IJCAT - International Journal of Computing and Technology, Volume 3, Issue 3, March 2016
ISSN : 2348 - 6090
www.IJCAT.org

Restriction System for Communication of Browsers in Web-RTC

[1] Pooja Birajdar, [2] Sagar Soni, [3] Nandkishor Surashe, [4] Vishal Telsang

[1,2,3,4] Dept. of Information Technology
AISSMS Institute of Information Technology, Pune

Abstract - Communication between people is an important factor for the development of the masses. From using animals for carrying the message to using air as a medium, communication has evolved into many forms. Sending letters for communication is a passé as people can now talk through web using video conferencing which is much more preferred and convenient. The drawbacks of the traditional system of VoIP which needs to install plugin or application to perform the operation can be overcome with the help of open source technology called Web-RTC. It allows users to directly communicate with each other directly over browser without the need of plugin. However, security of Web-RTC is main concern. Web-RTC provides security mechanisms like encryption, authentication andauthorization to the data being exchanged over network. But any untrusted user can interrupt the system and can directly start the communication. This interruption can be prevented by providing restriction on the untrusted user. The user who has already registered can only start the communication with the system and its users.

Keywords **- Authentication, Authorization, Browser, Certification Authority (CA), DTLS, Identity Provider (IdP), JavaScript Signaling, Peer-to-Peer (P2P), SRTP, Web-RTC (Web Real-Time Communication), Peer-to-Peer (P2P).**

1. Introduction

Web-RTC (Web Real-Time Communication) is an Application Programming Interface (API) definition drafted by the World Wide Web Consortium (W3C).W3C supports browser-to-browser applications for voice calling, video chat, and peer-to-peer file sharing where there is no need of either internal or external plugins.Web-RTC extends the Web Browsing model as it is a new standard and industry effort. Browsers were able to exchange real-time media in a Peer-to-Peer fashion directly with other browsers, for the first time. W3C and the IETF jointly defines both - the JavaScript APIs, the standard HTML5 tags and the underlying communication protocols for setting up and management of a reliable communication channel between any pair of next-generation web browsers.Google, Mozilla and Opera, amongst others, supported the Web-RTC initiative. AWeb-RTC API that enables a web application running on any device, through the secure access to the input peripherals (such as webcams and microphones) is the primary goal to be defined. Other goals are to exchange real-time media and data with a remote party in a peer-to-peer fashion.

2. Web-RTC Architecture

The *classic* Web architecture semantics is based on a client-server paradigm, where the browsers send an HTTP request for content to the web server and the response containing the information requested is replied from the server. A URI (Uniform Resource Identifier) or URL (Uniform Resource Locator) is an entity which is closely associated with the response served by a server. The server can embed some JavaScript code in the HTML page it sends back to the client in the web application scenario. The browser interacts with such a code through standard APIs and with the users through the user interface.

The client-server semantics is extended by the Web-RTC, by introducing a peer-to-peer communication paradigm between the browsers. The so-called SIP (Session Initiation Protocol) Trapezoid helps the most general Web-RTC architecture model to draw its inspiration. A web application is run on both the browsers in the Web-RTC Trapezoid model which is downloaded from a different Web Server. The media path to flow directly between browsers without any intervening servers is configured by a *Peer Connection*. HTTP or Web Sockets, via Web Servers helps carry the signaling that can modify, translate or manage signals as needed. In Web-RTC, the signaling between the browser and the server is considered to be part of the application. Hence, it should be noted that it is not standardized. The two web servers can communicate using a standard signaling protocol such as SIP. Otherwise, they

can use a proprietary signaling protocol. The situation where both browsers are running the same web application, downloaded from the same web page is considered to be the most common scenario of Web-RTC. On the establishment of the connection between two peers, the further communication is carried out between them without the interference of the server.To monitor and provide security to the data, protocols like HTTPS and DTLS-SRTP and algorithms like SHA and AES are used.

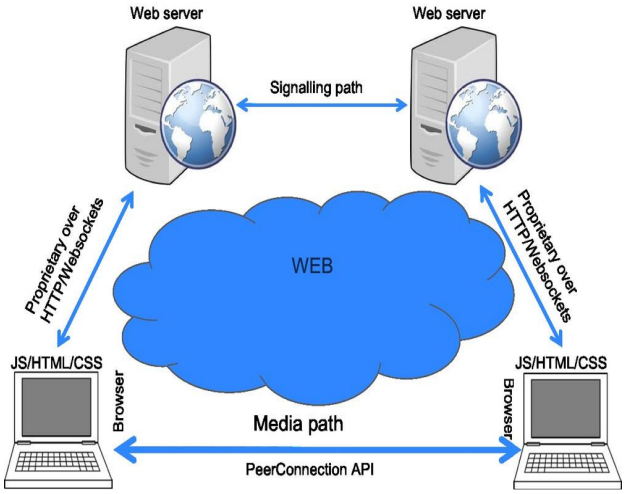

Figure 1: Working of Web-RTC Model

3. Protocols

HTTPS (Hypertext Transfer Protocol over TLS)is used to secure the data between web browser and web server during the signaling phase. This protocol makes use of HTTP with the additional functionality of TLS (Transport Layer Security) or its predecessor SSL (Secure Sockets Layer). Whenever the message is needed to be passed to the server, this protocol makes use of the standard HTTP protocol and sends the data to the server using the traditional method of HTTP. The additional functionality of security is provided by TLS protocol which is the improved version of SSL. TLS manages the encryption and authentication of the data, carried using HTTP which is not supported by this protocol. This is where the concept of digital certificates comes into picture. It is mandatory for the web server to install digital certificate issued from well-known certifying authority (CA).

This certificate consists the information of server such as host name of the server, name of the CA, digital signature, validity of the certificate and related security information. Whenever browser requests for the connection to server using HTTPS, the server first sends its digital certificate to the browser. Further the browser checks for the entry of CA in its list to check whether the certificate is issued from CA or if it is self-signed. If browser recognizes and trusts the CA it allows the further communication with server. Thus, the issue of authentication and encryption is managed by HTTPS protocol between browser and server and also prevents man-in-the-middle attack.

Once the connection is established between the two peers using server, the Web-RTC allows direct communication between end devices without the interference of the server. Now we cannot rely on HTTPS protocol for the security of the data. Hence DTLS-SRTP (Datagram Transport Layer Security – Secure Real Time Transfer Protocol) is introduced to handle this issue.The communicationsprivacy for datagram protocols is provided by the DTLS protocol. This protocol has been designed to overcome the drawback of TLS, as TLS cannot be used for datagram packets.Thus DTLS protocol is based on TLS and provides equivalent security guarantees and supports the datagram packets which is developed by slightly modifying the TLS protocol. Using DTLS, the keys are exchanged between two communicating peers in a secure way which are used for further encryption process of media channel. Also data channel is encrypted using DTLS.

The media channel which carries data like audio and video is secured using SRTP protocol. SRTP, is a protocol whichcan provide confidentiality, message authentication, and replay protection to the RTP traffic and to the control traffic for RTP. This protocol uses AES as the default encryption algorithm for encrypting media like audio and video. Thus DTLS and SRTP work hand-in-hand to provide security to media channel.

4. Literature Survey

ACLs (Access Control Lists) and CAP (Capability-based Security) [1] are the authorization models. The permissions needed by users to obtain objects is controlled by ACLs. Whereas, specific permissionon a given object using token is given using CAP.To compare the two proposed models' strengths and weaknesses, benchmark is generated for our Nubomedia prototype ACL and CAP implementations. Web Services can retrieve user identity information from a dedicated provider using Single-sign-on systems [2] (such as Facebook Connect). A most recent approach of SSO protocol like Browser-ID is used alongside OAuth2.0 and OpenID Connect.

The development of browser-based peer-to-peer web applications is enabled using ufo.js [3], which is a novel architecture. Ufo.js makes use of the W3C Web-RTC data-channel API.Security architecture to be integrated with federated Web identity systems is proposed by Web-RTC.

The identity protocols that don't satisfy Web-RTC requirements are incompatible with the security architecture. Therefore, Li Li [4] proposed two alternative architectures, Web-of-Trust model and a mirror presence system, to fill these gaps. Web identity authentication and resolution mechanisms provide the basic means for communication and to collaborate safely and efficiently. A system to create video conferencing application that is as simple as possible for the end user is proposed by R. Vapenik [5]. The creation of a pilot web based videoconferencing system was the main goal of the work. The user should be able to share its resources, such as video, audio and data. In addition, the protection of the system should be granted.A number of issues that are specific to Web-RTC enterprise usage are illustrated and discussed by Alan Johnston [6].

This paper has begun to look at enterprise requirements forpermitting Web-RTC media flows to cross enterpriseboundaries. The existing state-of-the-art Session BorderControllers will not work with Web-RTC and many of theirprinciples do not really apply to Web-RTC.The operation of outlining and discussion has been performed on a number of potential partial solution approaches. The analysis in this paper shows that while there aresome promising potential approaches, the design of a SecureEdge to permit enterprise authorization and application ofpolicy to Web-RTC traffic is far from solved today.

5. Restricting Untrusted User

The existing system of Web-RTC focuses and overcomes three important security issues:

- Secure signaling to the server
- Security of the media
- End-to end authentication

The first issue of secure signaling to the server is look upon by HTTPS protocol. Whatever messages are exchanged between the web browser and web server is totally secured by HTTPS. The security mechanisms like encryption and authentication of the data is managed by HTTPS. The second security issue of security to the media like audio and video exchanged between to peers is handled by DTLS-SRTP protocol. DTLS protocol is used to encrypt data channel and is also used to exchange keys over network. These exchanged keys are then used for encryption purpose by SRTP protocol. This protocol uses the keys to encrypt media like audio and video using the AES encryption algorithm.

Identity Provider (IdP) handles the third security issue of end-to-end authentication. IdP can be thought of as a service which provides identities to the web browsers when required. Whenever any browser wants to communicate with other browser, it first issues Id from the IdP. This process is known as Identity Generation. Then this Id is sent to the other browser prior to the communication. The other browser verifies this Id from IdP. This process is known as Identity Verification. Once both the browsers are verified, the communication starts. Thus Web-RTC provides end-to-end authentication.

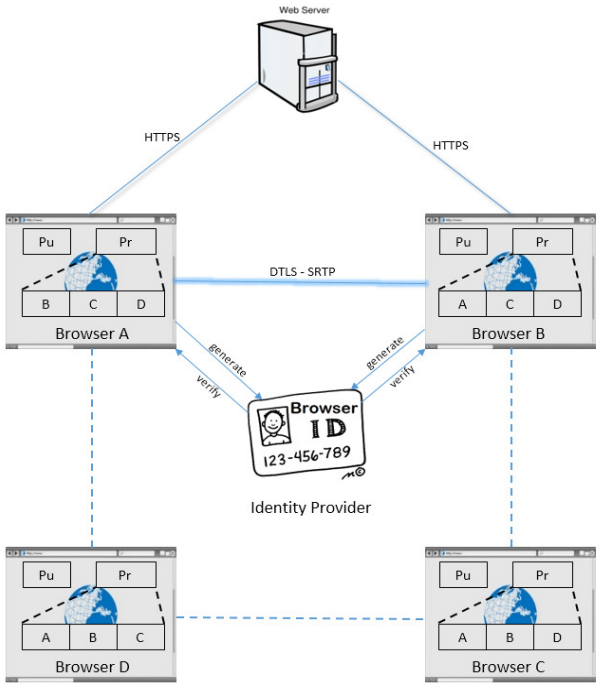

Figure 2: Concept of Public and Private Ids

There is another major problem in existing Web-RTC model. Any user can directly start communication with other users in the network.

Consider a scenario where a video conferencing technology is used by a teacher to deliver the lecture. But only those students should be allowed to join the conference who are students learning under the particular teacher. Web-RTC doesn't provide any mechanism as such. This issue is handled in the proposed system.

As shown in figure 2, each browser is slightly modified to contain public and private field (not to be confused with public and private keys).

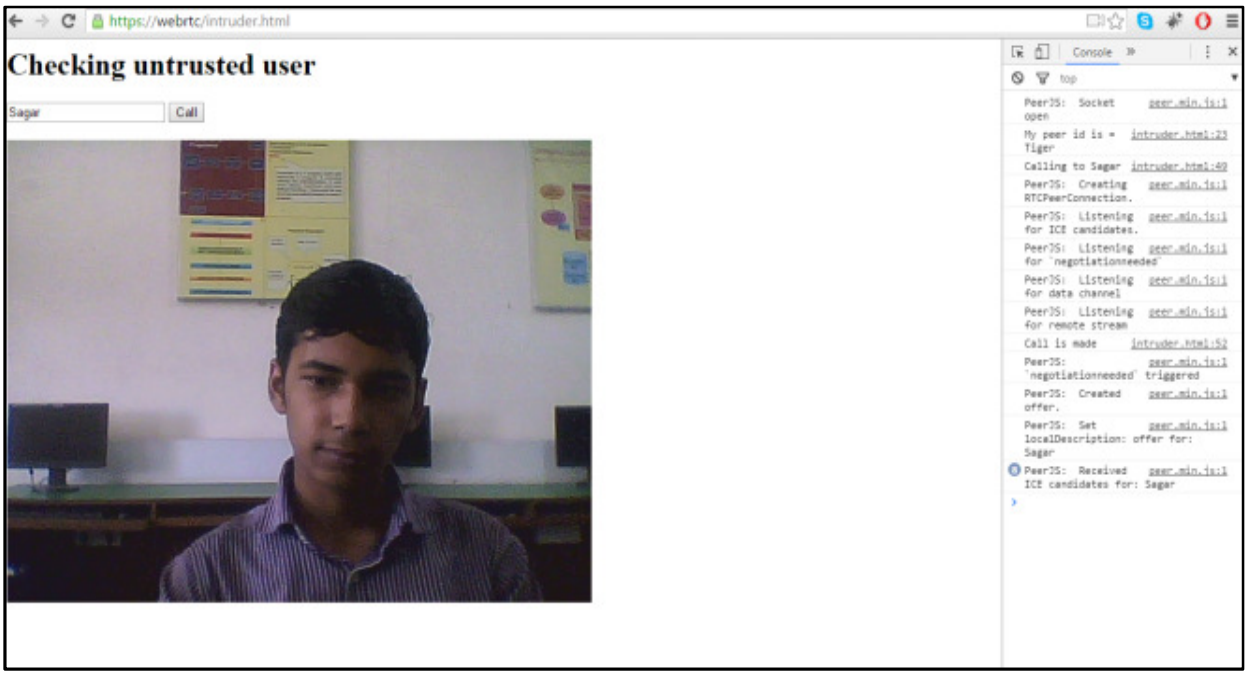

Figure 3.A: Untrusted user tring to call in the system.

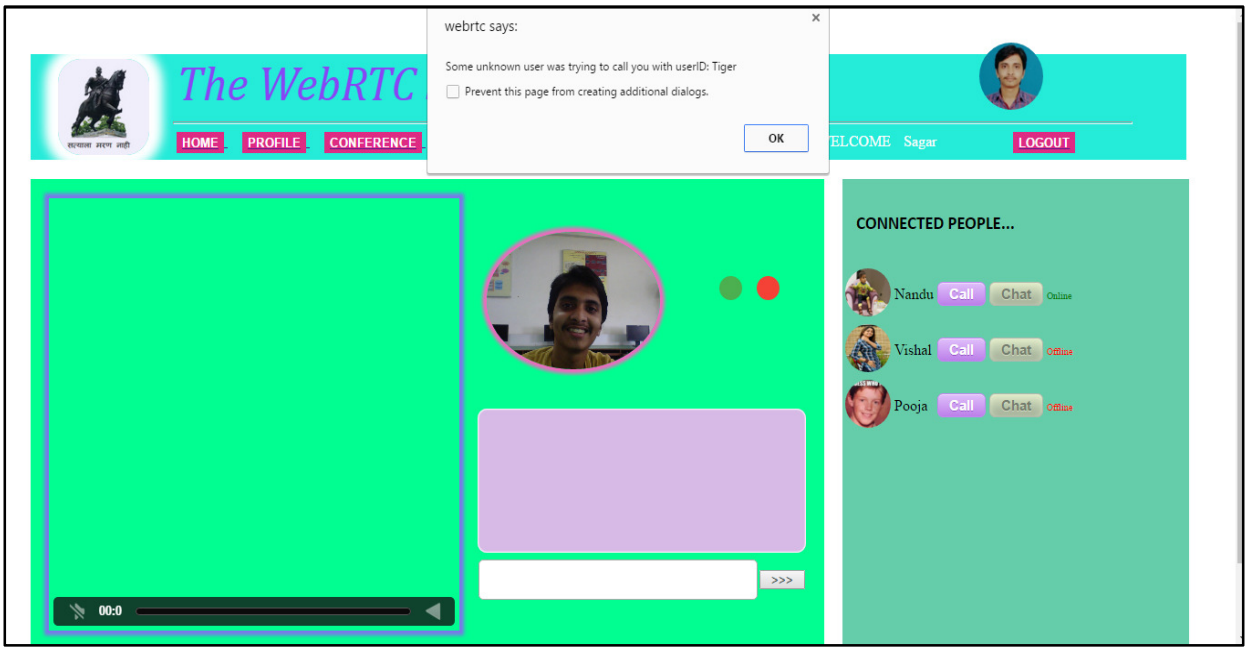

Figure 3.B: Message from the server informing about the intruder in the system

These fields perform certain functionality. Public field contains the Id of the self-browser. Whenever there is a request for Id of the browser from other browser, the public is accessed and Id is sent. On the other hand, private field contains the Ids of the other browsers in the network. So if any other browser wants to communicate with it, this browser will first scan the private field contained in it. If it finds an entry in the private field, then it allows the further communication, else it ignores the other browser. Thus, with the introduction of public and private fields, untrusted user can be restricted from communication.

Approach for achieving target:

1. Firstly, the browsers who want to communicate requests the server for a channel through the HTTPS protocol.
2. The server verifies the browsers.
3. Each browser then asks the IdPs for the keys of each other and also asks for the permission.
4. The IdP then verifies the browser and checks whether they are available for the communication in the format of peer-to-peer network.
5. If available, the Identity Provider generates and passes on the key to the browser.
6. The browsers seek the key and send it to each other for individual identification process.
7. After the verification process a secure link is established.
8. This secure link is made up by merging the DTLS and SRTP protocols.

The complete procedure can be explained in the figure 3.C below:

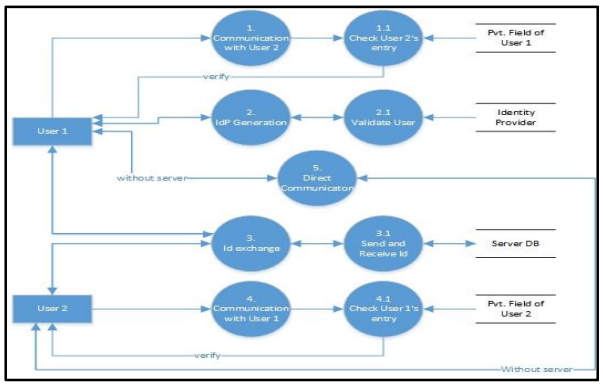

Figure 3.C: Flow of the proposed system

6. Future Scope

The near application of the project is implementation of the system in android phones as it is the most used OS on mobiles. According to the global market news, 80.3% of mobile phones sold out across the globe in 2015 were android powered.

7. Conclusions

As shown in the fig.3.A, an anonymous user to the server tries to call a user present in another system, which is prohibited by the server as depicted in fig.3.B which shows a message regarding the blocking of the intruder.

References

[1] Luis López-Fernández, MicaelGallego, and BoniGarcía-Universidad Rey Juan Carlos, David Fernández-López and Francisco Javier López-NaevaTec – "Authentication, Authorization, and Accounting in Web-RTC PaaS Infrastructures", IEEE Computer Society, November/December 2014.

[2] Victoria Beltran and Emmanuel Bertin, "User Identity for Web-RTC Services: A Matter of Trust", IEEE Computer Society, November/December 2014.

[3] Bevilacqua, P. Boemio, S.P. Romano – "Introducing ufo.js: a browser-oriented p2p network", International Conference on Computing, Networking and Communications, Internet Services and Applications Symposium, 2014.

[4] Li Li, Wu Chou, ZhihongQiu, and Tao Cai-Huawei Shannon (IT) Lab – "Who Is Calling Which Page on the Web?", IEEE Computer Society, November/December 2014.

[5] R. Vápeník, M. Michalko, J. Janitor and F. Jakab, "Secured Web Oriented Videoconferencing System for Educational Purposes Using Web-RTC Technology", 12th IEEE International Conference on Emerging eLearning Technologies and Applications, December 4-5, 2014.

[6] Alan Johnston, John Yoakum, and Kundan Singh, "Taking on Web-RTC in an Enterprise"- IEEE Communications Magazine, April 2013.

[7] WajdiElleuch, "Models for Multimedia Conference between Browsers based on Web-RTC", 2013 Sixth International Workshop on Selected Topics in Mobile and Wireless Computing.

[8] Kundan Singh and VenkateshKrishnaswamy, "A Case for SIP in JavaScript" IEEE Communications Magazine, April 2013.

Pooja Birajdar is Pursuing B.E. in Information Technology from AISSMS Institute of Information Technology 2015-16.

Sagar Soni is Pursuing B.E. in Information Technology from AISSMS Institute of Information Technology 2015-16.

Nandkishor Surashe is Pursuing B.E. in Information Technology from AISSMS Institute of Information Technology 2015-16.

Vishal Telsang is Pursuing B.E. in Information Technology from AISSMS Institute of Information Technology 2015-16.