                     Bootstrapping Clients using
      the Internet Small Computer System Interface (iSCSI) Protocol

Status of This Memo

Copyright Notice

Abstract

   Internet Small Computer System Interface (iSCSI) is a proposed
   transport protocol for Small Computer Systems Interface (SCSI) that
   operates on top of TCP.  This memo describes a standard mechanism for
   enabling clients to bootstrap themselves using the iSCSI protocol.
   The goal of this standard is to enable iSCSI boot clients to obtain
   the information to open an iSCSI session with the iSCSI boot server.

1.  Introduction

   The Small Computer Systems Interface (SCSI) is a popular family of
   protocols for communicating with I/O devices, especially storage
   devices.  SCSI can be characterized as a request/response messaging
   protocol with a standard architecture and componentized command sets
   for different device classes.

   iSCSI is a proposed transport protocol for SCSI that operates on top
   of TCP.  The role of iSCSI is necessitated by the evolution of the
   system interconnect from a shared bus to a switched network.  IP
   networks meet the architectural and performance requirements of
   transporting SCSI, paving the way for the iSCSI protocol.

Many diskless clients sometimes bootstrap off remote SCSI devices.
Such diskless entities are lightweight, space efficient, and power-
conserving and are increasingly popular in various environments.

This memo describes a standard mechanism for enabling clients to
bootstrap themselves using the iSCSI protocol.  The goal of this
standard is to enable iSCSI boot clients to obtain the information to
open an iSCSI session with the iSCSI boot server.  It is possible
that all the information is not available at the very outset, so the
memo describes steps to obtain the information required to bootstrap
clients off an iSCSI boot server.

1.1.  Keywords

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 [Bradner97].

2.  Requirements

   1. There must be no restriction of network topology between the iSCSI
      boot client and the boot server other than that in effect for
      establishing the iSCSI session.  Consequently, it is possible for
      an iSCSI boot client to boot from an iSCSI boot server behind
      gateways or firewalls as long as it is possible to establish an
      iSCSI session between the client and the server.

   2. The following represent the minimum information required for an
      iSCSI boot client to contact an iSCSI boot server: (a) the
      client's IP address (IPv6 or IPv4); (b) the server's iSCSI Target
      Name; and (c) mandatory iSCSI initiator capability.

   The above assume that the default LUN for the boot process is 0 and
   that the default port for the iSCSI boot server is the well-known
   iSCSI port [Satran02].  However, both may be overridden at the time
   of configuration.

   Additional information may be required at each stage of the boot
   process.

   3. It is possible for the iSCSI boot client to have none of the above
      information or capability on starting.

   4. The client should be able to complete boot without user
      intervention (for boots that occur during an unattended power-up).
      However, there should be a mechanism for the user to input values
      so as to bypass stages of the boot protocol.

   5. Additional protocol software (for example, BOOTP or DHCP) may be
      necessary if the minimum information required for an iSCSI session
      is not provided.

3.  Related Work

   The Reverse Address Resolution Protocol (RARP) [Finlayson84] through
   the extensions defined in the Dynamic RARP (DRARP) [Brownell96]
   explicitly addresses the problem of network address discovery, and
   includes an automatic IP address assignment mechanism.  The Trivial
   File Transfer Protocol (TFTP) [Sollins81] provides for transport of a
   boot image from a boot server.  BOOTP [Croft85, Reynolds93, Wimer93]
   is a transport mechanism for a collection of configuration
   information.  BOOTP is also extensible, and official extensions have
   been defined for several configuration parameters.  DHCPv4 [Droms97,
   Droms93] and DHCPv6 [Droms02] are standards by which hosts are to be
   dynamically configured in an IP network.  The Service Location
   Protocol (SLP) provides for location of higher-level services
   [Guttman99].

4.  Software Stage

   Some iSCSI boot clients may lack the resources to boot up with the
   mandatory iSCSI initiator capability.  Such boot clients may choose
   to obtain iSCSI initiator software from a boot server.  Currently,
   many established protocols allow such a service in order to enable
   clients to load software images.  For example, BOOTP and DHCP servers
   have the capability to provide the locations of servers that can
   serve software images on requests from boot clients.

   Note that this document does not recommend any of the above
   protocols, and the final decision of which boot protocol is to be
   used to load iSCSI initiator software is left to the discretion of
   the implementor.

5.  DHCP Stage

   In order to use an iSCSI boot server, the following pieces of
   information are required for an ISCSI boot client.

   - The IP address of the iSCSI boot client (IPv4 or IPv6)

   - The IP transport endpoint for the iSCSI Target Port for the iSCSI
     boot server.  If the transport is TCP, for example, this has to
     resolve to an IP address and a TCP port number.  TCP is currently
     the only transport approved for iSCSI.

   - The eight-byte LUN structure identifying the Logical Unit within
     the iSCSI boot server.

   At boot time, all or none of this information may be stored in the
   iSCSI boot client.  This section describes techniques for obtaining
   the required information via the DHCP stage.  Otherwise, if the iSCSI
   boot client has all the information, the boot client may proceed
   directly to the Boot stage.

   An iSCSI boot client that does not know its IP address at power-on
   may acquire it via BOOTP or DHCP (v4 or v6), or via IPv6 address
   autoconfiguration.  Please note that DHCP settings (such as the RA
   settings in DHCPv6) may prohibit the use of DHCP in distributing
   iSCSI boot information; in this case, the DHCP stage cannot be used.

   Unless specified otherwise here, BOOTP or DHCP fields such as the
   client ID and gateway information are used in an identical way as
   applications other than iSCSI.

   A BOOTP or DHCP server (v4 or v6) MAY instruct an iSCSI client how to
   reach its boot device.  This is done using the variable-length option
   named Root Path [Alexander93, Reynolds93].  The use of the option
   field is reserved for iSCSI boot use by prefacing the string with
   "iscsi:".  The Root Path option is not currently defined for DHCPv6;
   if the option is defined for DHCPv6 in the future, the use of the
   option as defined for iSCSI boot will apply.

   The option field consists of an UTF-8 [Yergeau98] string.  The string
   has the following composition:

   "iscsi:"<servername>":"<protocol>":"<port>":"<LUN>":"<targetname>

   The fields "servername", "port", "protocol", and "LUN" are OPTIONAL
   and should be left blank if there are no corresponding values.  The
   "targetname" field is not optional and MUST be provided.

   The "servername" is the name of iSCSI server and contains either a
   valid domain name, a literal IPv4 address, or a literal IPv6 address.
   The servername must follow the specifications outlined in Section
   3.2.2 of the URI Specification [Lee98] [Hinden99].  The characters
   allowed must also conform to Section 2.2 of the same specification.
   Servername compression MUST NOT be used in this field.

   The "protocol" field is the decimal representation of the IANA-
   approved string for the transport protocol to be used for iSCSI.  If
   the protocol field is left bank, the default value is assumed to be

   "6" for TCP.  The transport protocol MUST have been approved for use
   in iSCSI; currently, the only approved protocol is TCP.

   The "port" is the decimal representation of the port on which the
   iSCSI boot server is listening.  If not specified, the port defaults
   to the well-known iSCSI port [Satran02].

   The "LUN" field is a hexadecimal representation of the LU number.  If
   the LUN field is blank, then LUN 0 is assumed.  If the LUN field is
   not blank, the representation MUST be divided into four groups of
   four hexadecimal digits, separated by "-".  Digits above 9 may be
   either lower or upper case.  An example of such a representation
   would be 4752-3A4F-6b7e-2F99.  For the sake of brevity, at most three
   leading zero ("0") digits MAY be omitted in any group of hexadecimal
   digits.  Thus, the "LUN" representation 6734-9-156f-127 is equivalent
   to 6734-0009-156f-0127.  Furthermore, trailing groups containing only
   the "0" digit MAY be omitted along with the preceding "-".  So, the
   "LUN" representation 4186-9 is equivalent to 4186-0009-0000-0000.
   Other concise representations of the LUN field MUST NOT be used.

   Note that SCSI targets are allowed to present different LU numberings
   for different SCSI initiators, so to our knowledge nothing precludes
   a SCSI target from exporting several different LUs to several
   different SCSI initiators as their respective LUN 0s.

   The "targetname" field is an iSCSI Name that is defined by the iSCSI
   standard [Satran02] to uniquely identify an iSCSI target.  The
   approved characters in the targetname field are stated in the iSCSI
   String Profile document[Bakke04].

   If the "servername" field is provided by BOOTP or DHCP, then that
   field is used in conjunction with other associated fields to contact
   the boot server in the Boot stage (Section 7).  However, if the
   "servername" field is not provided, then the "targetname" field is
   then used in the Discovery Service stage in conjunction with other
   associated fields (Section 6).

6.  Discovery Service Stage

   This stage is required if the BOOTP or DHCP server (v4 or v6) is
   unaware of any iSCSI boot servers or if the BOOTP or DHCP server is
   unable to provide the minimum information required to connect to the
   iSCSI boot server, other than the targetname.

   The Discovery Service may be based on the SLP protocol [Guttman99,
   Bakke02] and is an instantiation of the SLP Service or Directory
   Agent.  Alternatively, the Discovery Service may be based on the iSNS
   protocol [Tseng03] and is an instantiation of the iSNS Server.

The iSCSI boot client may have obtained the targetname of the iSCSI
boot server in the DHCP stage (Section 5).  In that case, the iSCSI
boot client queries the SLP Discovery Service using query string 1 of
the iSCSI Target Concrete Service Type Template, as specified in
Section 6.2 of the iSCSI SLP interaction document [Bakke02], to
resolve the targetname to an IP address and port number.
Alternatively, the iSCSI boot client may query the iSNS Discovery
Service with a Device Attribute Query with the targetname as the
query parameter [Tseng03].  Once this is obtained, the iSCSI boot
client proceeds to the Boot stage (Section 7).

It is possible that the port number obtained from the Discovery
Service may conflict with the one obtained from the DHCP stage.  In
such a case, the implementor has the option to try both port numbers
in the Boot stage.

If the iSCSI boot client does not have any targetname information,
the iSCSI boot client may then query the SLP Discovery Service with
query string 4 of the iSCSI Target Concrete Service Type Template, as
specified in Section 6.2 of the iSCSI SLP interaction document
[Bakke02].  In response to this query, the SLP Discovery Service
provides the boot client with a list of iSCSI boot servers the boot
client is allowed to access.  Alternatively, the iSCSI boot client
can query the iSNS Discovery Service to verify if the targets in
particular Discovery Domain are bootable [Tseng03].

If the list of iSCSI boot servers is empty, subsequent actions are
left to the discretion of the implementor.  Otherwise, the iSCSI boot
client may contact any iSCSI boot server in the list.  Moreover, the
order in which iSCSI boot servers are contacted is also left to the
discretion of the implementor.

7.  Boot Stage

Once the iSCSI boot client has obtained the minimum information to
open an iSCSI session with the iSCSI boot server, the actual booting
process can start.

The actual sequence of iSCSI commands that are needed to complete the
boot process is left to the implementor.  This was done because of
varying requirements from different vendors and equipment, making it
difficult to specify a common subset of the iSCSI standard that would
be acceptable to everybody.

The iSCSI session established for boot may be taken over by the
booted software in the iSCSI boot client.

8.  Security Considerations

   The security discussion is centered around securing the communication
   involved in the iSCSI boot process.

   However, the issue of applying credentials to a boot image loaded
   through the iSCSI boot mechanism is outside the scope of this
   document.  One key difference between the iSCSI boot mechanism and
   BOOTP-based image loading is the fact that the identity of a boot
   image may not be known when the Boot stage starts.  The identity of
   certain boot images and their locations are known only after the
   contents of a boot disk exposed by the iSCSI boot service are
   examined.  Furthermore, images themselves may recursively load other
   images based on both hardware configurations and user input.
   Consequently, a practical way to verify loaded boot images is to make
   sure that each image loading software verifies the image to be loaded
   using a mechanism of their choice.

   The considerations involved in designing a security architecture for
   the iSCSI boot process include configuration, deployment, and
   provisioning issues apart from typical security considerations.
   Enabling iSCSI boot creates a critical operational dependence on an
   external system with obvious security implications, and thus
   administrator awareness of this enablement is extremely important.
   Therefore, iSCSI boot SHOULD NOT be enabled or put high in the boot
   order without an explicit administrative action.

   In all phases of the boot process, a client must ensure that a server
   is authorized to send it certain information.  This means that the
   authenticated identity of a server must have an authorization
   indication.  A list of authorized servers can be pre-configured into
   a client, or the list can be downloaded in an authenticated form from
   a prior stage in the boot process.

   The software stage SHOULD NOT be involved in a secure iSCSI boot
   process, as this would add the additional complexity of trying to
   secure the process of loading the software necessary to run the later
   stages of iSCSI boot.  Authentication and integrity protection of
   downloaded boot software has proven to be difficult and complex due
   to administrative issues and limitations of the BIOS environment.  It
   is therefore assumed that all the necessary software is resident on
   the iSCSI boot client.

   If the DHCP stage is implemented using the DHCP protocol, the iSCSI
   boot client SHOULD implement the DHCP authentication ([Droms01],
   [Droms02] for IPv6).  In this case, an administration interface
   SHOULD be provided for the configuration of the DHCP authentication
   credentials, both when the network interface is on the motherboard

and when it is removable.  Note that DHCP authentication
([Droms01],[Droms02] for IPv6) is focused on intra-domain
authentication, which is assumed to be enough for iSCSI boot
scenarios.  In the context of the secure iSCSI boot process, the
reply from the DHCP server in the DHCP stage SHOULD include the
serverName in IPv4 (or IPv6) format to avoid reliance on a DNS server
(for resolving names) or a Discovery Service entity (to look up
targetnames).  This reduces the number of entities involved in the
secure iSCSI boot process.

If the Discovery Service stage is implemented using SLP, the iSCSI
boot client SHOULD provide IPsec support (OPTIONAL to use) for the
SLP protocol, as defined in [Bakke02] and [Aboba03].  If the
Discovery Service stage is implemented using iSNS, the iSCSI boot
client SHOULD provide IPsec support (OPTIONAL to use) for the iSNS
protocol, as defined in [Tseng03] and [Aboba03].  When iSNS or SLP
are used to distribute security policy or configuration information,
at a minimum, per-packet data origin authentication, integrity, and
replay protection SHOULD be used to protect the discovery protocol.

For the final communication between the iSCSI boot client and the
iSCSI boot server in the Boot stage, IPsec and in-band authentication
SHOULD be implemented according to the guidelines in the main iSCSI
draft [Satran02] and [Aboba03].  Due to memory constraints, it is
expected that iSCSI boot clients will only support the pre-shared key
authentication in IKE.  Where the host IP address is assigned
dynamically, IKE main mode SHOULD NOT be used, as explained in
[Satran02] and [Aboba03].  Regardless of the way parameters in
previous stages (DHCP, SLP, iSNS) were obtained (securely or not),
the iSCSI boot session is vulnerable as any iSCSI session (see
[Satran02] and [Aboba03] for iSCSI security threats).  Therefore,
security for this session SHOULD be configured and used according to
[Satran02] and [Aboba03] guidelines.

Note that if a boot image inherits an iSCSI session from a previously
loaded boot image, it also inherits the security properties of the
iSCSI session.

Acknowledgements

We wish to thank John Hufferd for taking the initiative to form the
iSCSI boot team.  We also wish to thank Doug Otis, Julian Satran,
Bernard Aboba, David Robinson, Mark Bakke, Ofer Biran, and
Mallikarjun Chadalapaka for helpful suggestions and pointers
regarding the draft document.

Normative References

   [Aboba03]      Aboba, B., Tseng, J., Walker, J., Rangan, V., and F.
                  Travostino, "Securing Block Storage Protocols over
                  IP", RFC 3723, April 2004.

   [Alexander93]  Alexander, S. and R. Droms, "DHCP Options and BOOTP
                  Vendor Extensions", RFC 2132, March 1997.

   [Bakke02]      Bakke, M., Hufferd, J., Voruganti, K., Krueger, M.,
                  and T. Sperry, "Finding Internet Small Computer
                  Systems Interface (iSCSI) Targets and Name Servers by
                  Using Service Location Protocol version 2 (SLPv2)",
                  RFC 4018, April 2005.

   [Bakke04]      Bakke, M., "String Profile for Internet Small Computer
                  Systems Interface (iSCSI) Names", RFC 3722, April
                  2004.

   [Bradner97]    Bradner, S., "Key words for use in RFCs to Indicate
                  Requirement Levels", BCP 14, RFC 2119, March 1997.

   [Croft85]      Croft, W. and J. Gilmore, "Bootstrap Protocol", RFC
                  951, September 1985.

   [Droms93]      Droms, R., "Interoperation Between DHCP and BOOTP",
                  RFC 1534, October 1993.

   [Droms97]      Droms, R., "Dynamic Host Configuration Protocol", RFC
                  2131, March 1997.

   [Droms01]      Droms, R. and W. Arbaugh, "Authentication for DHCP
                  Messages", RFC 3118, June 2001.

   [Droms02]      Droms, R., Bound, J., Volz, B., Lemon, T., Perkins,
                  C., and M. Carney, "Dynamic Host Configuration
                  Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

   [Guttman99]    Guttman, E., Perkins, C., Veizades, J., and M. Day,
                  "Service Location Protocol, Version 2", RFC 2608, June
                  1999.

   [Hinden99]     Hinden, R., Carpenter, B., and L. Masinter, "Format
                  for Literal IPv6 Addresses in URL's", RFC 2732,
                  December 1999.

   [Lee98]            Berners-Lee, T., Fielding, R., and L. Masinter,
                      "Uniform Resource Identifiers (URI): Generic Syntax",
                      RFC 2396, August 1998.

   [Reynolds93]       Reynolds, J., "BOOTP Vendor Information Extensions",
                      RFC 1497, August 1993.

   [Satran02]         Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka,
                      M., and E. Zeidner, "Internet Small Computer Systems
                      Interface (iSCSI)", RFC 3720, April 2004.

   [Tseng03]          Tseng, J., Gibbons, K., Travostino, F., Du Laney, C.,
                      and J. Souza, "Internet Storage Name Service (iSNS)",
                      RFC 4171, April 2005.

   [Yergeau98]        Yergeau, F., "UTF-8, a transformation format of ISO
                      10646", STD 63, RFC 3629, November 2003.

   [Wimer93]          Wimer, W., "Clarifications and Extensions for the
                      Bootstrap Protocol", RFC 1542, October 1993.

Informative References

   [Brownell96]       Brownell, D., "Dynamic RARP Extensions for Automatic
                      Network Address Acquisition", RFC 1931, April 1996.

   [Finlayson84]      Finlayson, R., Mann, T., Mogul, J., and M. Theimer,
                      "Reverse Address Resolution Protocol", STD 38, RFC
                      903, June 1984.

   [Sollins81]        Sollins, K., "The TFTP Protocol (Revision 2)", STD 33,
                      RFC 1350, July 1992.

Authors' Addresses

   Prasenjit Sarkar
   IBM Almaden Research Center
   650 Harry Road
   San Jose, CA 95120, USA

   Phone: +1 408 927 1417
   EMail: psarkar@almaden.ibm.com


   Duncan Missimer
   Hewlett-Packard Company
   10955 Tantau Ave
   Cupertino, CA 95014, USA

   EMail: duncan.missimer@ieee.org


   Constantine Sapuntzakis
   Stanford University
   353 Serra Hall #407
   Stanford, CA 94305, USA

   EMail: csapuntz@alum.mit.edu

Full Copyright Statement

Intellectual Property

Acknowledgement