ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

Security in WebRTC

Chaitra Bhat M[1], Shabari Shedthi B[2]

II year M. Tech, Dept. of Computer Science and Engineering, NMAMIT, Nitte, Udupi district, Karnataka, India[1]

Asst Professor, Dept. of Computer Science and Engineering, NMAMIT, Nitte , Udupi district, Karnataka,India[2]

ABSTRACT: WebRTC is web real time communication. It connects two peers through their browsers. It uses HTML5 and javascript technology. WebRTC relies on web socket communication. WebRTC requires no plugins. WebRTC technology has open source nature. It causes concern regarding security in applications developed using WebRTC.

I. INTRODUCTION

WebRTC provides peer-to-peer communication opportunity. Two peers can directly connect to each other through their browsers and stream their audio, video, screen and data streams between each other. But this direct peer-to-peer connection also raises concern on security.

II. LITERATURE SURVEY

WebRTC Architecture

WebRTC implements (P2P) technology. WebRTC can be implemented through standard java script API that is inbuilt in browser. The communication between peers happens through signaling. Signaling plays role in initiating the call, connecting call between two peers. A flexibility that is provided in WebRTC is selection of signaling protocol. Signaling protocol can be chosen by developer of WebRTC application and not built in WebRTC.

IMPLEMENTING A WEBRTC APPLICATION

Three main API's used to implement WebRTC are:

getUserMedia :This API helps in accessing users camera and microphone streams. This API is invoked at first in WebRTC application.

RTCPeerConnection:WebRTC connections between peers is done by RTCPeerConnection API. The call initiation, connection are handled by this API.

RTCDataChannel:The data exchange between peers happens through this API. RTCDataChannel resembles web socket, but it takes peer to peer type while providing customizable delivary properties of the underlying protocol.

TECHNOLOGIES FOR RTC PEER CONNECTION AND RTC DATA CHANNEL API'S

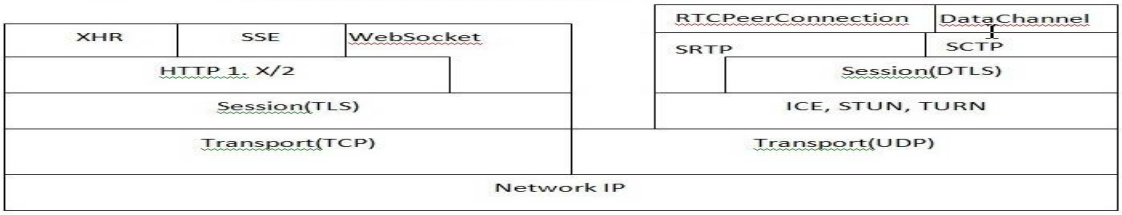

Figure 1: WebRTC protocol stack

The main protocols needed in establishment and maintenance of peer to peer connection is STUN, TURN and ICE. Peer to peer connection is usually made using UDP. To perform secure data transfer between peers DTLS is used. For security sake encryption is made mandatory. SRTP and SCTP protocols are used in WebRTC. WebRTC provides congestion control and flow control.It provides partially reliable delivery using UDP.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

SDP
Before getting connected to another user, preliminary information about the user should be known. The information to be known are audio and video codec the user supports, amount of data user can handle, data should be easily movable between clients, as amount of data to be transported is not pre decided it should support numerous types of transport protocol. SDP sends string based information to user.

SESSION TRAVERSAL UTILITIES FOR NAT
STUN is used to find connection between two peers. It helps identify each user on the internet and is used by other protocols to make connection between peers. First the initiation is done by sending request to server enabled with STUN protocol. Server identifies the IP address of client making the request and returns that to client. The client can then identify itself with the given IP address.

TRAVERSAL USING RELAY AROUND NAT
If there is firewall to be traversed, it may not allow stun based traffic to the other user. It happens in enterprises where port randomization techniques are used. In such cases TURN is used. This works by adding relay between clients that acts as peer to peer connection on behalf of client. Client gets information from TURN server. It is similar to streaming videos from one server by making request to another server.

INTERACTIVE CONNECTIVITY ESTABLISHMENT (ICE)
To get a successful route between peer connections using STUN or TURN. It searches address range for each user and tests each address in sorted order. Its stops when the combination works for both clients.

BROWSER LEVEL SECURITY
Real time communication always poses security threats. It poses threats both on network and user. This type of security vulnerability is found in all applications handling real time media or data.
WebRTC helps in building RTC applications. It provides strong and reliable infrastructure without compromising with security.

BROWSER TRUST MODEL
The WebRTC architecture is built based on a security perspective that network resources exist in hierarchy of trust. Browser is the Trusted Computing Base in users' perspective. Browser enforces all security policy that the user desires. Browser does verification of all third parties. Browser does the identity check of all authenticated entities.

SOP (SAME ORIGIN POLICY)
DOM is built on a concept that all web pages resources are fetched from the pages web server. When pages are loaded, SOP is applied on scripts to execute in isolated sandboxes of their originating domain. It prevents exchange of information between pages of different domains even iframes.

BYPASSING SOP
SOP can be bypassed by either using web sockets or allowing the browser to communicate with scripts originating server to find weather such transactions are allowed.

WEBRTC SECURITY CONSIDERATION INSTALLATIONS AND UPDATES
As webRTC applications are not installable applications there will be no malwares or virus. WebRTC applications are very safe if it is accessed through HTTPS domain Accessing HTTPS domain should be signed by a valid authenticator.
Patches for installation will need another installation step that makes application vulnerable, but in case of WebRTC patches are directly pushed on to server and user will access the latest version of WebRTC application always.

ACCESS TO SYSTEM RESOURCES
The browser can access system resources. It raises security concern regarding that web application. WebRTC over comes this by asking the user to give permission to access resources.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

But while screen sharing user should be careful while sharing knowing their extent of sharing and also in case of windows overlap user should be careful.

MEDIA ENCRYPTION AND COMMUNICATION SECURITY

Encryption is compulsory feature in WebRTC. Encryption is used in the entire WebRTC system even in signalling. All data exchanged using webRTC is securely encrypted. These encryptions are done using standard protocols.

DTLS

Datagram Transport Layer Security also called DTLS. WebRTC uses DTLS for encryption of information over data channels using DTLS. So all data sent in RTCDataChannel is secure.

SRTP

Secure Real-Time Transport Protocol also called as SRTP. RTP doesn't have any built in security mechanisms.RTP does not provide any protection or confidentiality of transmitted data. Applications using RTP relies on external mechanism for encryptions. WebRTC forbids unencrypted RTP. WebRTC utilizes SRTP for encryption of media streams.

ESTABLISHMENT OF SECURE LINK

Once the initial ICE checks have concluded; one or more secure channels will be setup between two peers. After ICE has established data channels between peers DTLS handshake is used to provide protection using encryption. In case of media channels some more precautions have to be taken.

After DTLS handshake finishes, the keys are used by SRTP for securing media channels. Both peers will then communicate using a security of DTLS key which is not known to any other malicious third party other than peers.

III. CONCLUSION

Now a day due to advancement in technology and high availability of mobile communication systems, communication has become a easier task. Also users have become aware regarding security in communication.

WebRTC is secure compared to all other VOIP communication systems. Users are ensured of security.

WebRTC is built keeping security in mind. Developers using WebRTC are enforced to maintain security in the applications built.

WebRTC enforces secure communication. WebRTC is the most secure VOIP solution. As WebRTC uses in built in security feature communication is always private. Security is now a basic feature and no longer a special feature.

WebRTC is available free to everyone and provides a reliable application.

Unfortunately, WebRTC media session have virtually no protection against Man in the Middle attack. Even though WebRTC uses DTLS to generate keys for the SRTP media session, the normal protections of TLS are not available.

REFERENCES

[1] Ravi S. Sand and Pierangela Samarthi. "Access Control: Principles and Practice." IEEE Communication Magazine September 1994.
[2] Bo Li and Jiang Chuan Liu. "Multirate Video Multicast Over the Internet An Overview." IEEE January.
[3] Pedro Rodriguez, Daniel Gallego, Javier Cervino, Fernando Escribano, Juan Quemada and Joaquin Salvachua. "Vaas: Video Conference as Service." ICST 10.41081.
[4] Luis Lopez Fernandez,Migueal Dars Diaz, Raul Bentez Mejas,Fransisco Javier Lopez and Joes Antonio Santos. "Kurento: A media server technology for convergent www/mobile real time multimedia communication supporting WebRTC." IEEE 2013.
[5]J. Muranyi, and J. Kotuliak. "Identity Mangement in WebRTC Domains." IEEE 2013.
[6] Christian Vogt, Jonas Werner and Thomas C. Schmidt. "Leveraging WebRTC for P2P content distributin in Web Browsers." IEEE 2013
[7]Luis Lopez Fernandez, Micael Gallego, Boni Garcia, David Fernandez Lopez and Fransisco Javier Lopez. "Experimenting with AAA in WebRTC Paas infrastructure: The case of kurento."
[8] Florian Rhinow, Pablo Porto Veleso,Conlos Puyelo, Stephen Berrett and Eamonn O Nvallain "P2P Live Video Streaming in WebRTC." IEEE 2014
[9] Christian Cola and Monoriu Valeun. "On multi-user. Web conference using WebRTC."