Secure Communication in Web-RTC using Browser's Identification Process

Anuja Phapale
AISSMS Institute of Information Technology, Pune
Department of Information Technology

Pooja Birajdar
AISSMS Institute of Information Technology, Pune
Department of Information Technology

Sagar Soni
AISSMS Institute of Information Technology, Pune
Department of Information Technology

Nandkishor Surashe
AISSMS Institute of Information Technology, Pune
Department of Information Technology

Vishal Telsang
AISSMS Institute of Information Technology, Pune
Department of Information Technology

ABSTRACT

Communication between people has evolved into many forms. From pigeons to internet, there has been drastic advancement in the means of communication. Over few centuries, with enhancement in internet, people can now communicate among themselves with an ease. Web-RTC is an open source application targeting towards accrediting the web with Real-time Communication competencies. It provides services like video-conferencing, audio-conferencing, web chatting etc. It is an API that can be used for the communication from browser to browser. It is also flexible as compared to existing systems, that's why it is getting a considerable attention worldwide. So, one of the consequential dispute in Web-RTC is Security. Various security goals like authentication, confidentiality are obtained through the protocols like HTTPS, DTLS-SRTP. Untrusted users which can interrupt the system are not restricted. This interruption can be avoided by providing the restriction to the user who desires to interact in the system. The user registered in the system will only be getting the rights to connect; else user's requestwill be discarded.

General Terms

Authentication, Authorization, Browser,Certification Authority (CA), DTLS, JavaScript Signaling, SRTP, Web-RTC (Web Real-Time Communication)

Keywords

Identity Provider (IdP), Peer-to-Peer (P2P)

1. INTRODUCTION

Web-RTC (Web Real-Time Communication) is an API definition précised by the World Wide Web Consortium (W3C) in support with the browser-to-browser applications for the purpose of voice calling, video chat, and P2P file sharing without the requirement of either internal or external plugins. Web-RTC is the root term for several emergent technologies and APIs that aim to bring such communications to the Web. Although at its infant stage, Web-RTC is a technological initiative getting considerable worldwide attention. One of the biggest challenges with Web-RTC is security. Amongst others, Google, Mozilla and Opera are in the support of Web-RTC project since the time it was initially released.

2. THE EXISTING MODEL AND PROTOCOLS

When a secured site is visited, the icon of lock in the URL field indicates that anything the browser displays is approved by a server. This is a reasonable security model in two-party, client-server interactions; it's less satisfactory in more complex interactions such as email or social networking sites, where the server is intervening communications between users. In these scenarios, users must entirely dependent the server so that it doesn't fake or modify their messages.The existing Web-RTC model highlights a few necessary security mechanisms:

- signaling traffic protection (between each browser and the server)
- media traffic protection (between browsers)
- end-to-end authentication of the communicating ends and
- Protection of media streams from the JavaScript code that handles them.

2.1 HTTPS

HTTPS is used for securing connection with the server. This protocol works over HTTP and additional security is provided by Transport Layer Security (TLS). This protocol is mainly used in Web-RTC for the goal of attaining security as it provides security like authentication mechanism that helps in maintaining integrity of the message and prevents attacks like man-in-the-middle, etc. It uses the port 443 by default and works on the application layer as HTTP. This protocol encrypts everything (like headers, request response load, sensitive data) when the message is transmitted through HTTP and decrypts the message when received.

2.1.1 Involvement of Certification Authority

When the browser is connected to server via HTTPS, it maintains the message's confidentiality and prevents any intruder from accessing the data. Also the server serves its digital certificate (either generated through CA or self-signed) to the browser which allows the browser to check the authentication of the communicating server. Authentication of the client browser is optional. One can issue the certificates either from CA or individually generate them. Self-signed certificates are useful mostly for the purpose of testing or in the intra-network with limited number of users. A key which is private, exchanged between server and client using Asymmetric key cryptography which is further used for Symmetric key cryptography.

Historically, HTTPS was mainly used for tasks like online payment, e-mail, and crucial transactions in corporate environment. But recently almost all the online connections are carried out through HTTPS due to more security provided by this protocol.

2.2 DTLS-SRTP

DTLSis the protocol which provides security for the communication carried out through various datagram protocols. It allows datagram-based applications which aredesigned to restrictactive and passive attacks**.** The protocol is rooted from the stream-steeredTLS protocol and has a provision of similar security guarantees. The SRTP gives a defined profile of RTP (Real-time Transport Protocol), purposedto provide security to the data in unicast and multicast implementations. For cryptographic purpose, it utilizes AES as the default cipher.

DTLS is used for encrypting data channel and SRTP is used for encrypting media channel like audio and video in Web-RTC. DTLS-SRTP is the mechanism used by default, meaning that if an offer is reaped that supports both DTLS-SRTP and Security Descriptions for Media Streams (SDES), the former must be chosen – even if signaling is not secured. When the secret key is shared among devices in Web-RTC through SDES, the hops can see or even modify the key. Therefore, the devices have to trust the intermediate hops. But if the security of these hops is compromised then the key can be acquired by the intruder. The keypair is used to resolve this problem. The keys are exchanged through the data channel rather than signaling messages.An interactive DTLS handshake on media ports is performed at both the endpoints, before the media streaming starts**.**An extraction process of the SRTP key is done with the help of symmetric key which was shared while establishing the resulting DTLS session.Then the encrypted SRTP media stream is started.

3. THE WORKING OF SYSTEM

In Web-RTC the communication is P2P. But there is the requirement of a signaling server for initial handshake. Here signaling server is any server with any messaging mechanism, like socket.io, websockets, XHR etc., which will exchange description of session between two peers. The need of exchanging the parameters of session is, (1) to know what format the system support and what format to send and (2) exchange network information for peer-to-peer setup. Signaling server can use any messaging protocol like SIP or XMPP standard protocol. Signaling can also be done by sending JSON file. Figure 1 shows the working of the signaling mechanism in Web-RTC.

The App section in the figure 1 gets the description about the session from one peer and send it to the server which forward this description to the another peer. Once reply is received from the other peer with the description of other peer, the session descriptions are passed down to the Web-RTC in browsers. Web-RTC then setup and conductthe media via peer-to-peer.

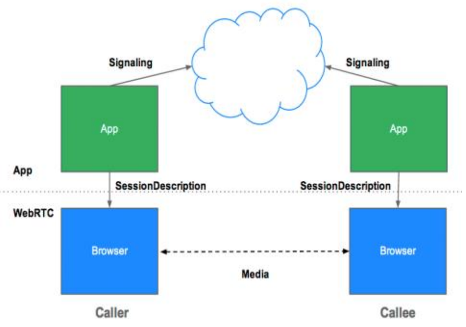

Figure 1: Working of Web-RTC and Signaling mechanism

Once the peer-to-peer connection is established it should start communicating. But it’s not true; because of NAT. NAT is the address translation mechanism of mapping a private IP with the public IP address. Two peers cannot communicate until one gets the public IP addresses of both. Now the technology comes into picture calledSTUN. STUN requests the public IP addresses of the peers and peer-to-peer connection is established. But not in every case the peer-to-peer mechanism works. So STUN server fails in some cases. To resolve this problem the technology called TURN was introduced which provides cloud fallback if peer-to-peer communication is not possible. The data is sent through servers, which uses the server’s bandwidth. It ensures that the call will work in almost any environment. Now here is STUN which is super cheap but doesn’t necessarily work and another is TURN which always works but has some cost to it. This is done by ICE which acts as a framework for connecting peers. It tries establishing the connection using STUN. If STUN fails than it turns towards the TURN. It always tries to find the best path for each call. The analysis made has indicated that the maximum number of calls uses STUN for establishing the connection rather than TURN. By statistics 86% of connection is with STUN and rest by TURN.

4. UNDERSTANDING THREAT IN THE SYSTEM AND SOLUTION TO THE THREAT

Web-RTC relies on JavaScript code from the calling site to create a UI and control the media flow. When these elements are added, the picture looks similar to a voice-over-IP (VoIP) scenario, with gateways for signaling and media, but with a twist: these gateways are represented by JavaScript code.

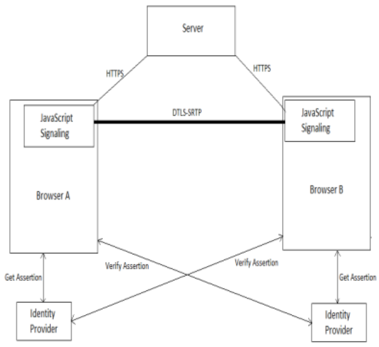

Figure 2: A real-time session mediated by JavaScript. Web-RTC relies on JavaScript code from the calling site to create a UI and control the media flow

As depicted in the system figure 2, Browser A and Browser B participate in the Web-RTC application by downloading some HTML and JavaScript from the server. The JavaScript creates the Web-RTC application by using the Web-RTC API to get and render media streams and create connections to other peers. In addition to providing the JavaScript, the server usually acts like signaling server to help the endpoints, Browsers A and B, find each other.

The responsibility of the server in this system is to establish connection between 2 peers. The HTTPS protocol gives a trust that the client communicating with the secured server. DTLS-SRTP protocols are used to secure the communication line between two peers. Security of media stream is carried out through these two protocols. Hence these protocols are used to achieve the confidentiality.

As depicted in figure 2, existing system uses two different IdPs. So, there is overhead of maintaining multiple IdPs. To deal with this overhead a single IdP is used for assertion, generation and verification. Figure 3 describes the public (Pu) and private (Pr) fields. Public field contains Id provided by IdP and the private field of particular browser contains Ids of browsers which have their entries in common IdP. For example, if browser-A wants to build connection with browser-B, then the private field of browser-B should contain Id of browser-A otherwise, it will not able to connect to browser-B. Approach for achieving target:

1. Firstly, the browsers who want to communicate, requests the server for a pathway through the HTTPS protocol.
2. The server verifies the browsers.
3. Each browser then asks the IdPs for the keys of the opposite peer and also asks for the permissions.
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 key is generated by the IdP and is sent to the browser.
6. The browsers seek the key and exchange it for individual identification process.
7. After the verification process a secure link is established.

This secure link is made up by merging the DTLS and SRTP protocols.

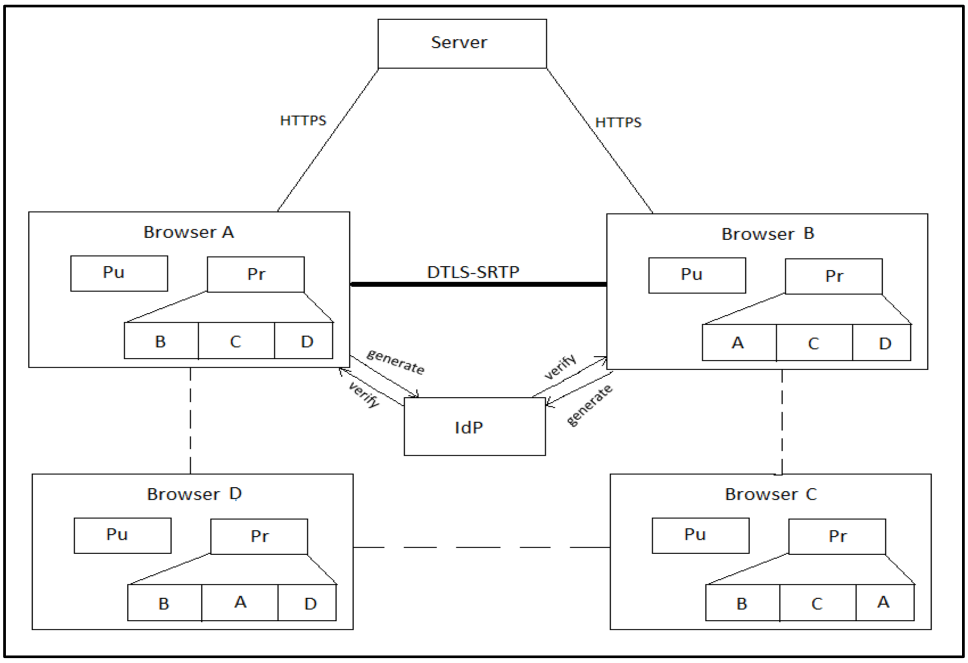

Figure 3: Extending security to restrict untrusted users

The launching of Web-RTCgave rise to the improvisation of the relevant protocols and in the IETF and APIs in W3C. The suggested system provides a service known as private field, is the strength in a way that it provides a restriction on the system and only trusted users can connect with other users. But, if a new user tends to communicate in the network which is not yet trusted by other users, there is a problem to always modify the private parts of the users in the network.

Web-RTC is a new set of technologies that will enrich the Web infrastructure and lead to new ways to create and share information. As a new platform for RTC, Web-RTC offers the opportunity to solve the end-to-end problems that have remained unsolved in email. But there are various threats in Web-RTC which aren't explored yet. The major threat is that any user can communicate or establish connection with other. Hence, there is a need of providing restriction to the users or browser.This proposed system will achieve various security goals like authentication, confidentiality, access control in Web-RTC so that a secure communication could be possible between two users.

5. ACKNOWLEDGEMENTS

We sincerely pay thanks to our guide and other staff members of the dept. of Information Technology,AISSMS IOIT.

6. CONCLUSION

Concluding that, the theory can also be appliedto Android technology, as the market of 2015 depicts 80% sales of Android phones.

7. REFERENCES

[1] Luis López-Fernández, MicaelGallego, and BoniGarcía-Universidad Rey Juan Carlos, David Fernández-López and Francisco Javier López-NaevaTec, Nov-Dec 2014, Authentication, Authorization, and Accounting in Web-RTC, PaaS Infrastructures.

[2] Victoria Beltran and Emmanuel Bertin-Orange Labs, Noël Crespi-Institut Mines-Telecom, November/December 2014, User Identity for Web-RTC Services: A Matter of Trust, IEEE Computer Society.

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

[4] Li Li, Wu Chou, ZhihongQiu, and Tao Cai-Huawei Shannon (IT) Lab,November/December 2014, Who Is

Calling Which Page on the Web?, IEEE Computer Society.

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

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

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

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