

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 19, 2007

R. Sayre
Mozilla Corporation
January 15, 2007

Security Requirements for HTTP
draft-sayre-http-security-variance-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of 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/lid-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 July 19, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Internet-Draft

Security Requirements for HTTP

January 2007

Abstract

Recent IESG practice dictates that IETF protocols must specify mandatory to implement 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.

Table of Contents

1.	Introduction	3
2.	Requirements Notation	4
3.	Existing HTTP Security Mechanisms	5
3.1.	Forms And Cookies	5
3.2.	HTTP Access Authentication	6
3.2.1.	Basic Authentication	6
3.2.2.	Digest Authentication	6
3.2.3.	Other Schemes	7
3.3.	Centrally-Issued Tickets	7
3.4.	Web Services	7
3.5.	Transport Layer Security	8
4.	Revisions To HTTP	9
5.	Security Considerations	10
6.	Normative References	11
	Author's Address	12
	Intellectual Property and Copyright Statements	13

1. Introduction

Note: this document is just a laundry list of security technologies and tradeoffs for the moment.

Recent IESG practice dictates that IETF protocols are required to specify mandatory to implement 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 implementors to provide strong security. The document 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.

2. Requirements Notation

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 [RFC 2119](#) [[RFC2119](#)].

3. Existing HTTP Security Mechanisms

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

3.1. Forms And Cookies

Almost all HTTP authentication is accomplished through HTML forms, with session keys stored in cookies. For cookies, most implementations rely on the Netscape specification. One update, HTTP State Management Mechanism [[RFC2109](#)] is relatively widely implemented, but most clients don't advertise support for it. HTTP State Management Mechanism was later updated [[RFC2965](#)], but the newer version is not widely implemented.

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

HTML forms provide a large degree of control over presentation, 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 [todo: citation]. 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).

HTML forms provide a facility for sites to indicate a password should never be pre-populated. [more on autocomplete]

The cookies that result from a successful form submission make it unnecessary to validate credentials with each HTTP request, 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, which many protocols

have no use for.

[3.2.](#) HTTP Access Authentication

HTTP 1.1 provides a simple authentication framework, and HTTP Authentication: Basic and Digest Access Authentication [[RFC2617](#)] 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.

[3.2.1.](#) Basic Authentication

Basic Authentication 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 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 Latin-1 range.

[3.2.2.](#) 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 along side 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. Additionally, implementation experience has shown that the mode is impractical, because it requires servers to analyze the full request before determining whether the client knows the shared secret.

Digest is extremely susceptible to offline dictionary attacks, making it practical for attackers to perform a namespace walk consisting of

a few million passwords [todo: cite].

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

Digest requires that authentication databases be expressly designed to accomodate it. As a result, many authentication databases 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 Latin-1 range.

[3.2.3.](#) Other 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.

[3.3.](#) 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, undocumented, and non-standard. There are many standardization efforts in progress, as usual.

[3.4.](#) Web Services

Many security properties mentioned above 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 [\[WS-Pagecount\]](#), with no documented change control procedures. These protocols usually don't have much in common the Architecture of the World Wide Web. It's not clear why term "Web" is used to group them, but they are obviously out of scope for HTTP-based application protocols.

[3.5.](#) Transport Layer Security

[todo]

4. Revisions To HTTP

It 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 MUST increment the HTTP version number appropriately. All widely used schemes are non-standard and/or proprietary.

[5.](#) Security Considerations

6. Normative References

[Apache_Digest]

ASF, "Apache HTTP Server - mod_auth_digest".

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

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

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

[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.

[WS-Pagecount]

Bray, T., "WS-Pagecount", September 2004.

Sayre

Expires July 19, 2007

[Page 11]

Internet-Draft

Security Requirements for HTTP

January 2007

Author's Address

Robert Sayre
Mozilla Corporation

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).