- **Descrição**: O proprietário do recurso controla quem pode acessá-lo e quais permissões são atribuídas. - **Como funciona**: Cada proprietário (usuário ou dispositivo) define as permissões de acesso ao recurso que possui. É um modelo flexível, mas pode ser menos seguro em ambientes complexos. - **Exemplo**: Em um ambiente de IoT doméstico, o proprietário de uma câmera de segurança pode decidir que um visitante tenha acesso temporário às gravações por um período limitado.

4. Controle de Acesso Obrigatório (MAC)

- **Descrição**: O acesso é regulado por políticas de segurança centralizadas, que não podem ser modificadas pelos proprietários dos recursos. - **Como funciona**: Uma autoridade central define e aplica as políticas de acesso de forma rigorosa, onde as permissões são concedidas com base em classificações de segurança e regras estabelecidas. - **Exemplo**: Em uma rede IoT militar, dispositivos e usuários recebem níveis de segurança específicos, e apenas aqueles com um nível igual ou superior ao do recurso podem acessá-lo.

5. Controle de Acesso Baseado em Blockchain

- **Descrição**: As permissões de acesso são gerenciadas de forma descentralizada, usando contratos inteligentes e registros imutáveis no blockchain. - **Como funciona**: As transações de autorização e os direitos de acesso são registrados em um ledger distribuído. O acesso é controlado e verificado por meio de contratos inteligentes que automatizam as permissões de forma segura e transparente. - **Exemplo**: Em um sistema de IoT distribuído, como uma rede de energia inteligente, o blockchain pode garantir que apenas dispositivos autorizados possam interagir com a rede elétrica.

6. Autorização Baseada em Contexto (Context-aware Authorization)

- **Descrição**: O acesso é concedido com base no contexto atual, como tempo, localização ou status do dispositivo. - **Como funciona**: A autorização só é concedida se certos requisitos contextuais forem atendidos, como o horário do dia, a posição geográfica do dispositivo ou o estado atual do sistema. - **Exemplo**: Um dispositivo IoT de segurança pode acessar dados de sensores apenas durante horários pré-definidos ou se estiver localizado dentro de uma área segura.

Glossário de Ataques em IoT

A

- **Accessibility**: Refere-se à capacidade de garantir que o sistema e os dados estejam disponíveis quando necessário. Falhas de acessibilidade podem ocorrer devido a ataques de negação de serviço (DoS) ou falhas no sistema.
- **Access Control**: Controle de acesso aos dados e aos dispositivos IoT, garantindo que apenas usuários e dispositivos autorizados possam interagir com o sistema.
- **Alteration**: Modificação não autorizada de dados ou sistemas, que pode comprometer a integridade da informação.
- **Authentication**: Processo de garantir que o usuário ou dispositivo é quem alega ser, essencial para evitar ataques de spoofing e acesso não autorizado.

B

- **Bluejacking**: Envio de mensagens não solicitadas via Bluetooth para dispositivos móveis.
- **Bluebugging**: Exploração de vulnerabilidades Bluetooth para obter controle remoto de um dispositivo.
- **Bluesnarfing**: Roubo de dados de um dispositivo via Bluetooth sem o conhecimento do proprietário.

C

- **Collisions**: Quando dois ou mais dispositivos tentam transmitir dados ao mesmo tempo na mesma rede, causando perda de pacotes de dados e potencial degradação do serviço.
- **Confidentiality**: Garantia de que a informação não é acessada por partes não autorizadas. Violações de confidencialidade podem ocorrer por meio de eavesdropping ou ataques de interceptação.
- **Consent**: A autorização de um usuário para coleta, uso e compartilhamento de seus dados. A falta de consentimento adequado pode violar privacidade.
- **Counterfeiting**: Criação de dispositivos ou dados falsificados que podem enganar o sistema IoT, comprometendo a autenticidade e integridade das operações.

D

- **Data Access Controls**: Mecanismos que regulam quem tem acesso a diferentes tipos de dados, fundamentais para a proteção contra roubo de informações.
- **Data Newness**: Garantia de que os dados são atuais e não reutilizados ou forjados por um atacante.
- **Denial of Service (DoS)**: Ataque que visa sobrecarregar um sistema ou rede IoT, tornando-os indisponíveis para usuários legítimos.
- **Disclosure of Information**: Exposição não autorizada de informações sensíveis, normalmente envolvendo quebra de confidencialidade.
- **Data Privacy**: Proteção de dados pessoais e sensíveis de usuários IoT contra acessos e compartilhamentos não autorizados.

E

- **Eavesdropping**: Interceptação passiva de dados transmitidos entre dispositivos IoT. Pode comprometer a confidencialidade de informações sensíveis.
- **Exhaustion**: Ataque que visa esgotar os recursos (como energia ou largura de banda) de um dispositivo IoT, muitas vezes como parte de um ataque de negação de serviço (DoS).

F

- **Fabrication**: Criação de informações falsas ou falsificação de dados no sistema para causar disfunções ou enganar outros dispositivos.

H

- **Hacking**: Acesso não autorizado a sistemas IoT para roubo de informações, controle indevido ou destruição de dados e funcionalidades.
- **Heterogeneity (inaccessibility to an authentic node)**: Refere-se à dificuldade de interoperabilidade entre dispositivos IoT de diferentes fabricantes ou protocolos, que podem ser exploradas por atacantes.

I

- **Identity Theft**: Roubo de informações de identificação de um usuário para realizar fraudes ou se passar por esse usuário em sistemas IoT.
- **Infrastructure Security**: Proteção da infraestrutura física e virtual que suporta a comunicação e operação dos dispositivos IoT.
- **Interception**: Captura de dados durante a transmissão sem o conhecimento dos emissores e receptores legítimos.
- **Integrity**: Garantia de que os dados não foram alterados ou manipulados durante a transmissão ou armazenamento.
- **Interruption**: Ataque que visa interromper o funcionamento de um sistema IoT, tornando-o incapaz de realizar suas operações normais.

J

- **Jamming**: Interferência intencional no sinal de comunicação entre dispositivos IoT, muitas vezes bloqueando ou degradando a conectividade.

M

- **Man-in-the-Middle Attack**: Um atacante intercepta e possivelmente altera a comunicação entre duas partes sem que elas percebam.
- **Misconfiguration**: Configuração incorreta de software ou hardware IoT, o que pode abrir vulnerabilidades que os atacantes podem explorar.

N

- **Node Authentication**: Processo de verificar a identidade de um nó (dispositivo) na rede IoT, para garantir que ele é legítimo e autorizado a participar da rede.
- **Node Failure**: Falha de um dispositivo IoT, que pode ser causada por falhas de hardware, esgotamento de energia ou ataques intencionais.
- **Node Subversion**: Quando um atacante compromete um nó da rede IoT, geralmente com o objetivo de roubar dados ou inserir informações falsas na rede.
- **Node Outage**: Quando um dispositivo IoT se torna inoperante, voluntária ou involuntariamente, o que pode comprometer o funcionamento da rede.

P

- **Passive Information Gathering**: Coleta de dados sem a interação ativa com o sistema, como a interceptação de pacotes de rede.
- **Physical Security**: Proteção física de dispositivos IoT contra danos ou roubo, muitas vezes desconsiderada, mas crucial em certos ambientes.
- **Privacy**: Proteção das informações pessoais e sensíveis de usuários e dispositivos em um ambiente IoT.

- **Protocol Tunneling**: Técnica utilizada por atacantes para encapsular um protocolo dentro de outro, frequentemente usado para contornar firewalls e outras medidas de segurança.

R

- **Repudiation**: Um usuário ou dispositivo nega ter realizado uma ação em um sistema IoT, o que pode ser usado para evitar a responsabilidade por ataques ou violações.
- **Reliability**: Refere-se à capacidade do sistema IoT de funcionar corretamente, mesmo sob condições adversas ou ataques.
- **Robustness**: Capacidade de um sistema IoT de resistir a ataques, falhas ou erros operacionais sem comprometer a integridade ou disponibilidade dos serviços.

S

- **Secure Localization**: Métodos para garantir que os dados de localização gerados pelos dispositivos IoT sejam confiáveis e não manipulados por atacantes.
- **Self-Organization**: Habilidade dos dispositivos IoT de formar e manter a rede automaticamente, o que pode ser uma vulnerabilidade se for comprometida.
- **Signal Loss**: Perda de conectividade ou sinal em dispositivos IoT, que pode ocorrer devido a interferência ou jamming.
- **Survivability**: Capacidade do sistema IoT de continuar operando ou se recuperar rapidamente após um ataque ou falha.

T

- **Tampering**: Alteração física ou lógica não autorizada de dispositivos ou dados IoT com o objetivo de comprometer o sistema.
- **Time Management**: Manter a precisão e integridade de relógios e timestamps em dispositivos IoT, crucial para a sincronização de eventos e a prevenção de ataques que exploram falhas temporais.
- **Tracking**: Rastreio de localização ou comportamento de dispositivos ou usuários IoT, o que pode comprometer a privacidade quando feito sem autorização.

U

- **Unfairness**: Ataque que visa degradar o desempenho de alguns dispositivos ou usuários em uma rede IoT em benefício de outros, causando desequilíbrio no sistema.

V

- **Viruses**: Malwares que podem se propagar em dispositivos IoT, comprometendo a funcionalidade e segurança dos sistemas.

W

- **War Dialing**: Técnica em que um atacante tenta conectar-se automaticamente a vários dispositivos IoT, buscando vulnerabilidades em conexões de rede.

Glossário de Soluções de Segurança em IoT

1. **Access Control**: Mecanismos que regulam quem pode acessar recursos em um sistema, garantindo que apenas usuários ou dispositivos autorizados tenham permissão.
2. **Data Encryption**: Processo de codificação de informações para que apenas partes autorizadas possam lê-las. Inclui algoritmos de chave não-linear.
3. **IPSec Protocol Utilization**: Protocolo de segurança que protege pacotes de dados IP, assegurando a confidencialidade e a integridade dos dados transmitidos.
4. **Cryptography Techniques to Protect Against Side Channel Attack**: Métodos de criptografia que visam proteger dados contra ataques que exploram informações disponíveis fora do sistema, como consumo de energia ou tempo de execução.
5. **Hashed-based Access Control**: Controle de acesso baseado em funções hash que garantem a autenticidade e a integridade das informações.
6. **Ciphertext Re-encryption to Hide Communication**: Técnica que permite a recriptação de dados cifrados, ocultando o conteúdo da comunicação.
7. **New Lightweight Implementation Using SHA-3 Appointed Function Keccak-f (200) and Keccak-f (400)**: Implementações leves do algoritmo de hash SHA-3, otimizadas para dispositivos com recursos limitados.
8. **Node Authentication**: Processo de verificação da identidade de um nó em uma rede IoT, assegurando que apenas dispositivos autorizados se conectem.
9. **Sensor Privacy**: Proteção da privacidade dos dados coletados por sensores, garantindo que informações sensíveis não sejam acessadas indevidamente.
10. **Message Security**: Conjunto de técnicas que protegem a integridade e a confidencialidade das mensagens trocadas entre dispositivos.
11. **Device Onboard Security**: Medidas de segurança implementadas diretamente em um dispositivo, para protegê-lo desde o momento em que é ligado.
12. **Integrations Security**: Segurança na integração de diferentes sistemas e dispositivos, garantindo que a comunicação entre eles seja segura.
13. **Pseudonym**: Técnica que utiliza nomes ou identificadores fictícios para proteger a identidade real do usuário ou dispositivo.
14. **Group Signature**: Método que permite que membros de um grupo assinem mensagens coletivamente, mantendo a identidade dos signatários em segredo.
15. **Connection Anonymization**: Processo que oculta a identidade dos dispositivos em uma rede, dificultando a rastreabilidade.
16. **One-way Trapdoor Permutation**: Função matemática que é fácil de calcular em uma direção, mas difícil de inverter sem uma chave secreta.
17. **Secret Sharing**: Técnica que divide uma informação secreta em partes, que podem ser distribuídas entre diferentes partes, garantindo que a informação não possa ser recuperada sem um número mínimo de partes.
18. **Game Theory**: Abordagem matemática que estuda interações estratégicas entre agentes, frequentemente utilizada para modelar e analisar segurança em

sistemas.

19. **Population Dynamic Model**: Modelo matemático que analisa a dinâmica de populações, podendo ser aplicado para entender comportamentos em redes de IoT.
20. **Packet Transmitting Witness**: Elemento que registra e valida a transmissão de pacotes em uma rede, garantindo a integridade da comunicação.
21. **Aggregated Transmission Evidence**: Coleta e combinação de provas de transmissões para assegurar a veracidade dos dados enviados.
22. **Cryptographic One-way Hash Chain**: Cadeia de hashes que permite verificar a integridade de dados de forma que, uma vez gerados, não podem ser revertidos.
23. **(Fully) Homomorphic Encryption**: Criptografia que permite realizar operações em dados cifrados, gerando resultados cifrados que, quando decifrados, correspondem ao resultado da operação realizada em texto claro.
24. **Zero Knowledge Proof**: Método em que uma parte pode provar a outra que possui uma informação sem revelar a própria informação.
25. **Identity Management for Confidentiality**: Sistemas que gerenciam identidades digitais, protegendo dados pessoais e garantindo acesso seguro.
26. **Encryption Schemes for Confidentiality of Communication Channels**: Métodos de criptografia que garantem a proteção dos canais de comunicação contra interceptações.
27. **Cloud-based Solutions to Establish Secure Channels Based on PKI for Data and Communication Confidentiality**: Soluções que utilizam a infraestrutura de chaves públicas (PKI) em nuvem para proteger a comunicação e os dados.
28. **Authentication, Key Agreement and Protection of User Privacy Across Heterogeneous Networks**: Conjunto de procedimentos que asseguram a autenticação de usuários e a proteção da privacidade em redes diversas.
29. **Datagram Transport Layer Security (DTLS) for End-to-End Security**: Protocolo de segurança que proporciona comunicação segura em redes baseadas em datagramas, garantindo a integridade e a confidencialidade dos dados.
30. **Information Flow Control**: Mecanismos que monitoram e controlam o fluxo de informações dentro de um sistema, prevenindo vazamentos de dados.