Aspectos de Segurança da Comunicação Baseada em Papéis usando WebRTC

Victor G. Netto[1], **Fábio M. Costa**[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74001-970 – Goiânia – RS – Brasil

victornetto@discente.ufg.br, fmc@inf.ufg.br

Abstract. *Model-driven approaches offer an effective way for non-IT experts to create complex software in different domains. One such domain is real-time communication, in which a high level modeling language is used to define communication sessions with non-trivial structure and behavior. Security concerns emerge naturally when implementing such a language. First, it is necessary to keep basic security requirements, such as confidentiality, integrity, and authentication. Second, it is necessary to enforce the communication constraints specified in session models. This paper presents an analysis of the security threats in the context of a communication modeling language called RBCML, which introduces a user role management layer on top of communication systems based on WebRTC. This work highlights the main threats in an RBCML implementation using WebRTC and proposes solutions for the security concerns pointed above.*

Resumo. *As abordagens dirigidas por modelos oferecem uma maneira efetiva para que pessoas não especializadas em desenvolvimento de software criem aplicações complexas em diferentes domínios. Um desses domínios refere-se à comunicação em tempo real, em que uma linguagem de modelagem de alto nível é usada para definir sessões de comunicação com estruturas e comportamentos não-triviais. O tema de segurança emerge naturalmente na implementação desse tipo de linguagem, seja na manutenção dos requisitos básicos como confidencialidade, integridade e autenticação, seja na aplicação das restrições de comunicação especificadas nos modelos de sessão. Este trabalho apresenta uma análise das ameaças presentes em uma linguagem de modelagem de comunicação denominada de RBCML, que introduz uma camada de gerenciamento de papéis de usuário em sistemas de comunicação baseados em WebRTC. O trabalho propõe um modelo de segurança para mitigar as ameaças identificadas na implementação da linguagem.*

1. Introdução

Aplicações para comunicação em tempo real (e.g., videoconferência) tornaram-se parte integrante do cotidiano e são, hoje, uma alternativa para atividades que exigiam o formato presencial [Sandhu et al. 2023, Deshmukh et al. 2023]. Para o suporte a cenários não triviais, em que ambientes de comunicação com diferentes características devem ser criados na mesma sessão (e.g., *breakout rooms*), tais aplicações tornam-se inerentemente complexas. Uma das fontes de complexidade refere-se à necessidade de garantir flexibilidade (permitindo a definição de sessões com diferentes características), eficiência

e, notadamente, segurança. Os dois primeiros itens são endereçados na proposta feita em [Vieira et al. 2020], enquanto que este trabalho busca complementá-lo ao identificar ameaças de segurança e apontar soluções.

Este trabalho concentra-se na análise do aspecto de segurança de sessões de comunicação construídas com o uso de RBCML (*Role-Based Communication Modeling Language*) [Vieira et al. 2020], uma linguagem de modelagem que permite a definição de sessões cuja estrutura e características de mídia são definidas com base nos papéis desempenhados pelos usuários conectados. Também são discutidas as garantias de segurança presentes no WebRTC [Rescorla 2021a], protocolo utilizado como base para a implementação de RBCML, assim como os requisitos de segurança adicionais impostos pela comunicação baseada em papéis.

O restante do artigo está estruturado como se segue. A Seção 2 aborda RBCML e WebRTC, servindo de alicerce para o restante do trabalho, enquanto a Seção 3 discute trabalhos relacionados. A Seção 4 apresenta o modelo de sistema e as principais ameaças. O modelo é usado na Seção 5 para identificar problemas de "segurança" em aplicações de comunicação em tempo real e baseada em papéis. Finalmente, a Seção 6 discute o estado atual do trabalho, principais limitações, próximos passos na pesquisa e trabalhos futuros.

2. Fundamentos

2.1. Comunicação baseada em papéis

A linguagem RBCML é motivada pela necessidade de facilitar a criação de sessões de comunicação em tempo real com estrutura e comportamento flexíveis, definidos com base nos papéis que usuários podem desempenhar [Vieira et al. 2020]. A RBCML é projetada para apoiar usuários *especialistas de domínio* na criação de aplicações de comunicação em tempo real. Embora o especialista de domínio não possua as habilidades necessárias para desenvolver aplicações, ele entende com profundidade as peculiaridades da sua área de atuação e, portanto, é a pessoa mais adequada para desenhar e auditar sessões de comunicação em tempo real dentro do seu campo de atuação. A RBCML permite abstrair a tecnologia, expondo um metamodelo de alto nível para especificação da constituição e características de sessões de comunicação em tempo real. Esse metamodelo é acessado através de uma interface gráfica ou textual.

Uma sessão de comunicação especificada em um modelo RBCML é descrita em termos dos papéis desempenhados pelos participantes. Um modelo RBCML define os papéis presentes na sessão, suas cardinalidades e as restrições de envio e recebimento de mídia (áudio, vídeo, texto e binário) entre os participantes que possuem esses papéis.

2.2. Implementação de RBCML com WebRTC

RBCML permite abstrair as tecnologias subjacentes ao definir sessões de comunicação. Sua implementação, no entanto, requer o uso de uma tecnologia de comunicação capaz de prover suporte para flexibilidade e eficiência na construção da estrutura da sessão, além de portabilidade e interoperabilidade para permitir seu uso em diferentes plataformas. Considerando esses requisitos, WebRTC torna-se uma escolha natural [Alvestrand 2021].

WebRTC é uma solução para comunicação *peer-to-peer* (P2P) padronizada pela *Internet Engineering Task Force* (IETF) e que usa navegadores Web como plataforma.

O documento raiz de sua especificação é o RFC 8825 [Alvestrand 2021], que define WebRTC como um conjunto de protocolos usados para comunicação multimídia e em tempo-real entre navegadores. A Figura 1 mostra os passos realizados numa conexão WebRTC. Como regra, cada passo precisa ser finalizado antes que o próximo inicie, embora existam otimizações que flexibilizam essa regra. Cada um desses passos consiste na aplicação de um conjunto de protocolos previamente existentes: **Sinalização** (SDP); **Conexão** (TURN, STUN e ICE); **Segurança** (DTLS e SRTP); e **Comunicação** (RTP e SCTP). Juntamente com a API JavaScript WebRTC implementada pelos navegadores, esses protocolos são usadas para criar sessões de comunicação.

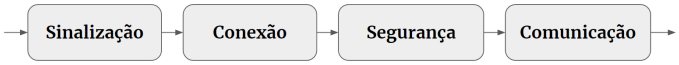

Figura 1. Passos de uma conexão P2P usando WebRTC

3. Trabalhos relacionados

[Correia et al. 2020] apresenta uma comparação entre aplicações de videoconferência com foco nos recursos pedagógicos oferecidos. O estudo conclui que muito ainda precisa ser feito para alcançar a plena experiência educacional, em parte pela ausência de controle fino na configuração das sessões. Embora o presente trabalho trate dos aspectos de segurança, tal ausência é preenchida pela adoção de RBCML, que permite a definição de sessões com configurações customizadas para cada tipo de aplicação e cenário.

Rescorla discute o tema segurança e privacidade em WebRTC nos RFCs 8826 [Rescorla 2021a] e 8827 [Rescorla 2021b], enquanto [Feher et al. 2016] traz cenários de uso e ataques reais contra WebRTC. Neste artigo, usamos estas três referências como ponto de partida para o tratamento de segurança em sessões RBCML. Ainda no tema segurança em WebRTC, [De Groef et al. 2016] põe o foco em autenticidade entre os pares, concluindo que a ausência de um ecossistema nativo de autenticação em WebRTC traz oportunidades para realização de ataques.

4. Modelo de sistema e principais ameaças

Umas das primeiras atividades na construção de softwares seguros é a *modelagem de ameaças*, que mapeia, em alto nível, as entidades que fazem parte do software e o fluxo de dados entre elas [Kohnfelder 2021]. A partir disso, é realizado um exercício para prever possíveis ameaças de segurança que podem impactar o software, tanto nas entidades do modelo quanto no fluxo de dados entre elas. Existem diversas metodologias para modelagem de ameaças (e.g., STRIDE, LINDDUN), mas sua aplicação foge ao escopo deste trabalho. Ainda assim, aqui são apresentadas as entidades e fluxos de dados (Figura 2) de uma sessão de comunicação construída com RBCML e WebRTC para servir de base para a análise aprsentada na Seção 5.

O especialista de domínio, representado pelo círculo com rótulo **E**, é responsável por criar o modelo da sessão de comunicação. O modelo possui internamente as entidades **R1** e **R2**, que representam dois papéis definidos pelo especialista de domínio. Papéis em RBCML permitem especificar as diferentes capacidades que usuários podem ter em sessões de comunicação, notadamente no que tange ao envio e recepção de diferentes

tipos de mídia. Além do especialista de domínio, o esquema ilustra dois usuários, representados pelos círculos com rótulos **A** e **B**, conectados por meio de uma instância da sessão de comunicação. A conexão é realizada via WebRTC e os dois usuários se relacionam, respectivamente, com instâncias dos papéis **R1** e **R2**. A **Aplicação**, por sua vez, representa o software de controle da sessão de comunicação que executa nos navegadores de ambos os usuários. O código dessa aplicação é gerado a partir do Modelo RBCML.

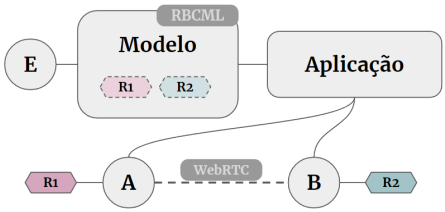

Figura 2. Modelo com as entidades e fluxos de dados de uma sessão de comunicação criada com RBCML e WebRTC

Neste trabalho, assumimos que o especialista de domínio E e o Modelo desenhado por ele são legítimos e não apresentam fragilidades de segurança. Por outro lado, os participantes da sessão de comunicação podem não ser legítimos (podem estar sujeitos a ataques de personificação), além de poderem ter intenções não-legítimas (por exemplo, ao tentarem desempenhar um papel para o qual não têm legitimidade), o que pode representar uma ameaça para a integridade da sessão.

5. Segurança em sessões RBCML

Dividimos as discussões sobre segurança em sessões de comunicação RBCML em duas categorias. A primeira diz respeito aos princípios clássicos de segurança [Kohnfelder 2021], notadamente confidencialidade e integridade. A segunda categoria trata de preocupações de segurança específicas de RBCML, oriundas de seu modelo de comunicação baseado em papéis.

5.1. Princípios clássicos de segurança

De forma genérica, *confidencialidade* significa que somente entidades autorizadas têm acesso à informação, enquanto *integridade* significa que a informação não é alterada por entidades sem autorização [Venter and Eloff 2003]. Existem outros requisitos básicos de segurança, mas eles fogem ao escopo deste trabalho. A quebra de confidencialidade ou integridade na comunicação P2P entre as entidades **A** e **B** do modelo de ameaças discutido na Seção 4 representa, portanto, uma ameça de segurança. O próximo parágrafo mostra como essa ameaça é mitigada nativamente no WebRTC.

A conexão P2P entre **A** e **B** é construída com WebRTC, herdando suas garantias de segurança. Como descrito na Seção 2, uma conexão P2P em WebRTC é estabelecida seguindo os passos da Figura 1. No passo de *sinalização*, os pares da conexão trocam entre si suas respectivas localizações na rede e credenciais são geradas e repassadas entre os pares. Essas credenciais são usadas para garantir a integridade nas operações efetuadas no passo seguinte. No passo de *conexão*, os pares realizam vários experimentos de envio e recebimento de pacotes com os endereços obtidos no passo anterior, até elegerem a configuração P2P mais viável para usar futuramente na conexão. Neste ponto, os pares

estão comunicando diretamente entre si e é possível garantir a integridade dos pacotes trocados. No passo de *segurança*, materiais criptográficos são gerados, permitindo que toda comunicação realizada no passo *comunicação* seja criptografada. Obtém-se assim as propriedades de confidencialidade e integridade na comunicação entre os pares **A** e **B**.

5.2. Requisitos de segurança específicos de RBCML

A noção de papel é ponto central da RBCML. Portanto, é fundamental garantir que os participantes possuam legitimidade para desempenhar os papéis estabelecidos no modelo de comunicação. Faz-se necessário, ainda, garantir as restrições de mídia especificadas no modelo da sessão. Por exemplo, caso o participante **P** desempenhe o papel **R**, cuja especificação define que o participante pode receber áudio e vídeo, enviar áudio mas não pode enviar vídeo, então: **1)** A aplicação deve que garantir que **P** possua exatamente o papel **R**; e **2)** A aplicação deve que garantir que o participante **P** não subverta as restrições impostas por **R** e passe a enviar vídeo em algum momento da sessão. Uma forma bem estabelecida para atender ao primeiro requisito é construir (a partir do modelo RBCML) **aplicações** que possuam mecanismos de *autenticação* e *autorização*. A autenticação solidifica o conceito de usuários da aplicação: apenas participantes previamente cadastrados na aplicação conseguem acessá-la. Já a autorização, que depende implicitamente da autenticação, cria o relacionamento participante–papel, garantindo que o participante vai desempenhar o papel apropriado na sessão de comunicação.

O atendimento do segundo requisito apontado no parágrafo anterior é, em parte, realizado pelo próprio WebRTC, que conta com uma forma nativa de impor restrições para o envio e recebimento de áudio e vídeo. A restrição é configurada durante a sinalização, momento em que os participantes trocam entre si mensagens SDP para definir os tipos de mídia que serão usadas na conexão WebRTC e os sentidos permitidos para o fluxo dessas mídias.

Ainda em relação ao segundo requisito, é necessário aplicar outro método para impor restrição de envio e recebimento de dados puros, textuais ou binários, já que o WebRTC não possui esse recurso nativamente [Rescorla 2021b]. Uma forma direta consiste em realizar esse tratamento na **Aplicação**, que é responsável por fornecer os *scripts* que ditam o comportamento da comunicação e, portanto, pode permitir ou não que um participante envie ou receba dados puros.

6. Conclusão

Neste trabalho abordamos o tema de comunicação em tempo real baseada em papéis, apresentando a linguagem RBCML de forma geral e discutindo seus aspectos de segurança. Vimos que a RBCML impõe medidas adicionais de segurança além dos princípios clássicos, como confidencialidade e integridade, que devem ser levados em consideração numa implementação segura de uma aplicação RBCML.

Como próximos passos, objetivamos implementar sessões de comunicação com RBCML e WebRTC para experimentar *de facto* os aspectos de segurança desse sistema. Dois itens serão abordados com maior atenção: ataques ao processo de sinalização (e.g., *SDP munging*) e medidas para garantir restrição no envio de dados puros, textuais ou binários usando WebRTC. Como trabalhos futuros, pode-se identificar outros cenários em que seja viável usar RBCML, bem como explorar a segurança de aplicações que usam WebRTC, incluindo aquelas em que é possível usar a linguagem RBCML.

Referências

Alvestrand, H. T. (2021). Overview: Real-Time Protocols for Browser-Based Applications. RFC 8825.

Correia, A.-P., Liu, C., and Xu, F. (2020). Evaluating videoconferencing systems for the quality of the educational experience. *Distance Education*, 41(4):429–452.

De Groef, W., Subramanian, D., Johns, M., Piessens, F., and Desmet, L. (2016). Ensuring endpoint authenticity in WebRTC peer-to-peer communication. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC 2016. ACM.

Deshmukh, R., Nand, N., Pawar, A., Wagh, D., and Kudale, A. (2023). Video conferencing using webrtc. In *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. IEEE.

Feher, B., Sidi, L., Shabtai, A., and Puzis, R. (2016). The Security of WebRTC.

Kohnfelder, L. (2021). *Designing secure software*. No Starch Press, San Francisco, CA.

Rescorla, E. (2021a). Security Considerations for WebRTC. RFC 8826.

Rescorla, E. (2021b). WebRTC Security Architecture. RFC 8827.

Sandhu, R. K., Vasconcelos-Gomes, J., Thomas, M. A., and Oliveira, T. (2023). Unfolding the popularity of video conferencing apps – a privacy calculus perspective. *International Journal of Information Management*, 68:102569.

Venter, H. and Eloff, J. (2003). A taxonomy for information security technologies. *Computers amp; Security*, 22(4):299–307.

Vieira, M. B. d. A., Carvalho, S. T., Costa, F. M., and Bromberg, D. (2020). A model-driven approach for real-time role-based communication. In *Anais XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2020)*, SBRC. Sociedade Brasileira de Computação.