Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 12, 2013

T. Stach, Ed.
A. Hutton
Siemens Enterprise Communications
J. Uberti
Google

March 11, 2013

RTCWEB Considerations for NATs, Firewalls and HTTP proxies
draft-hutton-rtcweb-nat-firewall-considerations-00

Abstract

This document describes mechanism to enable media stream establishment in the presence of NATs, firewalls and HTTP proxies. HTTP proxy and firewall policies applied in many private network domains introduce obstacles to the successful establishment of media stream via RTCWEB. This document examines some of these policies and develops requirements on the web browsers designed to provide the best possible chance of media connectivity between RTCWEB peers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Many organizations, e.g. an enterprise, a public service agency or a university, deploy NATs and firewalls at the border to the public internet. RTCWEB relies on ICE [RFC5245] in order to establish a media path between two RTCWEB peers in the presence of such NATs/FWs. As last resort in order to cater for NAT/FWs with address and port dependent filtering characteristics [RFC4787], the peers will introduce a TURN server [RFC5766] in the public internet as a media relay. Aspects of TURN server deployment in the RTCWEB environment are also considered in [draft-ietf-rtcweb-use-cases-and-requirements]

If an organization wants to support RTCWEB such a TURN server may be located in the DMZ of the private network of that organization where it is still under administrative control.

In certain environments with very restrictive FW policies a TURN server in the public internet may not be sufficient to establish

connectivity towards the RTCWEB peer for RTP-based media [RFC3550]. Such policies can include blocking of all UDP based traffic and allowing only HTTP(S) traffic to the TCP ports 80/443. In addition access to the World Wide Web from inside an organization is often only possible via a HTTP proxy.

This document examines impact of NAT/FW policies in Section 2. Additional impacts due to the presence of a HTTP proxy are examined in Section 3.

1.1. Requirements Language

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].

2. Considerations for NATs/Firewalls independent of HTTP proxies

This section covers aspects of how NAT/FW characteristic influence the establishement of a media stream. Additional aspects introduced by the presence of a HTTP proxy are covered in Section 3.

If the NAT shows port and address dependent filtering behavior there is the need for a TURN server arises in order to establish connectivity for media streams. The TURN server will relay the RTP packet to the RTCWEB peer using UDP. How the RTP packets are sent from the RTCWEB client within the private network to the TURN server depends on what the firewall will let pass through.

Other types of NATs do not require using the TURN relay. Nevertheless, the FW rules and policies still affect how media streams can be established.

2.1. Firewall open for outgoing UDP and TCP traffic

This scenario assumes that the NAT/FW is transparent for all outgoing traffic independent of using UDP or TCP as transport protocol. This case is used as starting point for introduction of a more restrictive firewall. It presents the least critical example with respect to the establishment of the media streams.

The TURN server can be reached directly from within via the NAT/FW and the ICE procedures will reveal that media can be sent via the TURN server. The TURN client will send its media to the allocated resources at the TURN server via UDP.

Dependent on the port range that is used for RTCWEB media streams, the same statement would be true if the NAT/Firewall would allow UDP traffic for that ports only.

2.2. Firewall open only for TCP traffic

This scenario assumes that the NAT/FW is transparent for outgoing traffic only using TCP as transport protocol. This gives two options for media stream establishment dependent on the NAT's filering characteristics. Either transport RTP over TCP or contacting the TURN server via TCP.

In the first case the browser needs use ICE-TCP [RFC6544] and could launch a successful connectivity check directly to the remote endpoint.

In the second case the browser needs to contact the TURN server via TCP for allocation of an UDP-based relay address at the TURN server. The ICE procedures will reveal that RTP media can be sent via the TURN relay using the TCP connection between TURN client and TURN server.

The TURN server would then relay the RTP packets using UDP. ICE-TCP [RFC6544] is not needed in this context.

2.3. Firewall open only for TCP-based HTTP(s) traffic

In this case the firewall blocks all outgoing traffic except for TCP traffic to port 80 for HTTP or 443 for HTTPS. A TURN server listening to its default ports (3478 for TCP/UDP, 5349 for TLS) would not be reachable in this case.

However, the TURN server could still be reached when it is configured to listen to the HTTP(S) ports as well. In addition the RTCWEB clients need to be configured to contact the TURN server over the HTTP(S) ports.

3. Considerations for NATs/Firewalls in presence of HTTP proxies

This section considers a scenario where all HTTP(S) traffic is routed via an HTTP proxy. Note: If both RTCWEB clients are located behind the same HTTP proxies, we, of course, assume that ICE would give us a direct media connection within the private network. We consider this case as out of the scope of this document.

3.1. HTTP proxy with NAT/firewall open for outgoing UDP and TCP traffic

As in Section 2.1 we assume that the NAT/FW is transparent for all outgoing traffic independent of using UDP or TCP as transport protocol. Consequently, the same considerations as in Section 2.1 apply with respect to the traversal of the NAT/FW.

3.2. HTTP proxy with NAT/firewall open only for TCP traffic

As in Section 2.2 we assume that the NAT/FW is transparent only for outgoing TCP traffic Consequently, the same consideration as in Section 2.2 apply with respect to the traversal of the NAT/FW.

3.3. HTTP proxy assisted TURN server connection

3.3.1. TURN server connection via TCP

Different from the previous scenarios, we assume that the NAT/FW accepts outgoing traffic only via a TCP connection that is initiated from the HTTP proxy. Consequently, a RTCWEB client would have to use the HTTP CONNECT method in order to get access to the TURN server via the HTTP proxy. The HTTP CONNECT request needs to convey the TURN Server URI or transport address. Afterwards, the RTCWEB client could upgrade the connection to use TLS, forward STUN/TURN traffic via the HTTP proxy and use the TURN server as media relay.

If it is not possible to use HTTP CONNECT in this way, WebRTC will not work. We consider this case as out of the scope of this document.

Strictly speaking the TLS upgrade is not necessary, but using TLS would also prevent the HTTP proxy from sniffing into the data stream and provides the same flow as HTTPS and might improve interoperability with proxy servers. Some tests (done a while ago) indicated that there are DPI proxies that expect to see at least a SSL handshake and, possibly, valid SSL records. The application has the ability to control whether SSL is used by the parameters it supplies to the TURN URI (e.g. turns: vs turn:), so the decision to do TURN/TCP to port 443 versus TURN/TLS to port 443 could be left up to the application.

In contrast to using UDP or TCP for transport of STUN messages, the browser would now need to first establish a HTTP over TCP connection to the HTTP proxy, upgrade to using TLS and then switch to using this TLS connection for transport of STUN messages. It is also desirable that the browser detects the need to connect to the TURN server through a HTTP proxy automatically in order to achieve seamless deployment and interoperability. The browser should use the same proxy selection procedure for TURN as currently done for HTTP. The user or network administrator should not be required to change browser or proxy script configuration.

3.3.2. TURN server connection via UDP

If a local TURN server under administrative control of the organization is deployed it is desirable to reach this TURN server via UDP. The TURN server could be specified in the proxy configuration script, giving the browser the possibility to learn how to access it. Then, when gathering candidates, this TURN server would always be used such the RTCWWEB client application could get UDP traffic out to the internet.

4. Requirements for RTCWEB-enabled browsers

For the purpose of relaying RTCWEB media streams or data channels a browser needs to be able to

- connect to a TURN server via UDP, TCP and TLS for the purpose of relaying RTCWEB media streams or data channels

- connect to a TURN server via a HTTP proxy using the HTTP connect method

- connect to a TURN server via the HTTP(s) ports 80/443 instead of the default STUN ports 3478/5349.

- upgrade the HTTP proxy-relayed connection to the TURN server to use TLS

- use the same proxy selection procedure for TURN as currently done for HTTP

- switch the usage of the HTTP proxy-relayed connection with the TURN server from HTTP to STUN/TURN in order to relay media streams or data channels.

- to use a preconfigured or standardized port range for UPD-based media streams or data channels.

- learn from the proxy configuration script about the presence of a local TURN server and use it for sending UDP traffic to the internet.

- support ICE-TCP for TCP-based direct media connection to the RTCWEB peer.

5. Acknowledgements

The authors want to thank Heinrich Haager for all his input during many valuable discussions.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

TBD

8. References

8.1. Normative References

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

8.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B.B., and A.B. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, March 2012.

[draft-ietf-rtcweb-use-cases-and-requirements]
C. Holmberg, S. Hakansson, G. Eriksson , "Web Real-Time Communication Use-cases and Requirements ", 2012, <http://tools.ietf.org/html/draft-ietf-rtcweb-use-cases-and-requirements>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Thomas Stach (editor)
Siemens Enterprise Communications
Dietrichgasse 27-29
Vienna 1030
AT

Email: thomas.stach@siemens-enterprise.com

Andrew Hutton
Siemens Enterprise Communications
Technology Drive
Nottingham NG9 1LA
UK

Email: andrew.hutton@siemens-enterprise.com

Justin Uberti
Google
5 Cambridge Center
Cambridge, MA 02142
US

Email: justin@uberti.name