Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

V. Beltran
E. Bertin
S. Cazeaux
Orange
March 9, 2015

Additional Use-cases and Requirements for WebRTC Identity Architecture
draft-cazeaux-rtcweb-oauth-identity-00

Abstract

This document discusses additional use-cases and requirements for the WebRTC identity architecture proposed by the RTCWEB working group.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2015.

Table of Contents

1. Introduction 2
2. Conventions used in this document 5
3. Some limitations of the current security architecture 5
3.1. Relationship between user and calling site 5
3.2. Relationship between user and IdP 6
3.3. Derived from the underlying identity protocol 6
4. Use cases for the security architecture 7
4.1. Call-center communication 7
4.2. Online game with voice communication 7
4.3. Enterprise video communication service 8
4.4. Wifi-based operator WebRTC service 8
4.5. Free Internet WebRTC service 9
5. Requirements for WebRTC identity provision 9
6. Security Considerations 12
7. IANA Considerations 12
8. Acknowledgements 12
9. References 12
9.1. Normative references 12
9.2. Informative references 12
Authors' Addresses 12

1. Introduction

The Real-Time Communications on the Web (RTCWEB) working group standardizes protocols for real-time communications between Web browsers. The major use cases for WebRTC technology are real-time audio and/or video calls, Web conferencing, and direct data transfer. Unlike most conventional real-time systems, (e.g., SIP-based [RFC3261] soft phones) WebRTC communications are directly controlled by some Web server, via a JavaScript (JS) API as shown in Figure 1.

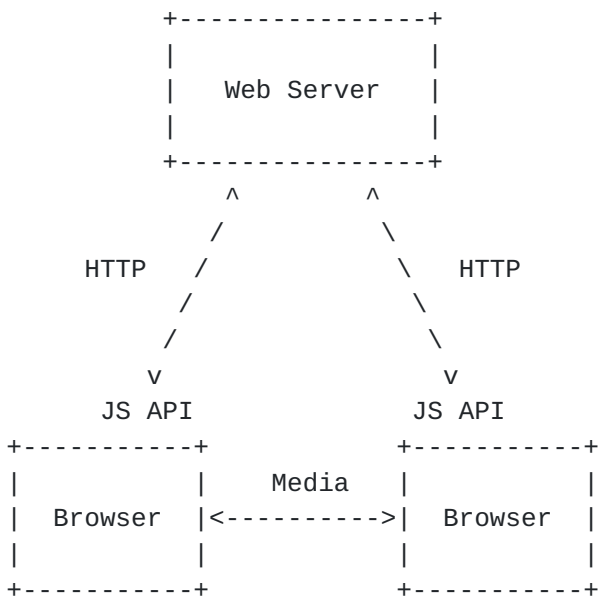

Figure 1: A simple WebRTC system

The WebRTC security architecture proposes a trust model [I-D.ietf-rtcweb-security-arch] to allow end users to identify each other (and hence to trustworthy decide to continue their communication, or not). In this model, users can directly identify each other without trusting the signaling service to which they are connected. For user authentication, Web-based identity technologies (OAuth, BrowserID, Facebook Connect) are used to provide browsers with user identities. All these technologies rely on a Web-based (i.e., HTTP/HTTPS) Identity Provider (IdP) that authenticates the user and generates his or her identity assertion. Whatever the underlying technology, the general principle is that the party which is being authenticated is the user (i.e; Alice)through his/her browser, rather than the calling service. The remote user (i.e. Bob) only needs to trust the identity assertion generated by Alices's IdP.

The WebRTC security architecture proposes that the WebRTC PeerConnection component interacts directly with the IdP, as shown in Figure 2. The general idea is that the PeerConnection component downloads JS from a specific location on the IdP dictated by the IdP domain name. The IdP can be selected by the calling service or set up in the browser. The IdP JS "endpoint", which is called IdP proxy, is used to both generate and verify user identity assertions.

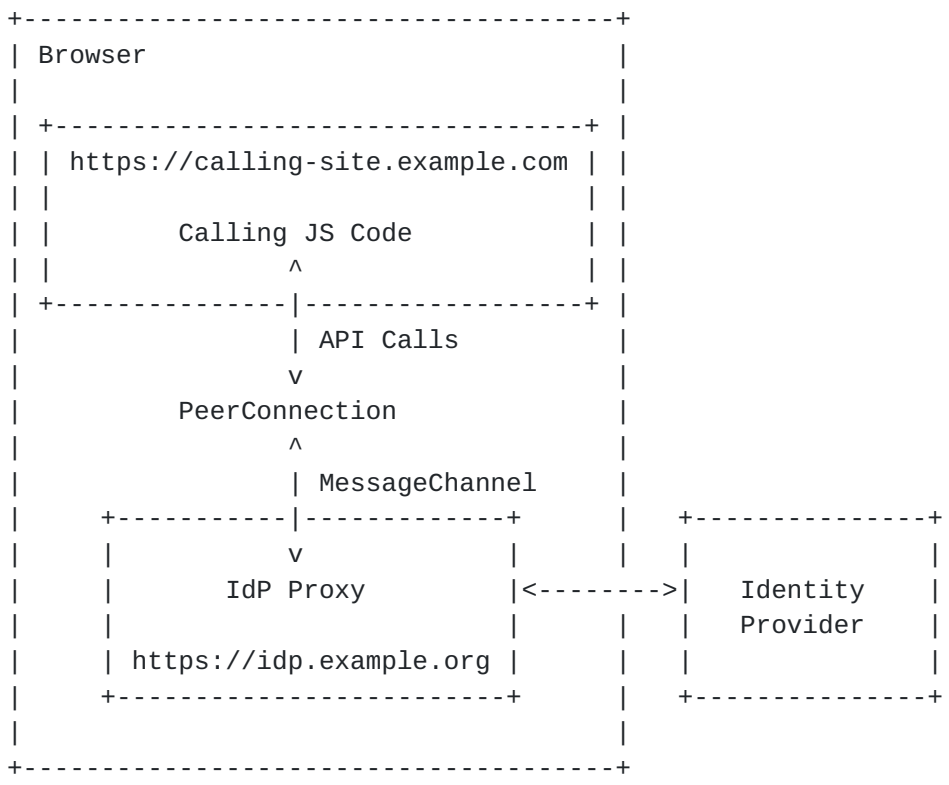

Figure 2: Web-Based Peer Authentication

When Alice wishes to call Bob, Alice's PeerConnection component instantiates the IdP Proxy of Alice's IdP. This IdP Proxy interacts with the IdP to authenticate Alice and, as a result, receive an her identity assertion from the IdP. Once the IdP Proxy replies back to the PeerConnection component with Alice's identity assertion, this component attaches the assertion to the call request to Bob. On Bob's side, when his browser receives the call request, the PeerConnection component instantiates the IdP Proxy of the IdP that generated the assertion and pass this assertion to it. This IdP Proxy communicates to the IdP to verify that the assertion was correctly generated by the IdP. The same procedure happens on the other way round. In the Bob's response to Alice, his browser attaches his identity assertion and Alice will verify this assertion against his IdP.

To associate Alice (Bob)'s identity assertion to the call that is being negotiated with Bob (Alice), her (his) assertion is bound to the call's DTLS-SRTP fingerprint. How this binding is actually done depends on the underlying SSO protocol. Two protocols are illustrated in [I-D.ietf-rtcweb-security-arch], namely, BrowserId and OAuth2.0.

In this draft, we discuss some shortcomings of this model (section 3). We introduce then different uses cases that should be covered by

the WebRTC identity architecture, along with the corresponding requirements (section 4). We finally discuss whether these requirements are fulfilled by the current WebRTC Security Architecture [I-D.ietf-rtcweb-security-arch] (section 5).

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

3. Some limitations of the current security architecture

3.1. Relationship between user and calling site

First, the calling service can see the user's fingerprint and identity assertions over the signaling path. When a user is going to make a call, he or she attaches his fingerprint and identity assertion to the call request. The callee in turn also attaches his or her fingerprint and identity assertion to the response to the request. Since the signaling path is not end-to-end encrypted, the calling service can see the users' identity assertions and fingerprints, which compromises the user privacy. On one hand, the calling service can see all the information contained in the identity assertions. On the other hand, the service can use either the fingerprint or the identity assertion to track the user's calls, as long as they are permanent across calls.

Moreover, the identity assertion is sent in clear before the end parties verify that there is not any man-in-the-middle attack (i.e., the DTLS handshake has not yet taken place). Thus, any successful man-in-the-middle attacker could get the users' identity assertions.

Authorization-oriented protocols such as OAuth2.0 can exchange authorization codes instead of identity assertions between the end points. This case also compromises user privacy since the calling service or a man-in-the-middle attacker can get the authorization code and use it against the IdP.

To avoid the calling service linking communication calls by fingerprints, the browser might be instructed to generate a separate DTLS key for each call. However, the CP is always able to link calls by identity assertions since there is no confidentiality for identity assertions.

Due to this lack of user privacy, the WebRTC security architecture assumes a non-malicious calling service (i.e., Section 6.1 in [I-D.ietf-rtcweb-security-arch]). This assumption is not however in

line with the proposed model for identity provision, which assumes that end users do not trust the calling service. If a user does not trust the calling service, this implies that this service cannot break the security or privacy of the user without his/her cooperation and knowledge. Since in the proposed model, the calling service can break the user privacy at will (by analyzing and linking identity assertions), either the model assumes that users trust the calling service or new mechanisms should be provided to protect user privacy.

3.2. Relationship between user and IdP

The WebRTC security architecture assumes that users trust their IdPs. Users should nonetheless be aware that such trust means that their IdPs can compromise their privacy. Any request that is sent to the IdP Proxy of the user's IdP contains an "origin" field that identifies the JS sending the request (i.e. the URL of the calling site). This field can be used by the IdP to identify the calling service and apply authorization policies. Thus, the user's IdP can trace the calling sites that the user visits. In addition, the called party (the party to which the user's identity assertion is sent) will communicate with the user's IdP in order to verify the user's identity assertion. Thus, the user IdP will be able to trace the sites that the user is visiting, the times at which he or she is placing a call and the called party's IP address (if the called party does not apply any mechanism for IP confidentiality). This is notably the case for calls to corporations, whose public IP address range can be easily retrieved.

3.3. Derived from the underlying identity protocol

Although the proposed identity model is aimed to be protocol-independent, the underlying identity protocol can impact user privacy. Developers and users should therefore be aware of the limitations imposed by the underlying identity protocol. We mention the main limitations of the two protocols addressed in [I-D.ietf-rtcweb-security-arch], namely BrowserID and OAuth2.0.The main limitation of BrowserID stems from its user identification scheme. It relies on public email address, and hence neither user anonymity, nor identity unlinkability are possible.

Regarding OAuth2.0, the main security limitation is the fact that the IdP has no actual way to verify who has been authorized by a user to get the user's identity assertion. This binding breaks OAuth2.0 authorization-oriented nature in several aspects. First, the IdP cannot show the user (i.e. Alice) information about who is going to receive the identity assertion (e.g., Bob). Indeed, the assertion is generated by the IdP to its own Id Proxy, instead of Bob. Second, the IdP cannot ensure that the user has authorized Bob. In fact, the IdP

does not perform any authorization policy when an authorization token to get Alice's identity is presented. Thus, the calling service (or a man in the middle attacker) may use the authorization code captured on the signaling path to get the user's identity assertion from the IdP.

4. Use cases for the security architecture

This uses cases are partly derivated from [I-D.ietf-rtcweb-use-cases-and-requirements]. In addition, motivating use cases for the enterprise market are provided, as this is one forecasted important usage of WebRTC applications.

4.1. Call-center communication

Alice is surfing the websites of several insurance or healthcare companies for information. She wishes to communicate through an WebRTC-based in-content communication tools provided by a partner of the websites (CP). Alice is concerned with her privacy. She prefers to remain anonymous if possible by avoiding any authentication system. For the websites that permit no authentication, she will not use any IdP. For the website that require authentication, she will authenticate to her IdP but she will require neither the CP, nor the IdP to be aware of the insurance or healthcare sites that she is visiting.

The requirements in this use case are:

- Anonymity by no authentication (no IdP)
- Site unlinkability by the IdP
- Caller unlinkability by the IdP

4.2. Online game with voice communication

Alice is playing poker on a gaming web site. She wishes to communicate with the other players through a voice channel. She is not registered on the gaming site as Alice but under a pseudonym: "PokerGirl". She prefers to present herself as PokerGirl to other players, instead of using her real identity at her IdP.

In this use case, she wishes to use her identity at the calling site rather than other external identity. Thus, the requirement is:

- Identity provision by the calling site

4.3. Enterprise video communication service

Alice is working for Acme Corporation. Her enterprise provides her with a comprehensive communication suite relying on a SaaS provider named "Comsforce". The Comsforce servers are thus located outside of the enterprise. Acme Corporation requires Comsforce to use the corporate IdP (based on the corporation's Active Directory server) to authenticate its employees. To this end, Comsforce obtained an API key and public key from Acme corporations. These credentials will be used by Comsforce to authenticate to Acme corporation (e.g., by creating a signature that contains the API key with the public key). Thus, when Comesforce requests the Acme corporation's IdP to authenticate a user, it needs to attach its credentials to the request.

Acme corporation's employees will have conversations of a very different nature. If the employee is a sales representative, he or she would prefer to place anonymous calls to customers while asserting that he or she is a sale representative of Acme Corporation. If the employee is a business manager contacting a colleague in the same corporation, he or she will need to present his or her corporate identity (i.e., indicating his or her name, department, etc.).

In this use case, we can see that different requirements coexist:

- Pre-determined IdP by the calling site
- Calling site authentication through credentials against IdP
- Anonymity by the IdP
- User identification

4.4. Wifi-based operator WebRTC service

A telecom operator offers a call-out communication service based on WebRTC over a Wifi network deployed by the operator. This offer is notably provided to companies that subscribe it for their employees to avoid roaming fees when traveling.

A business user is on a professional trip and is using his or her company's IdP to call colleagues and business partners. When the user authenticates, the operator's service reads the user's identity assertion to check out if the user's company has any offer subscribed for the WebRTC service. In this case, the company could have even agreed to include the payment method in the user's identity assertion. To improve the user experience, the operator WebRTC

service will detect the type of user device and it will request the IdP the most appropriate login type (e.g., simple form-based login for limited devices).

This use case is based on the trust that users give to the operator service and its network. We see the requirements:

- Site linkability
- Selection of IdP by the user
- Identity assertion analysis by the calling site
- User identification
- Selection of user login type by the calling site

4.5. Free Internet WebRTC service

Like the use case 4.4, this use case focuses on an employee that is traveling abroad for work. Conversely, this user does not find any trustworthy WebRTC service. He or she connects to a public wifi and finds a WebRTC provider on the Web. Although the employee does not trust this Web service that is new to him or her, he or she needs to communicate through it. The employee uses his or her company's IdP and requests this IdP to provide identity confidentiality. Logically, the user wishes to avoid the Web service from knowing his or her corporate identity and/or tracking his or her calls.

In this use case, user privacy against the calling site is necessary:

- Selection of IdP by the user
- Identity confidentiality

5. Requirements for WebRTC identity provision

From the section above, we can see that the WebRTC security architecture will have very different requirements based on the use case. A same use case may even require two features that are opposite as for example user identification and anonymity. The WebRTC security architecture should therefore be flexible enough to allow a dynamic configuration of features.

We outline below how the requirements are provided by the existing architecture, or not.

REQ-1 No Idp: The WebRTC API needs to be configurable for not using any IdP (i.e., sending call requests without identity assertions). This feature will provide users with anonymity by the lack of user authentication or it will allow the calling site to handle identity provision by itself.

REQ-2 Selection of IdP by the user: Users need to be able to select their IdPs. This feature is already given by default in WebRTC, as soon as the calling site does not choose a pre-determined IdP.

REQ-3 Site unlinkability requested by the user: Users need to be able to request that the IdP be not capable to know the calling sites that they visit.

REQ-4 Caller unlinkability by the IdP: Users need to be able to request that the IdP be not capable to know the IP address of the person they are communicating with.

REQ-5 Pre-determined IdP by the calling site: The calling site needs to be able to configure the IdP (or set of IdPs) that users can authenticate to. WebRTC already provides a method to do so (i.e., setIdentityProvider())

REQ-6 Calling site authentication through credentials: The calling site needs to be able to authenticate to the IdP by passing its credentials to the IdP Proxy. This can be done by including the credentials in the "origin" field of the messages sent to the IdP Proxy, as part of the calling site identifier.

REQ-7 Customization of IdP functionality by the calling site: The calling site needs to be able to pass the IdP parameters in order to customize the IdP functionality as for example the GUI for user login. This can be done by including these parameters in the "origin" field of the messages sent to the IdP Proxy. If the IdP Proxy does not support a feature required by the calling site an error message must be returned by the IdP Proxy.

REQ-8 User anonymity by the IdP: The user needs to be able to request the IdP to generate anonymous user identity assertions. Identity assertions must not include any personal identifiable information or permanent user identifier. If the IdP is not capable to generate anonymous assertions, the user should be notified. This feature is not a requirement of the WebRTC security architecture, but it is determined by the identity protocol implemented between the IdP and IdP Proxy. For example, since BrowserID relies on public email addresses, this protocol is not recommended to uses cases that require user anonymity from the IdP.

REQ-9 User identification by the calling site: The user needs to be able to request the IdP to generate identity assertions that uniquely identify his or her (i.e., no anonymous). This feature is already given by default in WebRTC.

REQ-10 Site linkability by the IdP: The IdP needs to be able to trace the calling sites that the user visits (for authorization, billing, or auditing purposes). This feature is already given by default in WebRTC, through the "origin" field of the messages sent to the IdP Proxy.

REQ-11 Identity assertion analysis by the calling site: The calling site needs to be able to analyze the content of user identity assertions. This is already allowed by the security architecture, since assertions are transmitted in clear.

REQ-12 Identity confidentiality requested by the user: Users need to be able to request the IdP to provide identity confidentiality against the calling site or any other entity different from the called party. The security architecture does not support identity confidentiality, and new methods need to be defined for these use cases.

REQ-13 Information displayed to the user: The user needs to be informed how the privacy will be handled for a specific WebRTC call that will take place with a given calling site using a given IdP: site linkabiliy or unlikability, is anonymity applied, is identity confidentiality applied.

6. Security Considerations

7. IANA Considerations

None.

8. Acknowledgements

Thanks to Xavier Marjou and Stephane Tuffin for their review.

9. References

9.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative references

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-13 (work in progress), November 2014.

[I-D.ietf-rtcweb-security-arch]
Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-10 (work in progress), July 2014.

[I-D.ietf-rtcweb-use-cases-and-requirements]
Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use-cases and Requirements", draft-ietf-rtcweb-use-cases-and-requirements-16 (work in progress), January 2015.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

Authors' Addresses

Victoria Beltran
Orange

Email: vicbelma@gmail.com

Emmanuel Bertin
Orange

Email: emmanuel.bertin@orange.com

Stephane Cazeaux
Orange

Email: stephane.cazeaux@orange.com