Проксирование websocket трафика при публикации /воспроизведении WebRTC

В некоторых случаях, например, по требованиям безопасности, необходимоскрыть Websocket порт WCS за прокси-сервером. Рассмотрим примеры конфигурации nginx в качестве обратного прокси-сервера и соответствующие настройки WCS.

Настройка обратного прокси с Basic авторизацией для Websocket

1. Включите базовую авторизацию по имени и паролю (Basic authentication)в настройках nginx

```
    auth_basic "Restricted Area";
    auth_basic_user_file /etc/nginx/.htpasswd;
```

2. Настройте сервер на прослушивание HTTPS (публикация и воспроизведение по WebRTC в большинстве браузеров работает только по безопасному соединению)

```
    server {
                listen 443 ssl;
                ssl_certificate /etc/pki/tls/yourdomain/yourdomain.crt;
                ssl_certificate_key /etc/pki/tls/yourdomain/yourdomain.key;
                server_name wcs.yourdomain.com;
                server_tokens off;
                client_max_body_size 500m;
                proxy_read_timeout 10m;

        root         /usr/share/nginx/html;
        ...
```

3. Настройте прокси наWebsocket порт WCS (предположим, что nginx установлен на том же сервере)

```
        location /wss {
            if ($http_connection !~* "upgrade") {
                return 403;
            }
            if ($http_upgrade !~* "websocket") {
                return 403;
            }
            proxy_set_header Host $host;
            proxy_pass https://localhost:8443;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_read_timeout 86400;
        }
```

4. Перезапустите nginx

5. Для установки Websocketсоединения из браузера используйте URL

```
wss://login:password@wcs.yourdomain.com:443/wss
```

Полный файл настройки nginx

```
http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    auth_basic "Restricted Area";
    auth_basic_user_file /etc/nginx/.htpasswd;

    include /etc/nginx/conf.d/*.conf;

    server {
                listen 443 ssl;
                ssl_certificate /etc/pki/tls/yourdomain/yourdomain.crt;
                ssl_certificate_key /etc/pki/tls/yourdomain/yourdomain.key;
                server_name wcs.yourdomain.com;
                server_tokens off;
                client_max_body_size 500m;
                proxy_read_timeout 10m;

        include /etc/nginx/default.d/*.conf;

        location / {
        }

        location /wss {
            if ($http_connection !~* "upgrade") {
                return 403;
            }
            if ($http_upgrade !~* "websocket") {
                return 403;
            }
            proxy_set_header Host $host;
            proxy_pass https://localhost:8443;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_read_timeout 86400;
        }

        error_page 404 /404.html;
            location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }

    }
}
```

Настройка обратного прокси с передачей токена авторизации в cookie

Передача параметров авторизации в URL объявлена устаревшей. При этом браузеры до сих пор не предлагают способов передать дополнительные заголовки при установке Websocket соединения. Поэтому альтернативой может быть передача токена авторизации в cookie с проверкой токена на стороне nginx.

Настройка клиента

Клиент должен установить cookie с токеном авторизации перед установкой websocket соединения:

```
    setCookie("AUTH", token, {secure: true, 'max-age': 3600});
    Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function (session) {
        ...
    });
```

Код для установки или изменения cookie в браузере

```
function setCookie(name, value, options = {}) {
  options = {
    path: '/',
    ...options
  };

  if (options.expires instanceof Date) {
    options.expires = options.expires.toUTCString();
  }

  let updatedCookie = encodeURIComponent(name) + "=" + encodeURIComponent(value);

  for (let optionKey in options) {
    updatedCookie += "; " + optionKey;
    let optionValue = options[optionKey];
    if (optionValue !== true) {
      updatedCookie += "=" + optionValue;
    }
  }

  document.cookie = updatedCookie;
}
```

При разрыве Websocket сессии cookie можно очистить

```
     Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function (session) {
        ...
    }).on(SESSION_STATUS.DISCONNECTED, function () {
        setCookie("AUTH", "", {'max-age': -1});
        ...
     }).on(SESSION_STATUS.FAILED, function () {
        setCookie("AUTH", "", {'max-age': -1});
        ...
     });
```

Настройка nginx

1. Создайте каталог для токенов авторизации

```
mkdir -p /var/lib/nginx/tokens
```

и назначьте пользователя, под которым запускается nginx, владельцем

```
chown -R nginx /var/lib/nginx/token
```

2. Добавьте в файл настройки nginx проверку токена

```
        location /wss {
            if (!-f /var/lib/nginx/tokens/$cookie_AUTH) {
               return 403;
            }
            proxy_set_header Host $host;
            proxy_pass https://localhost:8443;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_read_timeout 86400;
        }
```

3. Перезапустите nginx

Использование

Для подключения клиента необходимо создать файл токена

```
touch /var/lib/nginx/tokens/ABCDEF1234565789
chown nginx /var/lib/nginx/tokens/ABCDEF1234565789
```

и передать значение токена на клиент для установки cookie. Возможные способы передачи находятся за пределами данного описания.

Известные проблемы

В целях безопасности, необходимо контролировать заголовок `Origin` , и принимать cookie только с разрешенных доменов.

Настройка передачи реального IP адреса клиента на WCS

При вышеуказанной настройке прокси, все клиентские сессии, с точки зрения WCS, будут иметь IP адрес 127.0.0.1. Это затрудняет отладку в случае проблем с публикацией или проигрыванием потока, поскольку не дает идентифицировать реальный источник подключения, и не позволяет запустить сбор отладочных логов по IP-адресу клиента. Для того, чтобы обойти это ограничение, в сборке5.2.743добавлена настройка, позволяющая передать реальный адрес источника сессии при помощи HTTP-заголовка

```
ws.map_custom_headers=true
ws.ip_forward_header=X-Real-IP
```

По умолчанию, прокси-сервер должен передавать реальный адрес клиента в заголовке X-Real-IP.

Рассмотрим пример настройки nginx и WCS для передачи реального IP адреса клиента.

Настройка nginx

1. Добавьте к настройке Websocket прокси формирование заголовка X-Client-IP

```
        location /wss {
            proxy_set_header Host $host;
            proxy_set_header X-Client-IP $remote_addr:$remote_port;
            proxy_pass https://localhost:8443;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_read_timeout 86400;
        }
```

2. Перезапустите nginx

Настройка WCS

1. Добавьте в файл flashphoner.properties следующие параметры

```
ws.map_custom_headers=true
ws.ip_forward_header=X-Client-IP
```

2. Перезапустите WCS

После этого, sessionId на стороне WCSбудут содержать реальный IP адрес клиента. Кроме того, на бэкенд сервер в REST hook /connect также пойдет заголовок, добавленный прокси сервером:

```
{
  "nodeId" : "nziJYH0eu3D08Iu25sXbmwaCgSUuQyGL@192.168.130.39",
  "appKey" : "defaultApp",
  "sessionId" : "/192.168.23.83:65520/127.0.0.1:8443-8ef8fa79-a726-44d3-a20a-fe27b94bc51f",
  "useWsTunnel" : false,
  "useWsTunnelPacketization2" : false,
  "msePacketizationVersion" : 2,
  "useBase64BinaryEncoding" : false,
  "mediaProviders" : [ "WebRTC", "MSE", "WSPlayer" ],
  "clientVersion" : "0.5.28",
  "clientOSVersion" : "5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.
0.4147.135 Safari/537.36",
  "clientBrowserVersion" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/84.0.4147.135 Safari/537.36",
  "keepAlive" : false,
  "origin" : "https://wcs.yourdomain.com",
  "X-Client-IP" : "192.168.23.83:65520"
}
```