SentryIoTOAuth: um Provedor de Serviço de Autenticação e Autorização para Casas Inteligentes baseado no processo ACE-OAuth

Richardson B. da S. Andrade, José A. Suruagy Monteiro

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
CEP: 50740-560 – Cidade Universitária, Recife – PE – Brazil

{rbsa2,suruagy}@cin.ufpe.br

Abstract. *The Internet of Things(IoT) is present in many domains, among them the smart home. In this segment, most of the IoT applications perform cloud communication to offer various types of services for the devices. However, for the smart home scenario, through the ACE-OAuth framework, an authentication and authorization service can still be established in the LAN, making it easier to prevent cyber attacks. For this, the article presents the investigation of the proposal, the design and development of a prototype based on the ACE-OAuth framework. With this prototype in development it is intended that developers have a new practical understanding of the operation of new permission control trends for their applications*.

Resumo. *A Internet das Coisas (IoT) está presente em vários domínios, dentre eles o das casas inteligentes. Grande parte das aplicações de IoT realizam comunicação em nuvem para oferecer diversos tipos de serviços aos dispositivos. No entanto, para o cenário de casas inteligentes, por meio do framework ACE-OAuth, um serviço de autenticação e autorização pode ser estabelecido ainda na própria LAN, facilitando na prevenção contra ataques cibernéticos. Para isso, o artigo apresenta a investigação da proposta, a concepção e o desenvolvimento de um protótipo baseado no framework ACE-OAuth. Com esse protótipo, pretende-se que desenvolvedores tenham uma nova compreensão prática do funcionamento das novos processos de controle de permissão em suas aplicações*.

1. Introdução

A crescente proliferação massiva dos dispositivos da Internet das Coisas (*Internet of Things*, IoT) já tomou conta dos lares domésticos. A adesão a eles, muitos já com inteligência embarcada, possibilita maior conveniência e comodidade na rotina das pessoas. No entanto, ainda sofrem com frequência com brechas de segurança e vulnerabilidades. Por conta disso, por exemplo, o governo japonês tem a intenção de invadir alguns dispositivos de IoT, como roteadores e câmeras, das moradias dos seus cidadãos, como atividade preventiva contra aos ataques cibernéticos [DeLaOsa, 2019].

Diante disso, como grande parte dos dispositivos de IoT, têm recursos limitados, tais como memória, processamento, fonte de energia e capacidade de transmissão, é

difícil adotar técnicas que envolvam cálculos matemáticos, geração de números aleatórios e, consequentemente, o uso de algoritmos criptográficos [Styger, 2011] e, consequentemente, o uso de protocolos de segurança comumente utilizados para proteção contra ciberataques.

Nesse sentido, nasce a necessidade de buscar alternativas adequadas e mais leves e de maneira distribuída para contornar os desafios enfrentados para segurança de IoT com menor grau de complexidade. E uma dessas soluções advém das propostas de novos modelos de autenticação e autorização.

Assim sendo, na tentativa de reduzir a complexidade de segurança de IoT, novos processos de autenticação e autorização são necessários para atender novos requisitos de segurança em IoT, em específico no cenário de casas inteligentes, tendo em vista que nesse ambiente possui interações diferentes, contextos particulares e casos de usos específicos [Ed et al, 2016] .

Com este fim, a *Internet Engieneering Task Force* (IETF) propôs um processo denominado *Authentication and Authorization for Constrained Environments* (ACE), uma proposta híbrida que agrega um *framework*, tradicional na *web,* adaptado para cenário de IoT, o OAuth2, com o protocolo de comunicação CoAP (*Constrained Application Protocol*), com o objetivo de satisfazer os requisitos de autorização e autenticação em IoT [Seitz et al, 2018]. Além disso, recentes estudos, baseados nesse modelo ACE, projetam novos esquemas e implementam soluções na tentativa de melhorar a eficiência e a economia de recursos como consumo de energia e memória.

Dessa forma, diante dos processos insatisfatórios de autorização e autenticação aplicados à segurança de IoT, aliado às recomendações do modelo ACE proposto pela IETF, é que está sendo proposta a implementação de um protótipo para cenário de casas inteligentes com as seguintes características: geração e armazenamento de credenciais, notificação da solicitação ao proprietário do recurso e eventual consentimento pelo mesmo, possibilitando uma construção de uma solução prática, funcional e útil.

Este trabalho está organizado da seguinte maneira: a seção 2 apresenta os Conceitos fundamentais relacionados ao ACE-OAuth; a seção 3 apresenta os Trabalhos Relacionados; a seção 4 apresenta a Nossa Abordagem e sua implementação; a seção 5 apresenta a Demonstração referencia o código-fonte, a documentação e a apresentação em vídeo; finalmente, na seção 6, são citadas as principais considerações e os direcionamentos para trabalhos futuros.

2. Conceitos Fundamentais

A. ACE *framework*

Como descrito anteriormente, o framework ACE produzido pelo grupo de trabalho ACE, tem como referência o *framework* OAuth2 e mais quatros outros blocos dentre eles o CoAP.

B. *Tokens*

Tokens são credenciais necessárias para proteger os recursos. O *token* consiste de uma estrutura de dados que representa a permissão emitida por uma autoridade servidora para o cliente. Os *tokens* têm vários formatos e vários métodos de utilização (a exemplo o uso de propriedades criptográficas).

***Access Tokens* (AT)** são credenciais necessárias para acessar e proteger os recursos. O AT é uma estrutura de dados emitida pela entidade autorizadora para o cliente e consumida pelo servidor de recursos.

***Refresh Tokens* (RT)** são credenciais usadas para obter AT. Os RT são emitidos para o cliente pelo AS. Os RT são usados para obter um novo AT, quando o AT atual se torna inválido ou expira, ou para obter AT adicionais com escopo idêntico ou aproximado. Além disso, são intencionados apenas para uso com o AS e nunca são enviado ao RS.

***Proof-of-Possession* (PoP)** of *tokens* dado um *token* de acesso que pode se vincular a uma chave criptográfica usada pelo servidor de recurso para autenticar as requisições. Estas provas são geradas pelo cliente para demonstrar a posse de um segredo para o servidor de recursos. Além do mais, a chave PoP pode usar criptografia simétrica ou assimétrica.

C. Blocos

A especificação do *framework* ACE tem como arcabouço quatros blocos:

***Open Authentication* 2.0 (OAuth2)** [Ed & Microsoft, 2012] é um *framework* que permite que aplicações terceiras obtenham acesso limitado a um serviço, por meio do proprietário do recurso que cederá a permissão para liberar ou recusar o fluxo entre recurso e o serviço.

***Constrained Application Protocol* (CoAP)** [Shelby et al, 2014] é um protocolo leve para transferência *web*, especificamente projetado para ambientes com limitações de recursos. Esse protocolo tipicamente roda sobre UDP o que reduz o *overhead* e as trocas de mensagens.

***Concise Binary Object Representation* (CBOR)** [Bormann et al, 2013] é um formato leve para troca de dados. Esse protocolo utiliza uma codificação binária projetada para reduzir extremamente o tamanho do código e de forma amigável o tamanho da mensagem.

***CBOR-based secure message format* (COSE)** [Schaad & Cellars, 2017] é um formato de mensagem seguro compacot baseado no CBOR. COSE especifica como codificar as chaves criptográficas, código de autenticação de mensagem, conteúdo criptográfico e assinatura com CBOR [WG-NET, 2018]. Para o ACE-OAuth, o COSE tem função sobre a segurança dos *tokens* autocontidos e prova de possessão. O formato padrão do *token* é definido pelo CBOR *web* CWT que é uma declaração de algo verdadeiro para ser transferido entre duas partes [Jones, et al., 2018]. Além disso, é possível aplicar uma camada de segurança por meio do OSCORE um método para proteção na camada de aplicação do CoAP usando COSE [Selander et al., 2018], neste caso para assegurar a proteção dos *tokens*.

D. Atores

***Client* (C)** consiste de um dispositivo ou aplicação que tem a intenção de consumir algum recurso.

***Authorization Server* (AS)** é responsável pelo controle de acesso de identidade e controle de permissão sobre os recursos e fornecer o *token* para os CEs.

***Resource Server* (RS)** gerencia o acesso aos recursos para ser ofertados desde que seja concedida permissão pelo servidor de autorização AS.

***Resource Owner* (RO)** é a entidade proprietária de um recurso protegido hospedado no RS e com direito a conceder acesso a ele.

E. *Endpoints*

***Authorization Endpoint* (AE)** é suportado pelo AS, o AS envia uma notificação para o RO. RO que recebe uma solicitação de autorização sobre o recurso oferecido pelo RS. Em seguida, RO realiza uma requisição ao AS com informação do controle de permissão baseado no escopo, por meio da decisão de consentimento, homologando ou revogando, o acesso do cliente ao recurso.

***Client Endpoint* (CE)** consiste no terminal final que realiza as solicitações ao AS para obtenção do *token* e poder consumir recursos através do servidor de recursos RS.

***Introspection Endpoint* (IE)** é oferecido pelo AS e pode ser usado pelo RS se for necessária uma requisição de informação adicional relacionada a um AT recebido. Dessa forma, o RS faz uma requisição para o IE sobre o AS. O AS recebe a informação sobre ela e a processa. Se a requisição for processada com sucesso, o AT é retornado na resposta.

***Token Endpoint* (TE)** é hospedado pelo AS, o que permite o cliente realizar a requisição do AT. Dessa forma, o cliente faz uma requisição para o TE sobre o AS, depois, o AS recebe e a processa. Se a requisição for processada com sucesso, o AT é retornado na resposta.

F. Fluxo Básico do Protocolo do *Framework* ACE

Na Figura 1 é demonstrada a interação realizada entre as entidades AS, RS, Client e RO como concebido pelo IETF ACE WG. Além do mais, disso é possível observar a concentração dos *endpoints* hospedados no AS. Por fim, é notável perceber a interação do *framework* consiste em cinco etapas básicas, dentre elas duas são opcionais, que se concentram, principalmente, na comunicação entre RS e o AS.

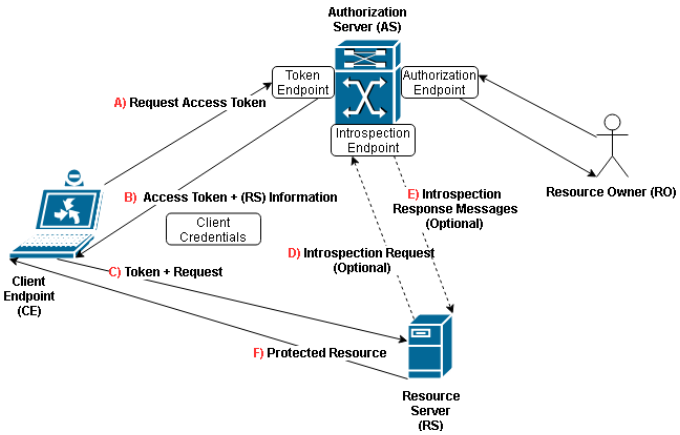

Figura 1 - Fluxo Básico do Protocolo OAuth2 no modelo proposto do ACE.
Fonte: Proprio autor.

(A) *Request Access Token* (RqAT). O cliente faz uma requisição ao *endpoint* do *token* no AS.

(B) *Response Access Token* (RsAT). Se o AS processar com sucesso a requisição do cliente, ele retorna um *token* de acesso.

C) *Request of Resource* (RR). O cliente interage com RS para solicitar acesso para o recurso protegido e oferece o *token* de acesso.

(D) *Introspection Request* (IRq). O servidor do recurso RS pode ser configurado para fazer uma introspecção realizando uma requisição para o AS, mas essa etapa é opcional.

(E) *Introspection Response* (IRs). O AS valida o *token* e retorna os parâmetros mais recentes associados com o *token* para o RS, mas essa etapa é opcional.

(F) *Protected Resource* (PR). Se a requisição do cliente for autorizada, o RS executa a requisição e retorna uma resposta com o código de resposta apropriado.

3. Trabalhos Relacionados

A. *Framework* ACE

Fremantle e Aziz (2018) propõem um modelo funcional em protótipo para IoT por meio do protocolo chamado Oauthing que permite a identificação dos usuários e dispositivos por meio de uma instância em serviços em nuvem. Segundo os autores, isso permite melhoramentos significativos de segurança e garantia mais forte de privacidade devido ao desacoplamento entre o gerenciamento da identidade e da autorização por meio do consentimento explícito do usuário. Contudo, houve incremento na latência, aumento linear no uso da memória do dispositivo e no consumo de energia em comparação à outra *baseline,* denominada *Mosquitto*.

Por outro lado, Sciancalepore et al. (2017) apresentam um *framework* denominado Oauth-IoT. Neste trabalho, os autores aplicam a ideia de um *gateway* que gerencia os dispositivos com recursos restritos e que realiza o controle de acesso de aplicações terceiras aos recursos. Além disso, o OAuth-IoT utiliza um mecanismo de *cache*, por meio do qual é esquematizado com três componentes: tabela de roteamento, diretório de recursos e tabela de dados. Dessa forma, por meio desse esquema evita-se que a requisição realize mais saltos até chegar à entidade que fornece a informação do recurso. No entanto, há a limitação no tempo de resposta quando todas as implementações foram adotadas no modelo de avaliação de desempenho.

No trabalho de [Aragon, 2018] é apresentado um perfil IPsec ao ACE-OAuth, que especifica como o cliente estabelece um canal seguro com servidor de recursos para forcar acesso autorizado por meio de controle granular sobre os recursos. Para isso, os autores consideram o protocolo IPsec e o protocolo de gerenciamento de chave IKEv2. Uma das desvantagens do estabelecimento desse perfil IPsec, é o impacto no tempo de comunicação. Além do mais, há aumento no tamanho do *token* de acesso, que em consequência disso, faz com que o tamanho da mensagem cresça, afetando na comunicação e também nas trocas de mensagens dos dispositivos com restrições de recursos.

4. Abordagem SentryIoTOAuth e Implementação

A. Funcionalidades

As seguintes funcionalidades foram implementadas:

a) Geração, gerenciamento e armazenamento do AT pela AS;
b) Verificação do AT e fornecimento aos recursos pelo RS;
c) Notificação ao proprietário do recurso e consentimento pelo mesmo;

d) Interface *web* administrativa sobre as principais entidades do domínio.

B. Cenário do SentryIoTOAuth

O cenário de atuação do SentryIoTOAuth é apresentado na Figura 2.

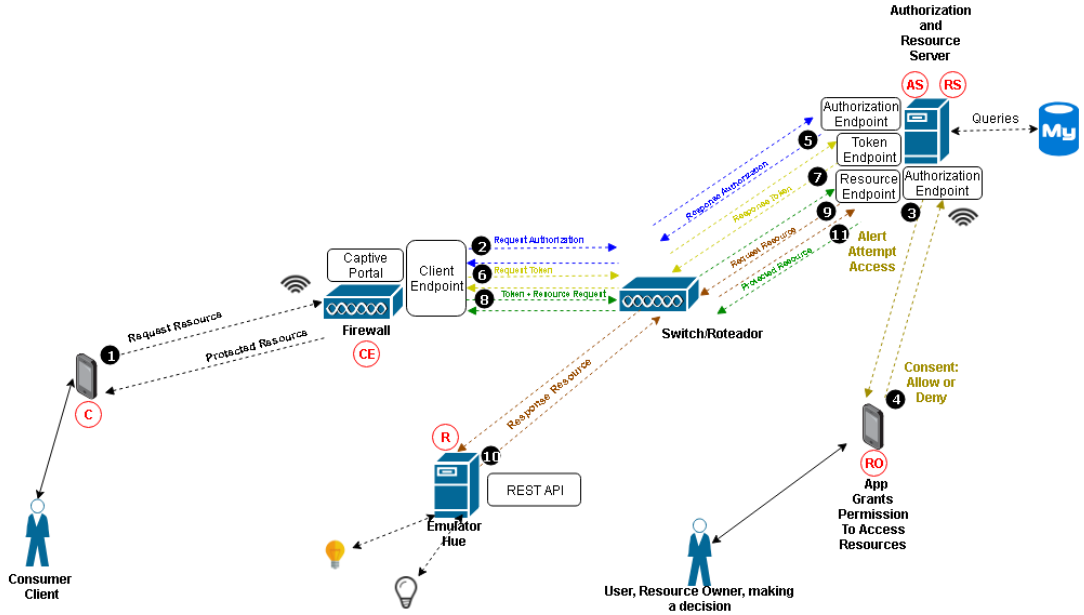

Figura 2 - Cenário de atuação do SentryIoTOAuth baseado no modelo proposto do ACE. Fonte: Próprio autor.

C. Fluxo do SentryIoTOAuth

A seguir é demonstrado o processo operacional do cenário apresentado na Figura 2.

1) Aplicação cliente consumidora (C) faz uma requisição para acessar o recurso (1) e sua requisição é encaminhada para o gateway do *firewall* ativando o serviço captive portal.
2) O CE fica posicionado no *firewall*, que recebe e agrega as requisições solicitadas pelos Cs para realizar RqAT para AS (2)
3) O AS recebe as RqAT solicitadas pelo CE e encaminha uma solicitação de permissão baseada no escopo para o RO (3), afim de liberar o AT para o CE.
4) O RO recebe a notificação na aplicação móvel enviada pelo AS e (4) toma a decisão sobre a permissão ao CE para acessar o recurso e retorna a resposta para o AS.
5) Caso tenha recebido a aprovação do consentimento pelo RO, o AS armazena no banco de dados e retorna a confirmação de autorização para o CE (5).
6) Depois da confirmação de autorização recebida pelo CE, o CE realiza uma requisição ao AT para o CE (6) e o AS envia a RsAT para o CE (7).
7) De posse do AT, o CE envia uma RR para o RS para verificar o AT (8).
8) O RS recebe o RR do CE com o AT e verifica o AT no banco de dados. Caso o AT seja válido, o RS realiza uma requisição (9), de acordo com operação solicitada pelo o CE, para o servidor (R) que emula as lâmpadas inteligentes.
9) Em seguida, servidor (R) retorna a resposta da mensagem para o RS (10). Com a mensagem recebida o RS encaminha uma mensagem de PR para CE (11).

5. Demonstração

Para o salão de ferramentas, será demonstrado o provedor de serviço realizando a autenticação, gerenciamento de *token* e autorização dos dispositivos. Assim como a realização do consentimento pelo usuário para que o cliente acesse os recursos protegidos. São disponibilizados o código fonte da implementação no repositório do GitLab[1], a documentação de apoio[2] com a especificação do ambiente real, com as tecnologias utilizadas e a instalação e configuração das ferramentas e uma *demo*[3].

6. Conclusão e Trabalhos Futuros

Este trabalho demonstrou o SentryIoTOAuth uma solução para autenticação e autorização em casas inteligentes baseado no modelo ACE-OAuth. Nosso protótipo mostrou que é possível realizar controle de permissão de maneira funcional e útil.

Os próximos passos são adicionar novas funcionalidades como aplicar chaves criptográficas, utilizar a troca de mensagens serializadas e realizar o consentimento pelo usuário por meio da Internet. Além disso, realizar experimentos em ambientes reais ou *testbed* com a utilização de outros tipos de conexão sem fio e realizar uma validação com usuário e aplicar em um caso de uso. Pretende-se ainda, realizar um teste de carga e uma análise de segurança para saber em que aspectos que a nossa solução atende os requisitos de proteção contra ataques cibernéticos no cenário da casa inteligente.

7. Agradecimentos

A pesquisa que conduziu a estes resultados recebeu o financiamento da Fundação Nacional de Ciências (NSF) dos EUA (2017-2019) e do Ministério de Ciência, Tecnologia, Inovação e Comunicação do Brasil (MCTIC) através da Rede Nacional de Ensino e Pesquisa (RNP) no projeto IoTFlows, convênio de concessão nº 002951. Este trabalho também foi financiado pela bolsa de projeto de pesquisa (IBPG-0026-1.03) da FACEPE.

Referências

Aragon, S. a. (2018). ACE of Spades in the IoT Security Game: A Flexible IPsec Security Profile for Access Control. *2018 IEEE Conference on Communications and Network Security (CNS)*.

Bormann, C. H. (Outubro de 2013). *Concise Binary Object Representation (CBOR).* Fonte: https://tools.ietf.org/html/rfc7049

Claeys, T. a. (2017). Securing Complex IoT Platforms with Token Based Access Control and Authenticated Key Establishment. *International Workshop on Secure Internet of Things (SIoT)*.

da Silva, G. C., de Mello, C., Wangham, E., Silva and Loli, M., & Bristot, S. (2018). Transposição da Autenticação Federada para uma Solução de Controle de Acesso Físico no contexto da Internet das Coisas. *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*.

[1] Código-fonte: https://gitcin.cin.ufpe.br/ace-oauth/prototype
[2] Documentação: https://gitcin.cin.ufpe.br/ace-oauth/documentation
[3] Demo video: https://www.youtube.com/playlist?list=PLp-wEbbPM1ySStQjdzEhUkeSwMqDQ1G1q

DeLaOsa, J. (02 de Fevereiro de 2019). *Japan Will Hack Into the Internet-Connected Devices of Its Own Citizens*. Fonte: https://www.ecnmag.com/news/2019/02/japan-will-hack-internet-connected-devices-its-own-citizens

Demetriou, S. a. (2017). HanGuard: SDN-driven protection of smart home WiFi devices from malicious mobile apps. *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*.

Ed, D. H., & Microsoft. (October de 2012). *The OAuth 2.0 Authorization Framework.* Fonte: https://tools.ietf.org/html/rfc6749

Ed, L. S., Ed, S. G., Selander, G., Mani, M., & Kumar, S. (january de 2016). *Use Cases for Authentication and Authorization.* Fonte: https://tools.ietf.org/html/rfc7744

Fremantle, Aziz, P. a., & Benjamin. (2018). Cloud-based federated identity for the Internet of Things. *Annals of Telecommunications*.

He, W. a. (2018). Rethinking access control and authentication for the home internet of things (IoT). *27th USENIX Security Symposium (USENIX Security 18)*.

Jones, M., Microsoft, Wahlstroem, E., Erdtman, S., Spotify, A., H, T., & ARM. (05 de 2018). *RFC 8392 - CBOR Web Token (CWT).* Fonte: https://tools.ietf.org/html/rfc8392

Porciúncula, C. B. (2018). Authentication and Authorization for Constrained Environments (ACE) com Framework OAuth e Protocolo CoAP. *Revista ComInG-Communications and Innovations Gazette*.

Rescorla, E. M. (2012). *Datagram Transport Layer Security Version 1.2.* Fonte: https://tools.ietf.org/html/rfc6347

Schaad, J., & Cellars, A. (2017). *CBOR Object Signing and Encryption (COSE).* Fonte: https://tools.ietf.org/html/rfc8152

Sciancalepore, S. a. (2017). OAuth-IoT: An access control framework for the Internet of Things based on open standards. *Computers and Communications (ISCC), 2017 IEEE Symposium on*.

Seitz, L. G. (2016). *Use Cases for Authentication and Authorization in Constrained Environments*. Fonte: https://tools.ietf.org/html/rfc7744

Seitz, L. S. (2018). *Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth) draft-ietf-ace-oauth-authz-17.* Fonte: https://tools.ietf.org/html/draft-ietf-ace-oauth-authz-16

Seitz, L., RISE, Selander, G., Ericsson, Wahlstroem, E., Erdtman, S., . . . Arm, L. (31 de Janeiro de 2019). *Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth).* Fonte: https://www.ietf.org/id/draft-ietf-ace-oauth-authz-19.txt

Selander, G., Mattsson, J., Palombini, F., Ericsson, A., Seitz, L., & RISE, S. (31 de Agosto de 2018). *Object Security for Constrained RESTful Environments (OSCORE) draft-ietf-core-object-security-15.* Fonte: https://tools.ietf.org/html/draft-ietf-core-object-security-15

Shelby, Z. H. (2014). *The Constrained Application Protocol (CoAP)*. Fonte: https://tools.ietf.org/html/rfc7252

Shelby, Z., Hartke, K., Bormann, C., ARM, & TZI, U. B. (junho de 2014). *The Constrained Application Protocol (CoAP).* Fonte: https://tools.ietf.org/html/rfc7252

Styger, E. (2011). *Introduction toSecurity and TLS.*

Tschofenig, H. (s.d.). *Analyzing the IETF ACE-OAuth Protocol.* Acesso em 2018, disponível em http://st.fbk.eu/sites/st.fbk.eu/files/osw2018-ace.pdf

WG-ACE. (2019). *Authentication and Authorization for Constrained Environments (ACE).* Fonte: https://datatracker.ietf.org/wg/ace/about/

WG-NET. (19 de Maio de 2018). *State-of-the-Art and Challenges for the Internet of Things Security*. Fonte: https://tools.ietf.org/html/draft-irtf-t2trg-iot-seccons-15