Network Working Group
Request for Comments: 4850
Updates: 3720
Category: Standards Track

D. Wysochanski
Network Appliance, Inc.

April 2007

Declarative Public Extension Key for Internet Small Computer Systems Interface (iSCSI) Node Architecture

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The Internet Small Computer Systems Interface (iSCSI) protocol, described in RFC 3720, allows for extension items to the protocol in the form of Private or Public Extension Keys. This document describes a Public Extension Key for the purpose of enhancing iSCSI supportability. The key accomplishes this objective by allowing iSCSI nodes to communicate architecture details during the iSCSI login sequence. The receiving node can then use this information for enhanced logging and support. This document updates RFC 3720 to allow iSCSI extension items to be defined by standards track RFCs and experimental RFCs in addition to informational RFCs.

1. Introduction

1.1. Overview

This document describes a declarative Public Extension Key, as defined by Section 12.22 of RFC 3720 [2], that may be used to communicate additional iSCSI node information to the peer node in a session. The information carried in the described key has been found to be valuable in real iSCSI customer environments as initiator and target vendors collaborate to resolve technical issues and better understand the interaction of iSCSI implementations.

The key has been modeled after the HTTP "Server" and "User-Agent" header fields as specified in Sections 14.38 and 14.43 of RFC 2616 [3], with the text-value(s) of the key roughly equivalent to Product Tokens in Section 3.8 of RFC 2616 [3]. Note, however, that the text-value(s) in the key's list-of-values MUST conform to the Text Format as specified in Section 5.1 of RFC 3720 [2].

The key is sent during operational parameter negotiation of an iSCSI session's login phase. The intended use of this key is to provide enhanced logging and support capabilities, and to enable collection of iSCSI implementation and usage information.

1.2. 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 RFC 2119 [1].

2. Definition

The definition of the key is as follows, conforming to Sections 11 and 12 of RFC 3720 [2], with example list-of-values conforming to Section 5.1 of RFC 3720 [2].

The key is defined with a use of "LO", making it a Leading Only key, and does not modify Sections 11 or 12 of RFC 3720 [2]. Thus, the key MUST only be sent on the leading connection, MUST NOT be changed after the leading connection login, and MUST only be sent after the security negotiation login stage has completed (during operational negotiation login stage). The key may be sent during normal or discovery sessions.

2.1. X#NodeArchitecture

Use: LO, Declarative
Senders: Initiator and Target
Scope: SW

X#NodeArchitecture=<list-of-values>

Examples:

```
X#NodeArchitecture=ExampleOS/v1234,ExampleInc_SW_Initiator/1.05a
X#NodeArchitecture=ExampleInc_HW_Initiator/4010,Firmware/2.0.0.5
X#NodeArchitecture=ExampleInc_SW_Initiator/2.1,CPU_Arch/i686
```

The initiator or target declares the details of its iSCSI node architecture to the remote endpoint. These details may include, but are not limited to, iSCSI vendor software, firmware, or hardware versions, the OS version, or hardware architecture.

The length of the key value (total length of the list-of-values) MUST NOT be greater than 255 bytes.

X#NodeArchitecture MUST NOT be redeclared.

3. Implementation

Functional behavior of the iSCSI node (this includes the iSCSI protocol logic -- the SCSI, iSCSI, and TCP/IP protocols) MUST NOT depend on the presence, absence, or content of the key. The key MUST NOT be used by iSCSI nodes for interoperability, or exclusion of other nodes. To ensure proper use, key values SHOULD be set by the node itself, and there SHOULD NOT be provisions for the key values to contain user-defined text.

Nodes implementing this key MUST choose one of the following implementation options:

- only transmit the key,
- only log the key values received from other nodes, or
- both transmit and log the key values.

Each node choosing to implement transmission of the key values MUST be prepared to handle the response of RFC 3720 [2] compliant nodes that do not understand the key (RFC 3720 [2] states that compliant nodes MUST respond with X#NodeArchitecture=NotUnderstood).

Nodes that implement transmission and/or logging of the key values may also implement administrative mechanisms that disable and/or

change the logging and key transmission detail (see Security Considerations). Thus, a valid behavior for this key may be that a node is completely silent (the node does not transmit any key value, and simply discards any key values it receives without issuing a NotUnderstood response).

4. Security Considerations

This extension key transmits specific implementation details about the node that sends it; such details may be considered sensitive in some environments. For example, if a certain software or firmware version is known to contain security weaknesses, announcing the presence of that version via this key may not be desirable. The countermeasures for this security concern are:

- sending less detailed information in the key values,
- not sending the extension key, or
- using IPsec to provide confidentiality for the iSCSI connection on which the key is sent (see RFC 3720 [2] and RFC 3723 [4]).

To support the first and second countermeasures, all implementations of this extension key MUST provide an administrative mechanism to disable sending the key. In addition, all implementations SHOULD provide an administrative mechanism to configure a verbosity level of the key value, thereby controlling the amount of information sent. For example, a lower verbosity might enable transmission of node architecture component names only, but no version numbers.

The choice of which countermeasure is most appropriate depends on the environment. However, sending less detailed information in the key values may be an acceptable countermeasure in many environments, since it provides a compromise between sending too much information and the other more complete countermeasures of not sending the key at all or using IPsec.

In addition to security considerations involving transmission of the key contents, any logging method(s) used for the key values MUST keep the information secure from intruders. For all implementations, the requirements to address this security concern are:

- Display of the log MUST only be possible with administrative rights to the node.
- Options to disable logging to disk and to keep logs for a fixed duration SHOULD be provided.

Finally, it is important to note that different nodes may have different levels of risk, and these differences may affect the implementation. The components of risk include assets, threats, and vulnerabilities. Consider the following example iSCSI nodes, which demonstrate differences in assets and vulnerabilities of the nodes, and as a result, differences in implementation:

- One iSCSI target based on a special-purpose operating system. Since the iSCSI target controls access to the data storage containing company assets, the asset level is seen as very high. Also, because of the special-purpose operating system, in which vulnerabilities are less well-known, the vulnerability level is viewed as low.

- Multiple iSCSI initiators in a blade farm, each running a general-purpose operating system. The asset level of each node is viewed as low, since blades are replaceable and low cost. However, the vulnerability level is viewed as high, since there are many well-known vulnerabilities to the general-purpose operating system.

For the above target, an appropriate implementation might be logging of received key values, but no transmission of the key. For the initiators, an appropriate implementation might be transmission of the key, but no logging of received key values.

5. IANA Considerations

The standards action of this document updates RFC 3720 to allow any iSCSI extension item, specifically X# extension text keys, Y# digest algorithms, and Z# authentication methods, to be defined by a standards track, experimental, or informational RFC. This document is a standards track RFC that defines an X# extension text key.

IANA registered this key as follows:

- Key Name: X#NodeArchitecture

- Description: Node architecture details

- Reference: [RFC4850]

The update to RFC 3720 to allow additional types of RFCs for iSCSI Extension items has the same effect as if the following changes were made to the text of RFC 3720 (RFC text cannot be changed after publication):

1) In Section 11.1, the requirement that Z# Authentication methods "MUST be described by an informational RFC." is changed to "MUST be described by a standards track RFC, an experimental RFC, or an informational RFC."

2) In Section 12.1, the requirement that Y# Digest algorithms "MUST be described by an informational RFC." is changed to "MUST be described by a standards track RFC, an experimental RFC, or an informational RFC."

3) In Section 12.22, the requirement that X# text keys "MUST be described by an informational RFC." is changed to "MUST be described by a standards track RFC, an experimental RFC, or an informational RFC."

4) In Section 13.3, the description of allowed RFC types for extension items is changed from "The RFC may be informational rather than Standards-Track," to "The RFC MUST be standards track, experimental, or informational,"

5) In Section 13.5.2, the phrase "standards track" is changed to "standards track or experimental" in the last sentence of the first paragraph, so that the sentence reads: "If the specification is a standards track or experimental document, the usual IETF procedures for such documents are followed."

The registries for iSCSI extension items should be managed as if these changes had been made to the text of RFC 3720.

6. References

6.1. Normative References

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

[2] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004.

6.2. Informative References

[3] 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.

[4] Aboba, B., Tseng, J., Walker, J., Rangan, V., and F. Travostino, "Securing Block Storage Protocols over IP", RFC 3723, April 2004.

Appendix A. Acknowledgments

The IP Storage (ips) Working Group in the Transport Area of IETF has been responsible for defining the iSCSI protocol (apart from a host of other relevant IP Storage protocols). The author acknowledges the contributions of the entire working group.

The following individuals directly contributed to identifying issues and/or suggesting resolutions to the issues found in this document: David Black, Mallikarjun Chadalapaka, Paul Koning, Julian Satran, John Hufferd, Claire Kraft, Ranga Sankar, Joseph Pittman, Greg Berg, John Forte, Jim Yuill, William Studenmund, and Ken Sandars. This document benefited from all these contributions.

Finally, the author recognizes Network Appliance, Inc. for sponsorship and support during the development of this work.

Author’s Address

Dave Wysochanski
8311 Brier Creek Parkway
Suite 105-296
Raleigh, NC 27617
US

Phone: +1 919 696 8130
EMail: wysochanski@pobox.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.