TURN Server REST API

Justin Uberti, Google
Version 0.9 (DRAFT)
March 29, 2013

This document describes a proposed standard API for allocating TURN services via HTTP, via the use of timelimited credentials. These credentials are vended by a web service, and enforced by the TURN server. This usage of time-limited credentials ensures that access to the TURN server is controlled even if the credentials cannot be kept secret, as is the case in WebRTC where the TURN credentials need to be specified in JS.

To use this mechanism, the only interaction needed between the web service and the TURN service is to share a secret key.

HTTP Interactions

A typical GET request is made with a username parameter, and optional TTL to control the lifetime of the TURN credentials. The TURN server addresses and credentials are returned as JSON, as well as a TTL parameter in time_t (seconds since 1970) format.

Since the returned credentials are time-limited, it is necessary to refresh them before they expire. This is done by simply re-issuing the original HTTP request with the same parameters; this extends the expiration time on the existing credentials.

To prevent unauthorized use, the HTTP requests can be ACLed by various means, e.g. IP address (if coming from a server), Origin header, User-Agent header, API key, etc.

Request

```
GET /?service=turn&username=<username>&ttl=<optional_requested_ttl>
```

Response

```
{
 // requested username : timestamp (time_t format)
 "username" : "foo:12334939",
 // base64(hmac(secret key, username))
 "password" : "adfsaflsjflds",
 // less than or equal to the requested TTL
 "ttl" : ttl,
 // URIs in the format of
```

```
  // http://tools.ietf.org/html/draft-petithuguenin-behave-turn-uris-03
  "uris" : [
    "turn:1.2.3.4:9991?transport=udp",
    "turn:1.2.3.4:9992?transport=tcp",
    "turns:1.2.3.4:443?transport=tcp"
  ]
}
```

WebRTC Interactions

The returned JSON is parsed and supplied when creating a PeerConnection.

```
var config = { "iceServers": [] };
for (var i = 0; i < response.uris.length; ++i) {
  var uri = response.uris[i];
  var iceServer = {
    "username":response.username,
    "credential":response.password,
    "uri":uri
  };
  config.iceServers.push(iceServer);
}
var pc = new PeerConnection(config);
```

TURN Interactions

Client

WebRTC's TURN request uses the "username" value for its USERNAME attribute, and the "password" value for the MESSAGE-INTEGRITY hash.

Server

As in RFC 5766, the TURN server MUST verify the MESSAGE-INTEGRITY using the shared secret key to validate that it matches the supplied USERNAME. It then splits the USERNAME attribute into its username and timestamp components, verifies that the timestamp is within the configured TTL, and optionally verifies that the username corresponds to a currently active user of the service (e.g. is currently logged in). If either verification fails, it SHOULD reject the request with a 401 (Not authorized) error.