Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2010

J. Hodges
PayPal
B. Leiba
Huawei Technologies

March 11, 2010

Security Requirements for HTTP
draft-ietf-httpbis-security-properties-05

Abstract

Recent IESG practice dictates that IETF protocols must specify mandatory-to-implement (MTI) security mechanisms, so that all conformant implementations share a common baseline. This document examines all widely deployed HTTP security technologies, and analyzes the trade-offs of each.

Status of this Memo

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

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

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 12, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 3
2. Existing HTTP Security Mechanisms 3
  2.1. Forms And Cookies 4
  2.2. HTTP Access Authentication 5
    2.2.1. Basic Authentication 6
    2.2.2. Digest Authentication 6
    2.2.3. Authentication Using Certificates in TLS 7
    2.2.4. Other Access Authentication Schemes 7
  2.3. Centrally-Issued Tickets 8
  2.4. Web Services 8
  2.5. Transport Layer Security 9
3. Revisions To HTTP 9
4. IANA Considerations 9
5. Security Considerations 9
6. References 9
  6.1. Normative References 9
  6.2. Informative References 11
Appendix A. Acknowledgements 11
Appendix B. Document History 11
  B.1. Changes between draft-sayre-http-security-variance-00
       and draft-ietf-httpbis-security-properties-00 11
  B.2. Changes between -00 and -01 11
  B.3. Changes between -01 and -02 12
  B.4. Changes between -02 and -03 12
  B.5. Changes between -03 and -04 12
  B.6. Changes between -04 and -05 13
Authors' Addresses 13

1. Introduction

Recent IESG practice dictates that IETF protocols be required to specify mandatory-to-implement (MTI) security mechanisms. "The IETF Standards Process" [RFC2026] does not require that protocols specify mandatory security mechanisms. "Strong Security Requirements for IETF Standard Protocols" [RFC3365] requires that all IETF protocols provide a mechanism for implementers to provide strong security. RFC 3365 does not define the term "strong security".

"Security Mechanisms for the Internet" [RFC3631] is not an IETF procedural RFC, but it is perhaps most relevant. Section 2.2 states:

> We have evolved in the IETF the notion of "mandatory to implement" mechanisms. This philosophy evolves from our primary desire to ensure interoperability between different implementations of a protocol. If a protocol offers many options for how to perform a particular task, but fails to provide for at least one that all must implement, it may be possible that multiple, non-interoperable implementations may result. This is the consequence of the selection of non-overlapping mechanisms being deployed in the different implementations.

This document examines the effects of applying security constraints to Web applications, documents the properties that result from each method, and will make Best Current Practice recommendations for HTTP security in a later document version. At the moment, it is mostly a laundry list of security technologies and tradeoffs.

[[OVERALL ISSUE: It isn't entirely clear to the present editors what the purpose of this document is. On one hand it could be a compendium of peer-entity authentication mechanisms (as it is presently) and make MTI recommendations thereof, or it could be a place for various security considerations (either coalesced here from the other httpbis specs, or reserved for the more gnarly cross-spec composite ones), or both. This needs to be clarified.]]

2. Existing HTTP Security Mechanisms

For HTTP, the IETF generally defines "security mechanisms" as some combination of access authentication and/or a secure transport.

[[There is a suggestion that this section be split into "browser-like" and "automation-like" subsections. See:

http://lists.w3.org/Archives/Public/ietf-http-wg/2008JanMar/0180.html

http://lists.w3.org/Archives/Public/ietf-http-wg/2008JanMar/0183.html

]]

[[NTLM (shudder) was brought up in the WG a few times in the discussion of the -00 draft. Should we add a section on it? See..

http://lists.w3.org/Archives/Public/ietf-http-wg/2008JanMar/0132.html

http://lists.w3.org/Archives/Public/ietf-http-wg/2008JanMar/0135.html

]]

2.1. Forms And Cookies

[[JH: I am not convinced that this subsection properly belongs in this overall section in that "HTTP+HTML Form based authentication" <http://en.wikipedia.org/wiki/HTTP%2BHTML_Form_based_authentication> is not properly a part of HTTP itself. Rather, it is a piece of applications layered on top of HTTP. Use of cookies for state management (e.g. session maintanence) can be considered such, however (although there is no overall specification for HTTP user agents stipulating that they must implement cookies (nominally [RFC2109])). Perhaps this section should be should be retitled "HTTP Authentication".

Note: The httpstate WG was recently chartered to develop a successor to [RFC2109]. See..

http://www.ietf.org/dyn/wg/charter/httpstate-charter.html

]]

Almost all HTTP authentication that involves a human using a web browser is accomplished through HTML forms, with session identifiers stored in cookies. For cookies, most implementations rely on the "Netscape specification", which is described loosely in section 10 of "HTTP State Management Mechanism" [RFC2109]. The protocol in RFC 2109 is relatively widely implemented, but most clients don't advertise support for it. RFC 2109 was later updated [RFC2965], but the newer version is not widely implemented.

Forms and cookies have many properties that make them an excellent solution for some implementers. However, many of those properties introduce serious security trade-offs.

HTML forms provide a large degree of control over presentation, which is an imperative for many websites. However, this increases user reliance on the appearance of the interface. Many users do not understand the construction of URIs [RFC3986], or their presentation in common clients [PhishingHOWTO]. As a result, forms are extremely vulnerable to spoofing.

HTML forms provide acceptable internationalization if used carefully, at the cost of being transmitted as normal HTTP content in all cases (credentials are not differentiated in the protocol).

Many Web browsers have an auto-complete feature that stores a user's information and pre-populates fields in forms. This is considered to be a convenience mechanism, and convenience mechanisms often have negative security properties. The security concerns with auto-completion are particularly poignant for web browsers that reside on computers with multiple users. HTML forms provide a facility for sites to indicate that a field, such as a password, should never be pre-populated. However, it is clear that some form creators do not use this facility when they should.

The cookies that result from a successful form submission make it unnecessary to validate credentials with each HTTP request; this makes cookies an excellent property for scalability. Cookies are susceptible to a large variety of XSS (cross-site scripting) attacks, and measures to prevent such attacks will never be as stringent as necessary for authentication credentials because cookies are used for many purposes. Cookies are also susceptible to a wide variety of attacks from malicious intermediaries and observers. The possible attacks depend on the contents of the cookie data. There is no standard format for most of the data.

HTML forms and cookies provide flexible ways of ending a session from the client.

HTML forms require an HTML rendering engine for which many protocols have no use.

2.2. HTTP Access Authentication

HTTP 1.1 provides a simple authentication framework, "HTTP Authentication: Basic and Digest Access Authentication" [RFC2617], which defines two optional mechanisms. Both of these mechanisms are extremely rarely used in comparison to forms and cookies, but some degree of support for one or both is available in many implementations. Neither scheme provides presentation control, logout capabilities, or interoperable internationalization.

2.2.1. Basic Authentication

Basic Authentication (normally called just "Basic") transmits usernames and passwords in the clear. It is very easy to implement, but not at all secure unless used over a secure transport.

Basic has very poor scalability properties because credentials must be revalidated with every request, and because secure transports negate many of HTTP's caching mechanisms. Some implementations use cookies in combination with Basic credentials, but there is no standard method of doing so.

Since Basic credentials are clear text, they are reusable by any party. This makes them compatible with any authentication database, at the cost of making the user vulnerable to mismanaged or malicious servers, even over a secure channel.

Basic is not interoperable when used with credentials that contain characters outside of the ISO 8859-1 repertoire.

2.2.2. Digest Authentication

In Digest Authentication, the client transmits the results of hashing user credentials with properties of the request and values from the server challenge. Digest is susceptible to man-in-the-middle attacks when not used over a secure transport.

Digest has some properties that are preferable to Basic and Cookies. Credentials are not immediately reusable by parties that observe or receive them, and session data can be transmitted alongside credentials with each request, allowing servers to validate credentials only when absolutely necessary. Authentication data session keys are distinct from other protocol traffic.

Digest includes many modes of operation, but only the simplest modes enjoy any degree of interoperability. For example, most implementations do not implement the mode that provides full message integrity. Perhaps one reason is that implementation experience has shown that in some cases, especially those involving large requests or responses such as streams, the message integrity mode is impractical because it requires servers to analyze the full request before determining whether the client knows the shared secret or whether message-body integrity has been violated and hence whether the request can be processed.

Digest is extremely susceptible to offline dictionary attacks, making it practical for attackers to perform a namespace walk consisting of a few million passwords for most users.

Many of the most widely-deployed HTTP/1.1 clients are not compliant when GET requests include a query string [Apache_Digest].

Digest either requires that authentication databases be expressly designed to accommodate it, or requires access to cleartext passwords. As a result, many authentication databases that chose to do the former are incompatible, including the most common method of storing passwords for use with Forms and Cookies.

Many Digest capabilities included to prevent replay attacks expose the server to Denial of Service attacks.

Digest is not interoperable when used with credentials that contain characters outside of the ISO 8859-1 repertoire.

2.2.3. Authentication Using Certificates in TLS

Running HTTP over TLS provides authentication of the HTTP server to the client. HTTP over TLS can also provides authentication of the client to the server using certificates. Although forms are a much more common way to authenticate users to HTTP servers, TLS client certificates are widely used in some environments. The public key infrastructure (PKI) used to validate certificates in TLS can be rooted in public trust anchors or can be based on local trust anchors.

2.2.4. Other Access Authentication Schemes

There are many niche schemes that make use of the HTTP Authentication framework, but very few are well documented. Some are bound to transport layer connections.

2.2.4.1. Negotiate (GSS-API) Authentication

Microsoft has designed an HTTP authentication mechanism that utilizes SPNEGO [RFC4178] GSSAPI [RFC4559]. In Microsoft's implementation, SPNEGO allows selection between Kerberos and NTLM (Microsoft NT Lan Manager protocols).

In Kerberos, clients and servers rely on a trusted third-party authentication service which maintains its own authentication database. Kerberos is typically used with shared secret key cryptography, but extensions for use of other authentication mechnanisms such as PKIX certificates and two-factor tokens are also common. Kerberos was designed to work under the assumption that packets traveling along the network can be read, modified, and inserted at will.

Unlike Digest, Negotiate authentication can take multiple round trips (client sending authentication data in Authorization, server sending authentication data in WWW-Authenticate) to complete.

Kerberos authentication is generally more secure than Digest. However the requirement for having a separate network authentication service might be a barrier to deployment.

2.2.4.2. OAuth

[[See..

http://www.ietf.org/id/draft-hammer-http-token-auth-01.txt

http://www.ietf.org/id/draft-hammer-oauth-10.txt

]]

2.3. Centrally-Issued Tickets

Many large Internet services rely on authentication schemes that center on clients consulting a single service for a time-limited ticket that is validated with undocumented heuristics. Centralized ticket issuing has the advantage that users may employ one set of credentials for many services, and clients don't send credentials to many servers. This approach is often no more than a sophisticated application of forms and cookies.

All of the schemes in wide use are proprietary and non-standard, and usually are undocumented. There are many standardization efforts in progress, as usual.

2.4. Web Services

Many security properties mentioned in this document have been recast in XML-based protocols, using HTTP as a substitute for TCP. Like the amalgam of HTTP technologies mentioned above, the XML-based protocols are defined by an ever-changing combination of standard and vendor-produced specifications, some of which may be obsoleted at any time [WS-Pagecount] without any documented change control procedures. These protocols usually don't have much in common with the Architecture of the World Wide Web. It's not clear why the term "Web" is used to group them, but they are obviously out of scope for HTTP-based application protocols.

[[This section could really use a good definition of "Web Services" to differentiate it from REST. See..

http://lists.w3.org/Archives/Public/ietf-http-wg/2008JanMar/0536.html

]]

2.5. Transport Layer Security

In addition to using TLS for client and/or server authentication, it is also very commonly used to protect the confidentiality and integrity of the HTTP session. For instance, both HTTP Basic authentication and Cookies are often protected against snooping by TLS.

It should be noted that, in that case, TLS does not protect against a breach of the credential store at the server or against a keylogger or phishing interface at the client. TLS does not change the fact that Basic Authentication passwords are reusable and does not address that weakness.

3. Revisions To HTTP

Is is possible that HTTP will be revised in the future. "HTTP/1.1" [RFC2616] and "Use and Interpretation of HTTP Version Numbers" [RFC2145] define conformance requirements in relation to version numbers. In HTTP 1.1, all authentication mechanisms are optional, and no single transport substrate is specified. Any HTTP revision that adds a mandatory security mechanism or transport substrate will have to increment the HTTP version number appropriately. All widely used schemes are non-standard and/or proprietary.

4. IANA Considerations

This document has no actions for IANA.

5. Security Considerations

This entire document is about security considerations.

6. References

6.1. Normative References

[Apache_Digest]
Apache Software Foundation, "Apache HTTP Server -

mod_auth_digest", <http://httpd.apache.org/docs/1.3/mod/mod_auth_digest.html>.

[PhishingHOWTO]
Gutmann, P., "Phishing Tips and Techniques", February 2008, <http://www.cs.auckland.ac.nz/~pgut001/pubs/phishing.pdf>.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2145] Mogul, J., Fielding, R., Gettys, J., and H. Nielsen, "Use and Interpretation of HTTP Version Numbers", RFC 2145, May 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2965] Kristol, D. and L. Montulli, "HTTP State Management Mechanism", RFC 2965, October 2000.

[RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, August 2002.

[RFC3631] Bellovin, S., Schiller, J., and C. Kaufman, "Security Mechanisms for the Internet", RFC 3631, December 2003.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and W. Ingersoll, "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, October 2005.

[RFC4559] Jaganathan, K., Zhu, L., and J. Brezak, "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006.

[WS-Pagecount]

Bray, T., "WS-Pagecount", September 2004, <http://www.tbray.org/ongoing/When/200x/2004/09/21/WS-Research>.

6.2. Informative References

[RFC2109] Kristol, D. and L. Montulli, "HTTP State Management Mechanism", RFC 2109, February 1997.

Appendix A. Acknowledgements

Much of the material in this document was written by Rob Sayre, who first promoted the topic. Many others on the HTTPbis Working Group have contributed to this document in the discussion.

Appendix B. Document History

[This entire section is to be removed when published as an RFC.]

B.1. Changes between draft-sayre-http-security-variance-00 and draft-ietf-httpbis-security-properties-00

Changed the authors to Paul Hoffman and Alexey Melnikov, with permission of Rob Sayre.

Made lots of minor editorial changes.

Removed what was section 2 (Requirements Notation), the reference to RFC 2119, and any use of 2119ish all-caps words.

In 3.2.1 and 3.2.2, changed "Latin-1 range" to "ISO 8859-1 repertoire" to match the definition of "TEXT" in RFC 2616.

Added minor text to the Security Considerations section.

Added URLs to the two non-RFC references.

B.2. Changes between -00 and -01

Fixed some editorial nits reported by Iain Calder.

Added the suggestions about splitting for browsers and automation, and about adding NTLM, to be beginning of 2.

In 2.1, added "that involves a human using a web browser" in the first sentence.

In 2.1, changed "session key" to "session identifier".

In 2.2.2, changed

Digest includes many modes of operation, but only the simplest modes enjoy any degree of interoperability. For example, most implementations do not implement the mode that provides full message integrity. Additionally, implementation experience has shown that the message integrity mode is impractical because it requires servers to analyze the full request before determining whether the client knows the shared secret.

to

Digest includes many modes of operation, but only the simplest modes enjoy any degree of interoperability. For example, most implementations do not implement the mode that provides full message integrity. Perhaps one reason is that implementation experience has shown that in some cases, especially those involving large requests or responses such as streams, the message integrity mode is impractical because it requires servers to analyze the full request before determining whether the client knows the shared secret or whether message-body integrity has been violated and hence whether the request can be processed.

In 2.4, asked for a definition of "Web Services".

In A, added the WG.

B.3. Changes between -01 and -02

In section 2.1, added more to the paragraph on auto-completion of HTML forms.

Added the section on TLS for authentication.

Filled in section 2.5.

B.4. Changes between -02 and -03

Changed IPR licensing from "full3978" to "pre5378Trust200902".

B.5. Changes between -03 and -04

Changed authors to be Jeff Hodges (JH) and Barry Leiba (BL) with permission of Paul Hoffman, Alexey Melnikov, and Mark Nottingham (httpbis chair).

Added "OVERALL ISSUE" to introduction.

Added links to email messages on mailing list(s) where various suggestions for this document were brought up. I.e. added various links to those comments herein delimited by "[[...]]" braces.

Noted JH's belief that "HTTP+HTML Form based authentication" aka "Forms And Cookies" doesn't properly belong in the section where it presently resides. Added link to httpstate WG.

Added references to OAuth. Section needs to be filled-in as yet.

Moved ref to RFC2109 to new "Informative References" section, and added a placeholder "IANA Considerations" section in order to satisfy IDnits checking.

B.6. Changes between -04 and -05

Fixed incorrect <date> year from 2009 to 2010. mea culpa.

Authors' Addresses

Jeff Hodges
PayPal

Email: Jeff.Hodges@PayPal.com

Barry Leiba
Huawei Technologies

Phone: +1 646 827 0648
Email: barryleiba@computer.org
URI: http://internetmessagingtechnology.org/