Workgroup: HTTPBIS
Internet-Draft:
draft-schinazi-httpbis-unprompted-auth-00
Published: 13 October 2022
Intended Status: Standards Track
Expires: 16 April 2023
Authors: D. Schinazi
Google LLC
D. Oliver
Guardian Project
J. Hoyland
Cloudflare Inc.

HTTP Unprompted Authentication

Abstract

Existing HTTP authentication mechanisms are probeable in the sense that it is possible for an unauthenticated client to probe whether an origin serves resources that require authentication. It is possible for an origin to hide the fact that it requires authentication by not generating Unauthorized status codes, however that only works with non-cryptographic authentication schemes: cryptographic schemes (such as signatures or message authentication codes) require a fresh nonce to be signed, and there is no existing way for the origin to share such a nonce without exposing the fact that it serves resources that require authentication. This document proposes a new non-probeable cryptographic authentication scheme.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://DavidSchinazi.github.io/draft-schinazi-httpbis-transport-auth/draft-schinazi-httpbis-unprompted-auth.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-schinazi-httpbis-unprompted-auth/.

Discussion of this document takes place on the HTTP Working Group mailing list (mailto:ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/.

Source for this draft and an issue tracker can be found at https://github.com/DavidSchinazi/draft-schinazi-httpbis-transport-auth.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 16 April 2023.

Copyright Notice

Table of Contents

1. Introduction

Existing HTTP authentication mechanisms (see Section 11 of [HTTP]) are probeable in the sense that it is possible for an unauthenticated client to probe whether an origin serves resources that require authentication. It is possible for an origin to hide the fact that it requires authentication by not generating Unauthorized status codes, however that only works with non-cryptographic authentication schemes: cryptographic schemes (such as signatures or message authentication codes) require a fresh nonce to be signed, and there is no existing way for the origin to share such a nonce without exposing the fact that it serves resources that require authentication. This document proposes a new non-probeable cryptographic authentication scheme.

There are scenarios where servers may want to expose the fact that authentication is required for access to specific resources. This is left for future work.

1.1. Conventions and Definitions

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the following terminology from Section 3 of [STRUCTURED-FIELDS] to specify syntax and parsing: Integer, Token and Byte Sequence.

2. Computing the Authentication Proof

This document only defines Unprompted Authentication for uses of HTTP with TLS [TLS]. This includes any use of HTTP over TLS as typically used for HTTP/2 [HTTP/2], or HTTP/3 [HTTP/3] where the transport protocol uses TLS as its authentication and key exchange mechanism [QUIC-TLS].

The user agent leverages a TLS keying material exporter [KEY-EXPORT] to generate a nonce which can be signed using the user's key. The keying material exporter uses a label that starts with the characters "EXPORTER-HTTP-Unprompted-Authentication-" (see Section 4 for the labels and contexts used by each scheme). The TLS keying material exporter is used to generate a 32-byte key which is then used as a nonce.

3. Header Field Definition

The "Unprompted-Authentication" header field allows a user agent to authenticate with an origin server. The authentication is scoped to the HTTP request associated with this header field. The value of the Unprompted-Authentication header field is a token which represents the Unpromted Authentication Scheme; see Section 4. This header field supports parameters.

3.1. The u Parameter

The **OPTIONAL** "u" (user ID) parameter is a byte sequence that specifies the user ID that the user agent wishes to authenticate.

3.2. The p Parameter

The **OPTIONAL** "p" (proof) parameter is a byte sequence that specifies the proof that the user agent provides to attest to possessing the credential that matches its user ID.

3.3. The s Parameter

The **OPTIONAL** "s" (signature) parameter is an integer that specifies the signature algorithm used to compute the proof transmitted in the "p" directive. Its value is an integer between 0 and 255 inclusive from the IANA "TLS SignatureAlgorithm" registry maintained at <https://www.iana.org/assignments/tls-parameters#tls-parameters-16>.

3.4. The h Parameter

The **OPTIONAL** "h" (hash) parameter is an integer that specifies the hash algorithm used to compute the proof transmitted in the "p" directive. Its value is an integer between 0 and 255 inclusive from the IANA "TLS HashAlgorithm" registry maintained at <https://www.iana.org/assignments/tls-parameters#tls-parameters-18>.

4. Unprompted Authentication Schemes

The Unprompted Authentication Framework allows defining Unprompted Authentication Schemes, which specify how to authenticate user IDs. This documents defined the "Signature" and "HMAC" schemes.

4.1. Signature

The "Signature" Unprompted Authentication Scheme uses asymmetric cyptography. User agents possess a user ID and a public/private key pair, and origin servers maintain a mapping of authorized user IDs to their associated public keys. When using this scheme, the "u", "p", and "s" parameters are **REQUIRED**. The TLS keying material export label for this scheme is "EXPORTER-HTTP-Unprompted-Authentication-

Signature" and the associated context is empty. The nonce is then signed using the selected asymmetric signature algorithm and transmitted as the proof directive.

For example, the user ID "john.doe" authenticating using Ed25519 [ED25519] could produce the following header field (lines are folded to fit):

```
Unprompted-Authentication: Signature u=:am9obi5kb2U=:;s=7;
p=:SW5zZXJ0IHNpZ25hdHVyZSBvZiBub25jZSBoZXJlIHdo
aWNoIHRha2VzIDUxMiBiaXRzIGZvciBFZDI1NTE5IQ==:
```

4.2. HMAC

The "HMAC" Unprompted Authentication Scheme uses symmetric cyptography. User agents possess a user ID and a secret key, and origin servers maintain a mapping of authorized user IDs to their associated secret key. When using this scheme, the "u", "p", and "h" parameters are **REQUIRED**. The TLS keying material export label for this scheme is "EXPORTER-HTTP-Unprompted-Authentication-HMAC" and the associated context is empty. The nonce is then HMACed using the selected HMAC algorithm and transmitted as the proof directive.

For example, the user ID "john.doe" authenticating using HMAC-SHA-512 [SHA] could produce the following header field (lines are folded to fit):

```
Unprompted-Authentication: HMAC u="am9obi5kb2U=";h=6;
p="SW5zZXJ0IEhNQUMgb2Ygbm9uY2UgaGVyZSB3aGljaCB0YWtl
cyA1MTIgYml0cyBmb3IgU0hBLTUxMiEhISEhIQ=="
```

5. Intermediary Considerations

Since Unprompted Authentication leverages TLS keying material exporters, it cannot be transparently forwarded by HTTP intermediaries. HTTP intermediaries that support this specification will validate the authentication received from the client themselves, then inform the upstream HTTP server of the presence of valid authentication using some other mechanism.

6. Security Considerations

Unprompted Authentication allows a user-agent to authenticate to an origin server while guaranteeing freshness and without the need for the server to transmit a nonce to the user agent. This allows the server to accept authenticated clients without revealing that it supports or expects authentication for some resources. It also allows authentication without the user agent leaking the presence of authentication to observers due to clear-text TLS Client Hello extensions.

The authentication proofs described in this document are not bound to individual HTTP requests; if the same user sends an authentication proof on multiple requests they will all be identical. This allows for better compression when sending over the wire, but implies that client implementations that multiplex different security contexts over a single HTTP connection need to ensure that those contexts cannot read each other's header fields. Otherwise, one context would be able to replay the unprompted authentication header field of another. This constraint is met by modern Web browsers. If an attacker were to compromise the browser such that it could access another context's memory, the attacker might also be able to access the corresponding key, so binding authentication to requests would not provide much benefit in practice.

7. IANA Considerations

7.1. Unprompted-Authentication Header Field

This document will request IANA to register the following entry in the "HTTP Field Name" registry maintained at <https://www.iana.org/assignments/http-fields>:

Field Name: Unprompted-Authentication
Template: None
Status: provisional (permanent if this document is approved)
Reference: This document
Comments: None

7.2. Unprompted Authentication Schemes Registry

This document, if approved, requests IANA to create a new "HTTP Unprompted Authentication Schemes" Registry. This new registry contains strings and is covered by the First Come First Served policy from Section 4.4 of [IANA-POLICY]. Each entry contains an optional "Reference" field.

It initially contains the following entries:

*Signature

*HMAC

The reference for both is this document.

7.3. TLS Keying Material Exporter Labels

This document, if approved, requests IANA to register the following entries in the "TLS Exporter Labels" registry maintained at <https://www.iana.org/assignments/tls-parameters#exporter-labels>:

*EXPORTER-HTTP-Unprompted-Authentication-Signature

*EXPORTER-HTTP-Unprompted-Authentication-HMAC

Both of these entries are listed with the following qualifiers:

DTLS-OK: N
Recommended: Y
Reference: This document

8. References

8.1. Normative References

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[IANA-POLICY] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[KEY-EXPORT] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <https://www.rfc-editor.org/rfc/rfc5705>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[STRUCTURED-FIELDS] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/rfc/rfc8941>.

[TLS] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

8.2. Informative References

[ED25519] Josefsson, S. and J. Schaad, "Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure", RFC 8410, DOI 10.17487/RFC8410, August 2018, <https://www.rfc-editor.org/rfc/rfc8410>.

[HTTP/2] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/rfc/rfc9113>.

[HTTP/3] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/rfc/rfc9114>.

[QUIC-TLS] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

[SHA] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/rfc/rfc6234>.

Acknowledgments

The authors would like to thank many members of the IETF community, as this document is the fruit of many hallway conversations.

Authors' Addresses

David Schinazi
Google LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America

Email: dschinazi.ietf@gmail.com

David M. Oliver
Guardian Project

Email: david@guardianproject.info
URI: https://guardianproject.info

Jonathan Hoyland
Cloudflare Inc.

Email: jonathan.hoyland@gmail.com