Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2015

S. Loreto, Ed.
J. Mattsson
R. Skog
H. Spaak
Ericsson
G. Bourg
D. Druta
M. Hafeez
AT&T

July 3, 2014

# Explicitly Authenticated Proxy in HTTP/2.0
# draft-loreto-httpbis-explicitly-auth-proxy-01

Abstract

This document proposes the definition of an Explicitly Authenticated Proxy as intermediary of normally unprotected "http" URI scheme requests and responses of HTTP2 traffic.

An Explicitly Authenticated Proxy is a message forwarding agent that is selected, with explicit user's consent, and configured by the user agent to receive exclusively "http" URI scheme requests and attempt to satisfy those requests on behalf of the user agent. A client is connected to an Explicitly Authenticated Proxy through an authenticated TLS secured connection.

This document describes a method for a user agent to automatically discover and authenticate, and for an user to provide consent for an Explicitly Authenticated Proxy. This enables proxied communication to be encrypted and authenticated, explicitly acknowledged by the user agent and visible to the server end point.

This Internet-Draft will expire on January 4, 2015.



Table of Contents

## 1. Introduction

HTTP/1.1 and earlier allowed for the use of proxies and gateways to satisfy requests through a chain of connections. This has made possible a Web ecosystem of various kinds of proxies and gateways: cache servers, security gateways, web accelerators, content filters, and many others. In some cases their presence is explicit (configured proxies), and in other they are completely transparent to the end user (interception proxies, and gateways such as reverse proxies).

The success and the presence of the proxies and gateways is also a problem for the evolution of the HTTP as their behaviour on protocol extensions, and especially on alternative wire formats of the protocol, is not predictable. This unpredictable behaviour can lead to difficulties to deploy new versions of the protocol before the intermediaries are themselves updated. As an example, see the difficulties in deploying the WebSocket Protocol [RFC6455] in clear. It can also lead to potentially problematic trust models where proxies are accessing traffic content without the user being aware. Relying on establishing an HTTPS tunnel has then become the popular way to bypass the intermediate proxies as it provides reliable deployment model for web protocols. The encrypted tunnel obfuscates the data from all intermediaries and provides integrity validation.

HTTPS tunnels, while speeding up the deployment, make it difficult for a forward proxy and other gateways to be used to enable caching, enhance anonymity for a user agent, or enhance security by scanning content for virus and malware. HTTPS tunnels also remove the possibility to enhance delivery performance based on the knowledge of the network status, and this become an important limitation especially with HTTP2 when multiple streams are multiplexed on top of the same TCP connection.

Several drafts analysing the role and the requirements for proxy have been submitted:

1. [I-D.nottingham-http-proxy-problem] discusses the use and configuration of proxies in HTTP, pointing out problems in the currently deployed Web infrastructure along the way

2. [I-D.vidya-httpbis-explicit-proxy-ps] describes the issues with HTTP proxies for TLS protected traffic and motivates the need for explicit proxying capability in HTTP. It also presents the goals that such a solution would need to satisfy and some example solution directions.

3. [I-D.rpeon-httpbis-exproxy] describes a method for connecting to a proxy via a secure channel, allowing, disallowing, and detecting any transforms that the proxy may perform, and allowing the proxy to connect via secure channel to another site on the user's behalf.

Use cases in form of stories for proxies are also listed in the wiki Proxy-User-Stories [1] and analysed in a matrix form in Trusted Proxy Use Case Analysis and Alternatives [2].

This draft explicitly narrows down the general discussion to the role of Proxy as intermediary of "http" scheme URIs of HTTP2 traffic.

## 1.1. Goals and non Goals

The primary goal is to define an intermediary to 'http' traffic, that is TLS connected to the browser, operates with the knowledge and explicit consent of the user.

Non goal is to define an intermediary for 'https' URI. However the intermediary's expected behaviour for this case is listed for completeness.

## 1.2. Explicitly Authenticated Proxy

An "Explicitly Authenticated", as defined in this document, is an HTTP Proxy (see section 2.3 [I-D.ietf-httpbis-p1-messaging]) that is certificate authenticated, user acknowledged and connected to over a TLS encrypted (and possibly integrity protected) connection. An Explicitly Authenticated Proxy is configured by the user agent to exclusively receive "http" URI scheme requests and attempt to satisfy those requests on behalf of the user agent.

The presence of a configured Explicitly Authenticated Proxy MUST NOT change the user agent behaviour for the "https" URI scheme requests.

To distinguish between an HTTP2 connection meant to transport "https" URIs resources and an HTTP2 connection meant to transport "http" URIs resource, this document defines the ALPN [I-D.ietf-tls-applayerprotoneg] identifier "h2c" to signal that HTTP2 transports "http" URI requests and resources over TLS.

This document describes a method for an user agent to automatically discover and then for an user to accept or reject (i.e. to provide consent for) an Explicitly Authenticated Proxy to be securely involved when a request to an "http" URI resource is made.

Section 3 defines a solution based on sending a proxy certificate in the TLS handshake.

Section 5 describes the role of the Explicitly Authenticated Proxy in helping the user to fetch "http" URIs resource when the user has provided consent to the Explicitly Authenticated Proxy to be involved.

## 2. Terminology

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 [RFC2119].

This document defines the following terms:

Explicit proxy: an intercepting proxy (see section 2.3 [I-D.ietf-httpbis-p1-messaging]) that communicates its presence to the user agent and destination server..

Explicitly Authenticated Proxy: an HTTP Proxy that is certificate authenticated, user acknowledged and connected to over a TLS encrypted (and possibly integrity protected) connection. An Explicitly Authenticated Proxy is configured by the user agent to exclusively receive "http" URI scheme requests and attempt to satisfy those requests on behalf of the user agent. The presence of a configured Explicitly Authenticated Proxy MUST NOT change the user agent behaviour for the "https" URI scheme requests.

## 3. Establishing proxy connection

An Explicitly Authenticated Proxy indicates its presence, identity and willingness to serve the user agent by intercepting TLS ClientHello message containing "h2c" value (a new ALPN protocol type assigned for this purpose) in the ALPN [I-D.ietf-tls-applayerprotoneg] negotiation extension field. It answers the TLS initiation with a TLS ServerHello message containing the Proxy certificate Appendix A .

### 3.1. TLS Handshake with Proxy certificate

When a (TLS and HTTP) user agent receives a Server Certificate message, it checks whether the certificate contains an Extended Key Usage extension and if so whether the "proxyAuthentication" key purpose id is included. If it is included, the user agent concludes that the certificate belongs to a proxy. The user agent then SHOULD ensure user consent.

If the user provides consent, the user agent continues the TLS handshake with the proxy.

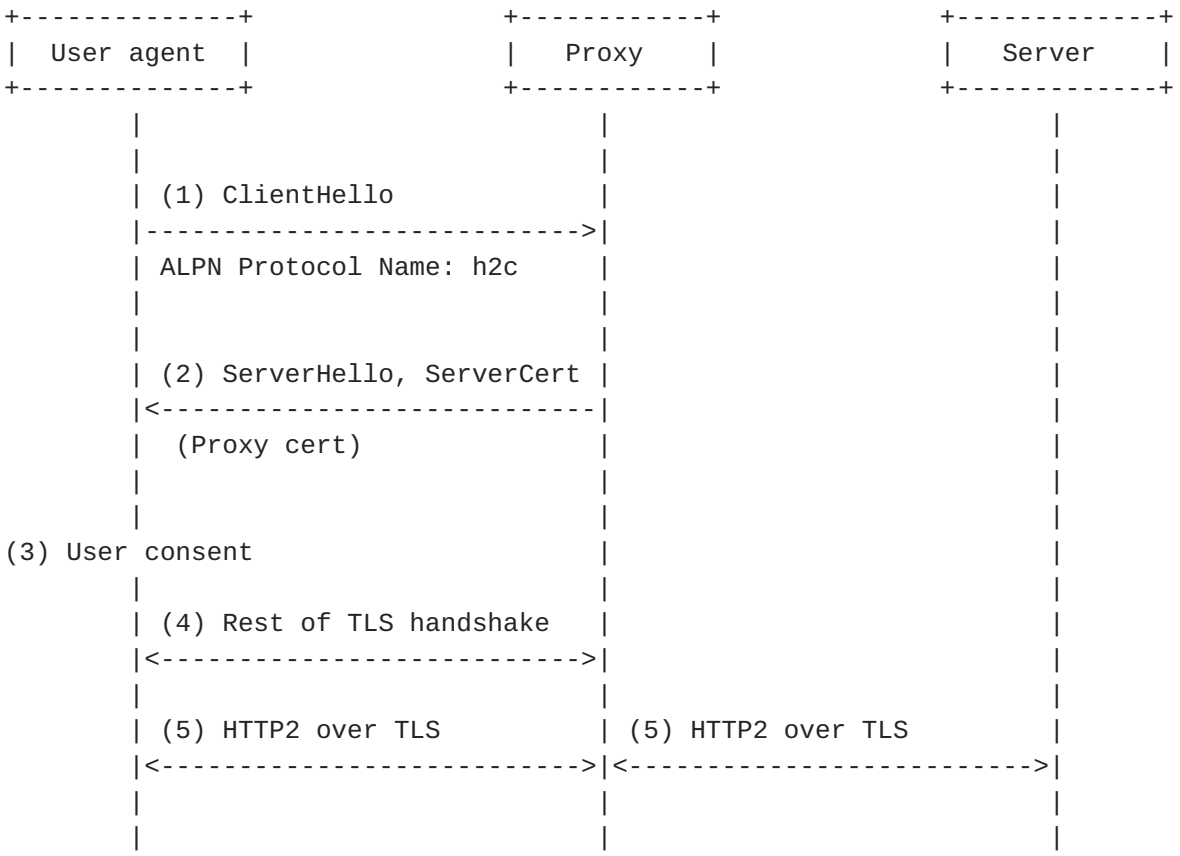


Figure 1: TLS Handshake with Proxy certificate

## 4. Connection to a mobile network

When a handset connects to a mobile network it is desirable to preserve the integrity of its exchange with the servers which host the services of this network entity. These use cases are described in [I-D.nottingham-http-proxy-problem] and in the [Proxy-User-Stories].

This section proposes a solution for such use cases. The proposal is inspired on the connection management specified in the section 9.1 of [I-D.ietf-httpbis-http2]. The connection with this proxy is used for all the servers' names listed in the "subjectAltName" field (http://tools.ietf.org/html/rfc5280#page-35) of the certificate of this proxy.

### 4.1. proxy discovery in a mobile network

At the network attachment, as usual, the network entity provides the handset with an IP address and with other pieces of information like DNS resolvers IP addresses. The network entity additionally provides the handset with the server name (e.g. pr.example.com) of the Explicitly Authenticated Proxy in charge of the domain names this network entity is authoritative on. These pieces of information are provided to the handset through a secure channel which preserves the integrity of the information.

## 5. Explicit Proxy behaviour

This section describes the role of the Explicitly Authenticated Proxy in helping the user to fetch http URI resources when the user has provided consent to the Explicitly Authenticated Proxy to be involved.

### 5.1. Explicitly Authenticated Forward Proxy towards HTTP2 origin server

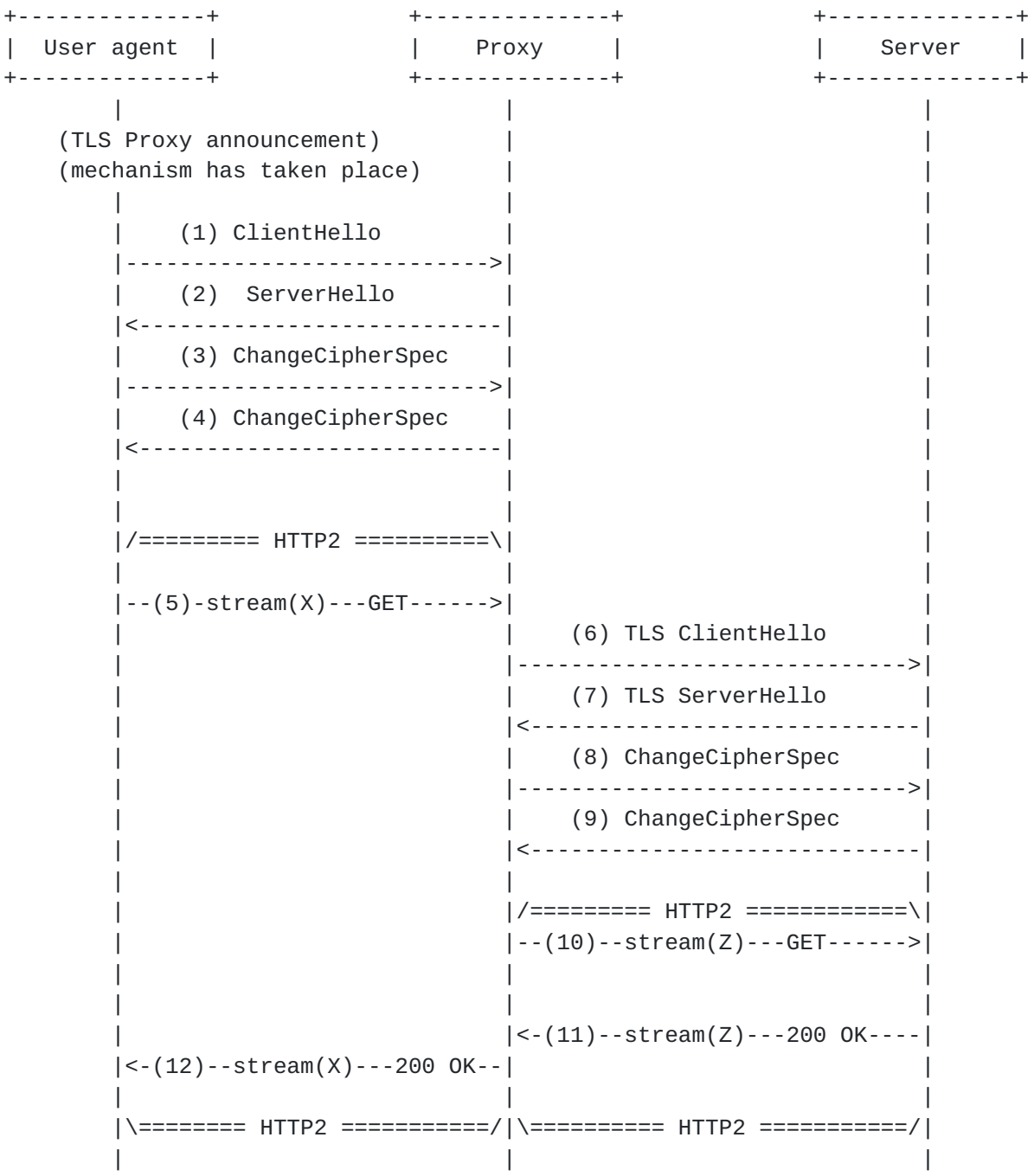


Figure 2: Requesting an HTTP resource

(0) The TLS Proxy Announcement (Section 3) mechanism has already taken place, so the user agent is now configured in the proxy mode.

(1)...(4) For each "http" URI resource towards a not yet contacted Server Origin, the user agent negotiates a new TLS session, using

the ALPN extension containing the "h2c" tag, to establish an HTTP2 connection.

(5) The user agent will then use the streams in the HTTP2 connection to request any resources hosted on that Origin Server.

(6)...(9) In the case the Proxy receives a request for a resource towards a not yet contacted Server Origin, the Explicitly Authenticated Proxy negotiates a new TLS session, using the ALPN extension containing the "h2c" ALPN identifier, to establish an HTTP2 connection.

(10) Once the Proxy has established the HTTP2 connection toward the origin, it picks one stream to forward the request

(11), (12) The Proxy forwards the answer it receives from the Origin Server to the user agent.

### 5.2. Explicitly Authenticated Forward Proxy towards HTTP/1.1 Origin Server

In the case the proxy has a privies knowledge about the fact that the "http" URI resources requested by the user agent will be only available over HTTP/1.1 or the proxy does not have a previous knowledge about it, the proxy will then attempt to contact the resource based on its knowledge.

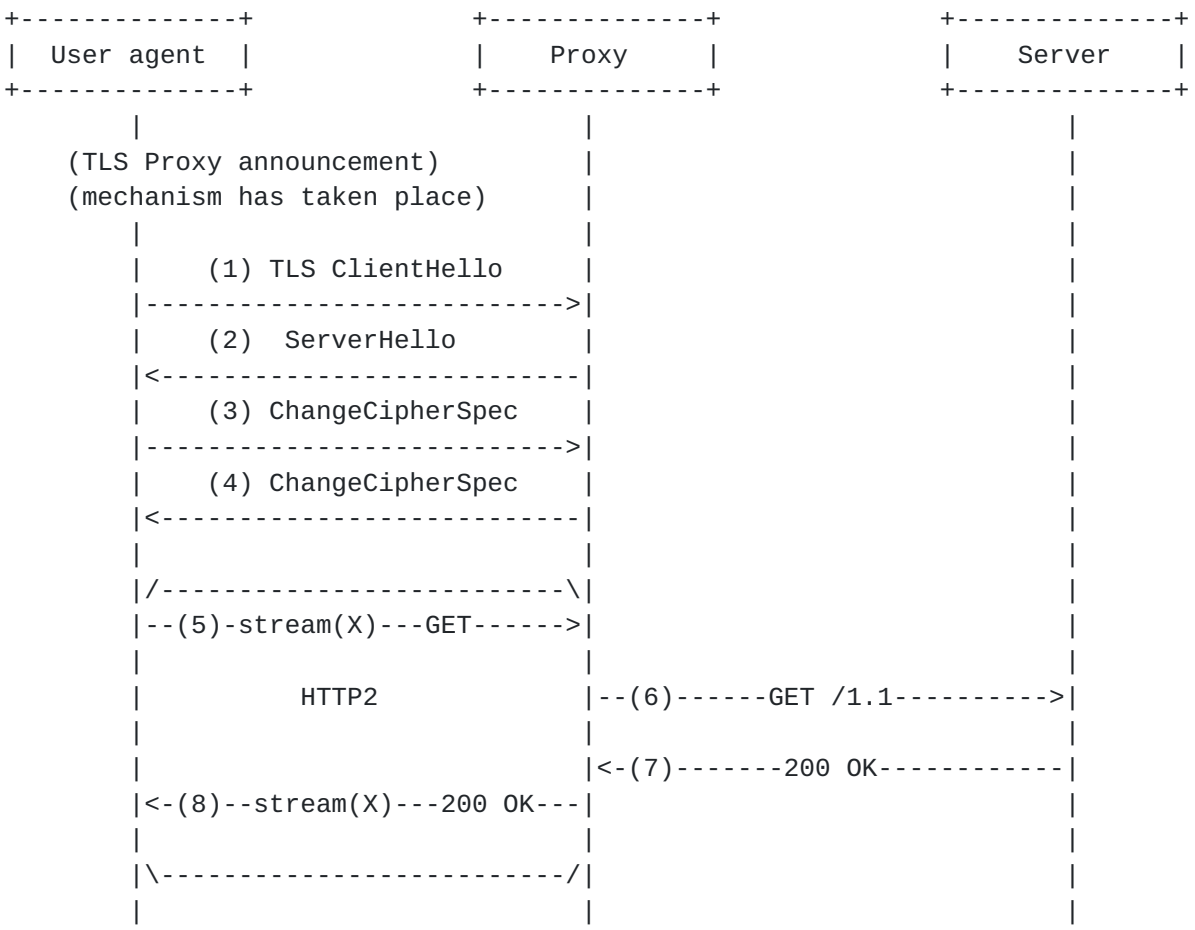


Figure 3: Origin server with only HTTP/1.1 support

## 5.3. Explicitly Authenticated Forward Proxy and https URIs

A user agent MUST NOT use "h2c" as ALPN extension field in request for https resources.

The Proxy that intercepts the TLS ClientHello analyses the ALPN extension field and if it does not contain the "h2c" value it does not do anything and lets the TLS handshake continue and the TLS session be established between the user agent and the Server (see Figure 4).


Figure 4: Explicitly Authenticated Proxy and https URI resources

## 6. User Consent

This document proposes an approach to making the presence of proxy explicit, explaining the functions it provides to users and letting them decide whether they accept that. A user can opt out and choose to bypass the proxy. This ensures that a proxy never acts as intermediary for HTTP2 traffic unless authorised by the user.

The user selection can be cached by the user agent. A consent SHOULD however be limited to the specific network access (such as APN or SSID) and may be limited to a single connection to that access or limited in time. How the consent information is stored is implementation specific, but as a network may have several proxies (for network resilience) it is RECOMMENDED that the consent is only tied to the Subject field of the proxy certificate so that the consent applies to all proxy certificates with the same name.

### 6.1. Expected behaviour if the user opts out/revokes consent

If the user does not give consent, or decides to opt out from the proxy for a specific connection, the user agent will negotiate HTTP2 connection using "h2" value in the ALPN extension field. The proxy will then treat the connection as an "https" connection and will forward the ClientHello message to the Server, establishing an end-to-end TLS connection between the user agent and the destination server.

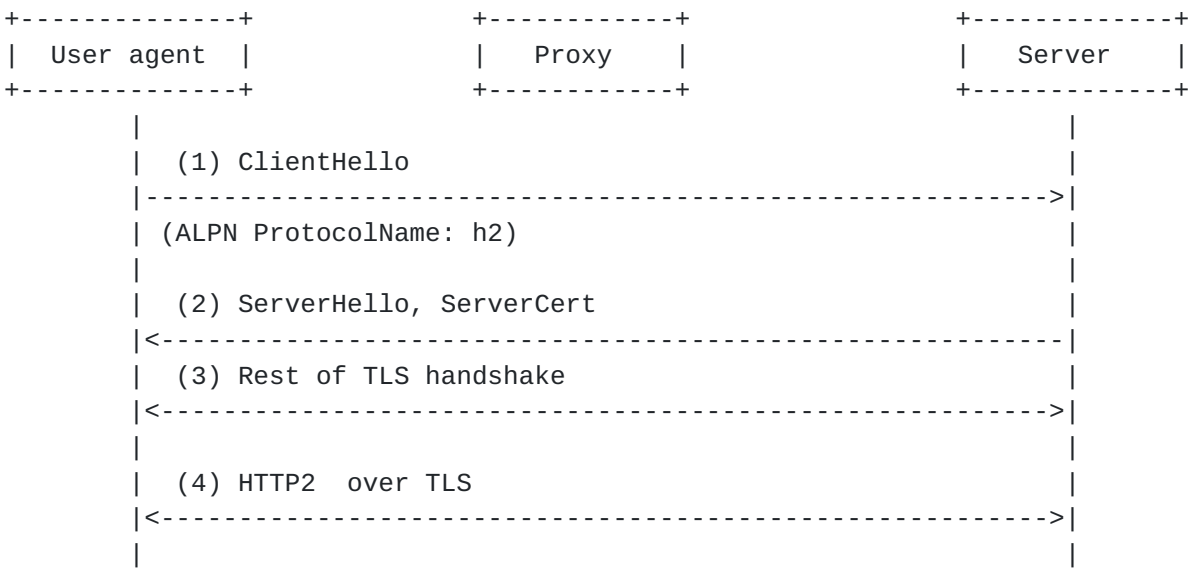


Figure 5: Opt Out

## 7. Signalling the presence of a Proxy in between

The presence of Explicitly Authenticated Proxy in between an user agent and the origin server must be signalled to the origin server using an already defined HTTP header.

The Explicitly Authenticated proxy MUST add, or update when already present, the Forwarded HTTP header field [I-D.ietf-appsawg-http-forwarded] "for" parameter.

## 8. Security Considerations

This document addresses Explicitly Authenticated proxies that act as intermediary for HTTP2 traffic and therefore the security and privacy implications of having those proxies in the path need to be considered. MITM [3], [I-D.nottingham-http-proxy-problem] and [I-D.vidya-httpbis-explicit-proxy-ps] discuss various security and privacy issues associated with the use of proxies.

It should however be noticed that the presence of the Explicitly Authenticated proxy as discussed in this document does not in any way affect "https" URI resources. Those resources are protected end-to-end between user agent and origin server as usual. Only for "http" URI resources the achievable security level of hop-by-hop protection may be different than end-to-end protection, because it is now also dependent on the security features/capabilities of the proxy as to what cipher suites it supports, which root CA certificates it trusts, how it checks certificate revocation status, etc. Users should also

be made aware that the proxy has visibility to the actual content they exchange with Web servers, including personal and sensitive information.

The TLS connection from the user agent to the Explicitly Authenticated proxy is always authenticated. In case the origin server only offers unauthenticated TLS (e.g. by using a self-signed certificate) the explicit Explicitly Authenticated proxy increases the security in the access network (e.g. an unencrypted hotspot) by ensuring that there is no unwanted MITMs in this part of the network.

To ensure the trustfulness of proxies, certification authorities validation procedure for issuing proxy certificates should be more rigorous than for issuing normal certificates and may also include technical details and processes relevant for the security assurance. The owner of the proxy could for example be obliged to apply security patches in a timely fashion.

When negotiating ciphersuite with the server, the Explicitly Authenticated proxy SHALL offer the ciphersuite negotiated between the user-agent and the proxy. Ciphersuites with a higher security level that the ciphersuite negotiated between the user-agent and proxy MAY be given a higher preference than the ciphersuite negotiated between the user-agent and proxy. Ciphersuites with a lower security level that the ciphersuite negotiated between the user-agent and proxy SHALL NOT be given a higher preference than the ciphersuite negotiated between the user-agent and proxy. While AES-256 is no weaker (an most probably much stronger) than AES-128, the relative security between different algorithm e.g SHA-256 vs Keccak-256 is not that clear. With security level we mean the complexity of the best known attack on that ciphersuite. The Explicitly Authenticated proxy SHOULD therefore be up to date with the best current practices regarding TLS.

This document proposes an approach to making the presence of proxy explicit to users and letting them decide whether they accept that. A user can opt out and choose to bypass the proxy. This ensures that a proxy never acts as intermediary for HTTP2 traffic unless authorised by the user.

When the user has given consent to the presence of the proxy, the user agent switches to a Proxy mode in which it does not check the hostname of the origin server against the server's identity as presented in the Server Certificate message. However if any of the following checks fails the user agent should immediately exit this Proxy mode:

1. the server's certificate is issued by a trusted CA and the certificate is valid;

2. the Extended Key Usage extension is present in the certificate and indicates the owner of this certificate is a proxy;

3. the server possesses the private key corresponding to the certificate.

## 9. Acknowledgments

The authors wish to thank Yi Cheng, Goran Eriksson, Stefan Hakansson, Nicolas Mailhot, Martin Nilsson, Emile Stephan (Connection with prior knowledge) and Salman Taj for their ideas, technical suggestions and comments.

## 10. References

### 10.1. Normative References

[I-D.ietf-appsawg-http-forwarded]
Petersson, A. and M. Nilsson, "Forwarded HTTP Extension", draft-ietf-appsawg-http-forwarded-10 (work in progress), October 2012.

[I-D.ietf-httpbis-http2]
Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", draft-ietf-httpbis-http2-13 (work in progress), June 2014.

[I-D.ietf-httpbis-p1-messaging]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", draft-ietf-httpbis-p1-messaging-26 (work in progress), February 2014.

[I-D.ietf-tls-applayerprotoneg]
Friedl, S., Popov, A., Langley, A., and S. Emile, "Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension", draft-ietf-tls-applayerprotoneg-05 (work in progress), March 2014.

[I-D.nottingham-http-proxy-problem]
Nottingham, M., "Problems with Proxies in HTTP", draft-nottingham-http-proxy-problem-00 (work in progress), October 2013.

[I-D.rpeon-httpbis-exproxy]
Peon, R., "Explicit Proxies for HTTP/2.0", draft-rpeon-httpbis-exproxy-00 (work in progress), June 2012.

[I-D.vidya-httpbis-explicit-proxy-ps]
Narayanan, V., "Explicit Proxying in HTTP - Problem Statement And Goals", draft-vidya-httpbis-explicit-proxy-ps-00 (work in progress), October 2013.

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

[RFC3709] Santesson, S., Housley, R., and T. Freeman, "Internet X.509 Public Key Infrastructure: Logotypes in X.509 Certificates", RFC 3709, February 2004.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

### 10.2. Informative References

[RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, December 2011.

### 10.3. URIs

[1] https://github.com/http2/http2-spec/wiki/Proxy-User-Stories

[2] https://github.com/bizzbyster/TrustedProxy/wiki/Trusted-Proxy-Use-Case-Analysis-and-Alternatives

[3] Jarmoc, J., SSL/TLS Interception Proxies and Transitive Trust, 2012 https://www.grc.com/miscfiles/HTTPS_Interception_Proxies.pdf

## Appendix A. Proxy certificate

To help HTTP user agents identify and distinguish Explicitly Authenticated proxies from other servers (e.g. web servers), Explicitly Authenticated proxies should have a certification authority issued public key certificate.

More specifically, the certification authority SHOULD use the Extended Key Usage extension as specified in [RFC5280] to indicate a key purpose "proxyAuthentication" (a new object identifier needs to be assigned by IANA for this key purpose). The certification authority also marks this Extended Key Usage extension as critical.

As the user needs to have high trust in the Proxy, it is desirable that the validation procedure for issuing proxy certificates be more rigorous than for issuing ordinary SSL certificates.

A proxy certificate MUST contain the SubjectAltName extension as defined in [RFC5280]. A name identifying the legal entity that is operating the proxy should be given in this extension.

To help end users understand the reason why the proxy is offered (in other words, the benefits of having the proxy in the path), a new X.509 certificate extension ProxyFunctions is introduced to list the functions the proxy is performing. More specifically, the ProxyFunction extension consists of a sequence of ProxyFunctionId which are object identifiers. The user agent should check the presence of this extension in the proxy certificate and present the proxy functions in a human readable format.

The user agent will provide the user with an opportunity to graphically view the results of a successful proxy certificate-based identification process leveraging on the usage of logotypes in public key certificates and attribute certificates as specified in [RFC3709].

## Authors' Addresses

Salvatore Loreto (editor)
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: salvatore.loreto@ericsson.com

John Mattsson
Ericsson
Kista
Sweden

Email: john.mattsson@ericsson.com

Robert Skog
Ericsson
Kista
Sweden

Email: robert.skog@ericsson.com

Hans Spaak
Ericsson
Kista
Sweden

Email: hans.spaak@ericsson.com

Gus Bourg
AT&T

Email: gb3635@att.com

Dan Druta
AT&T

Email: dd5826@att.com

Mohammad Hafeez
AT&T

Email: mh2897@att.com