Network Working Group
Request for Comments: 3721
Category: Informational

M. Bakke
Cisco
J. Hafner
J. Hufferd
K. Voruganti
IBM
M. Krueger
Hewlett-Packard

April 2004

Internet Small Computer Systems Interface (iSCSI) Naming and Discovery

Status of this Memo

Copyright Notice

Abstract

This document provides examples of the Internet Small Computer Systems Interface (iSCSI; or SCSI over TCP) name construction and discussion of discovery of iSCSI resources (targets) by iSCSI initiators. This document complements the iSCSI protocol document. Flexibility is the key guiding principle behind this document. That is, an effort has been made to satisfy the needs of both small isolated environments, as well as large environments requiring secure/scalable solutions.

Table of Contents

1. iSCSI Names and Addresses

The main addressable, discoverable entity in iSCSI is an iSCSI Node. An iSCSI node can be either an initiator, a target, or both. The rules for constructing an iSCSI name are specified in [RFC3720].

This document provides examples of name construction that might be used by a naming authority.

Both targets and initiators require names for the purpose of identification, so that iSCSI storage resources can be managed regardless of location (address). An iSCSI name is the unique identifier for an iSCSI node, and is also the SCSI device name [SAM2] of an iSCSI device. The iSCSI name is the principal object used in authentication of targets to initiators and initiators to targets. This name is also used to identify and manage iSCSI storage resources.

Furthermore, iSCSI names are associated with iSCSI nodes instead of with network adapter cards to ensure the free movement of network HBAs between hosts without loss of SCSI state information (reservations, mode page settings etc) and authorization configuration.

An iSCSI node also has one or more addresses. An iSCSI address specifies a single path to an iSCSI node and consists of the iSCSI name, plus a transport (TCP) address which uses the following format:

<domain-name>[:<port>]

Where <domain-name> is one of:

- IPv4 address, in dotted decimal notation. Assumed if the name contains exactly four numbers, separated by dots (.), where each number is in the range 0..255.

- IPv6 address, in colon-separated hexadecimal notation, as specified in [RFC3513] and enclosed in "[" and "]" characters, as specified in [RFC2732].

- Fully Qualified Domain Name (host name). Assumed if the <domain-name> is neither an IPv4 nor an IPv6 address.

For iSCSI targets, the <port> in the address is optional; if specified, it is the TCP port on which the target is listening for connections. If the <port> is not specified, the default port 3260, assigned by IANA, will be assumed. For iSCSI initiators, the <port> is omitted.

Examples of addresses:

```
192.0.2.2
192.0.2.23:5003
[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]
[1080:0:0:0:8:800:200C:417A]
[3ffe:2a00:100:7031::1]
[1080::8:800:200C:417A]
[1080::8:800:200C:417A]:3260
[::192.0.2.5]
mydisks.example.com
moredisks.example.com:5003
```

The concepts of names and addresses have been carefully separated in iSCSI:

- An iSCSI Name is a location-independent, permanent identifier for an iSCSI node. An iSCSI node has one iSCSI name, which stays constant for the life of the node. The terms "initiator name" and "target name" also refer to an iSCSI name.

- An iSCSI Address specifies not only the iSCSI name of an iSCSI node, but also a location of that node. The address consists of a host name or IP address, a TCP port number (for the target), and the iSCSI Name of the node. An iSCSI node can have any number of addresses, which can change at any time, particularly if they are assigned via DHCP.

A similar analogy exists for people. A person in the USA might be:

```
Robert Smith
SSN+DateOfBirth: 333-44-5555 14-MAR-1960
Phone: +1 (763) 555.1212
Home Address: 555 Big Road, Minneapolis, MN 55444
Work Address: 222 Freeway Blvd, St. Paul, MN 55333
```

In this case, Robert's globally unique name is really his Social Security Number plus Date of Birth. His common name, "Robert Smith", is not guaranteed to be unique. Robert has three locations at which he may be reached; two Physical addresses, and a phone number.

In this example, Robert's SSN+DOB is like the iSCSI Name (date of birth is required to disambiguate SSNs that have been reused), his phone number and addresses are analogous to an iSCSI node's TCP addresses, and "Robert Smith" would be a human-friendly label for this person.

To assist in providing a more human-readable user interface for devices that contain iSCSI targets and initiators, a target or initiator may also provide an alias. This alias is a simple UTF-8 string, is not globally unique, and is never interpreted or used to identify an initiator or device within the iSCSI protocol. Its use is described further in section 2.

1.1. Constructing iSCSI names using the iqn. format

The iSCSI naming scheme was constructed to give an organizational naming authority the flexibility to further subdivide the responsibility for name creation to subordinate naming authorities. The iSCSI qualified name format is defined in [RFC3720] and contains (in order):

- The string "iqn."
- A date code specifying the year and month in which the organization registered the domain or sub-domain name used as the naming authority string.
- The organizational naming authority string, which consists of a valid, reversed domain or subdomain name.
- Optionally, a ':', followed by a string of the assigning organization's choosing, which must make each assigned iSCSI name unique.

The following is an example of an iSCSI qualified name from an equipment vendor:

```
     Organizational      Subgroup Naming Authority
             Naming      and/or string Defined by
Type  Date     Auth      Org. or Local Naming Authority
+--++-----+ +---------+ +--------------------------------+
|  ||     | |         | |                                |
```

iqn.2001-04.com.example:diskarrays-sn-a8675309

Where:

"iqn" specifies the use of the iSCSI qualified name as the authority.

"2001-04" is the year and month on which the naming authority acquired the domain name used in this iSCSI name. This is used to ensure that when domain names are sold or transferred to another organization, iSCSI names generated by these organizations will be unique.

"com.example" is a reversed DNS name, and defines the organizational naming authority. The owner of the DNS name "example.com" has the sole right of use of this name as this part of an iSCSI name, as well as the responsibility to keep the remainder of the iSCSI name unique. In this case, example.com happens to manufacture disk arrays.

"diskarrays" was picked arbitrarily by example.com to identify the disk arrays they manufacture. Another product that ACME makes might use a different name, and have its own namespace independent of the disk array group. The owner of "example.com" is responsible for keeping this structure unique.

"sn" was picked by the disk array group of ACME to show that what follows is a serial number. They could have just assumed that all iSCSI Names are based on serial numbers, but they thought that perhaps later products might be better identified by something else. Adding "sn" was a future-proof measure.

"a8675309" is the serial number of the disk array, uniquely identifying it from all other arrays.

Another example shows how the ':' separator helps owners of sub-domains to keep their name spaces unique:

```
                Naming            Defined by
Type  Date       Authority         Naming Authority
+--++-----+ +-----------------+ +-----------+
|  ||     | |                 | |           |

iqn.2001-04.com.example.storage:tape.sys1.xyz

                Naming                Defined by
Type  Date       Authority            Naming Authority
+--++-----+ +----------------------+ +-----------+
|  ||     | |                      | |           |

iqn.2001-04.com.example.storage.tape:sys1.xyz
```

Note that, except for the ':' separator, both names are identical. The first was assigned by the owner of the subdomain "storage.example.com"; the second was assigned by the owner of "tape.storage.example.com". These are both legal names, and are unique.

The following is an example of a name that might be constructed by a research organization:

```
              Naming        Defined by  Defined by
Type  Date    Authority      cs dept    User "oaks"
 +-+ +-----+ +------------+ +--------+ +-----------+
 | | |     | |            | |        | |           |
 iqn.2000-02.edu.example.cs:users.oaks:proto.target4
```

In the above example, Professor Oaks of Example University is building research prototypes of iSCSI targets. EU's computer science department allows each user to use his or her user name as a naming authority for this type of work, by attaching "users.<username>" after the ':', and another ':', followed by a string of the user's choosing (the user is responsible for making this part unique). Professor Oaks chose to use "proto.target4" for this particular target.

The following is an example of an iSCSI name string from a storage service provider:

```
              Organization           String
                 Naming           Defined by Org.
Type  Date    Authority          Naming Authority
 +-+ +-----+ +-------------+ +----------------------+
 | | |     | |             | |                      |
 iqn.1995-11.com.example.ssp:customers.4567.disks.107
```

In this case, a storage service provider (ssp.example.com) has decided to re-name the targets from the manufacturer, to provide the flexibility to move the customer's data to a different storage subsystem should the need arise.

The Storage Service Provider (SSP) has configured the iSCSI Name on this particular target for one of its customers, and has determined that it made the most sense to track these targets by their Customer ID number and a disk number. This target was created for use by customer #4567, and is the 107th target configured for this customer.

Note that when reversing these domain names, the first component (after the "iqn.") will always be a top-level domain name, which includes "com", "edu", "gov", "org", "net", "mil", or one of the

two-letter country codes. The use of anything else as the first component of these names is not allowed. In particular, companies generating these names must not eliminate their "com." from the string.

Again, these iSCSI names are NOT addresses. Even though they make use of DNS domain names, they are used only to specify the naming authority. An iSCSI name contains no implications of the iSCSI target or initiator's location. The use of the domain name is only a method of re-using an already ubiquitous name space.

1.2. Constructing iSCSI names using the eui. format

The iSCSI eui. naming format allows a naming authority to use IEEE EUI-64 identifiers in constructing iSCSI names. The details of constructing EUI-64 identifiers are specified by the IEEE Registration Authority (see [EUI64]).

Example iSCSI name:

```
Type  EUI-64 identifier (ASCII-encoded hexadecimal)
+--++--------------+
|  ||               |
eui.02004567A425678D
```

2. iSCSI Alias

The iSCSI alias is a UTF-8 text string that may be used as an additional descriptive name for an initiator and target. This may not be used to identify a target or initiator during login, and does not have to follow the uniqueness or other requirements of the iSCSI name. The alias strings are communicated between the initiator and target at login, and can be displayed by a user interface on either end, helping the user tell at a glance whether the initiators and/or targets at the other end appear to be correct. The alias must NOT be used to identify, address, or authenticate initiators and targets.

The alias is a variable length string, between 0 and 255 characters, and is terminated with at least one NULL (0x00) character, as defined in [RFC3720]. No other structure is imposed upon this string.

2.1. Purpose of an Alias

Initiators and targets are uniquely identified by an iSCSI Name. These identifiers may be assigned by a hardware or software manufacturer, a service provider, or even the customer. Although these identifiers are nominally human-readable, they are likely to be assigned from a point of view different from that of the other side

of the connection. For instance, a target name for a disk array may be built from the array's serial number, and some sort of internal target ID. Although this would still be human-readable and transcribable, it offers little assurance to someone at a user interface who would like to see "at-a-glance" whether this target is really the correct one.

The use of an alias helps solve that problem. An alias is simply a descriptive name that can be assigned to an initiator or target, that is independent of the name, and does not have to be unique. Since it is not unique, the alias must be used in a purely informational way. It may not be used to specify a target at login, or used during authentication.

Both targets and initiators may have aliases.

2.2. Target Alias

To show the utility of an alias, here is an example using an alias for an iSCSI target.

Imagine sitting at a desktop station that is using some iSCSI devices over a network. The user requires another iSCSI disk, and calls the storage services person (internal or external), giving any authentication information that the storage device will require for the host. The services person allocates a new target for the host, and sends the Target Name for the new target, and probably an address, back to the user. The user then adds this Target Name to the configuration file on the host, and discovers the new device.

Without an alias, a user managing an iSCSI host would click on some sort of management "show targets" button to show the targets to which the host is currently connected.

```
+--Connected-To-These-Targets----------------------
|
|  Target Name
|
|  iqn.1995-04.com.example:sn.5551212.target.450
|  iqn.1995-04.com.example:sn.5551212.target.489
|  iqn.1995-04.com.example:sn.8675309
|  iqn.2001-04.com.example.storage:tape.sys1.xyz
|  iqn.2001-04.com.example.storage.tape:sys1.xyz
|
+-------------------------------------------------
```

In the above example, the user sees a collection of iSCSI Names, but with no real description of what they are for. They will, of course, map to a system-dependent device file or drive letter, but it's not easy looking at numbers quickly to see if everything is there.

If a storage administrator configures an alias for each target name, the alias can provide a more descriptive name. This alias may be sent back to the initiator as part of the login response, or found in the iSCSI MIB. It then might be used in a display such as the following:

```
+--Connected-To-These-Targets----------------------
|
|  Alias          Target Name
|
|  Oracle 1       iqn.1995-04.com.example:sn.5551212.target.450
|  Local Disk     iqn.1995-04.com.example:sn.5551212.target.489
|  Exchange 2     iqn.1995-04.com.example:sn.8675309
|
+-------------------------------------------------
```

This would give the user a better idea of what's really there.

In general, flexible, configured aliases will probably be supported by larger storage subsystems and configurable gateways. Simpler devices will likely not keep configuration data around for things such as an alias. The TargetAlias string could be either left unsupported (not given to the initiator during login) or could be returned as whatever the "next best thing" that the target has that might better describe it. Since it does not have to be unique, it could even return SCSI inquiry string data.

Note that if a simple initiator does not wish to keep or display alias information, it can be simply ignored if seen in the login response.

2.3. Initiator Alias

An initiator alias can be used in the same manner as a target alias. An initiator may send the alias in a login request, when it sends its iSCSI Initiator Name. The alias is not used for authentication, but may be kept with the session information for display through a management Graphical User Interface (GUI) or command-line interface (for a more complex subsystem or gateway), or through the iSCSI MIB.

Note that a simple target can just ignore the Initiator Alias if it has no management interface on which to display it.

Usually just the hostname would be sufficient for an initiator alias, but a custom alias could be configured for the sake of the service provider if needed. Even better would be a description of what the machine was used for, such as "Exchange Server 1", or "User Web Server".

Here's an example of a management interface showing a list of sessions on an iSCSI target network entity. For this display, the targets are using an internal target number, which is a fictional field that has purely internal significance.

```
+--Connected-To-These-Initiators-------------------
|
|  Target   Initiator Name
|
|  450      iqn.1995-04.com.example.sw:cd.12345678-OEM-456
|  451      iqn.1995-04.com.example.os:hostid.A598B45C
|  309      iqn.1995-04.com.example.sw:cd.87654321-OEM-259
|
+-------------------------------------------------
```

And with the initiator alias displayed:

```
+--Connected-To-These-Initiators-------------------
|
|  Target Alias               Initiator Name
|
|  450    Web Server 4        iqn.1995-04.com.example.sw:cd.12...
|  451    scsigw.example.com  iqn.1995-04.com.example.os:hosti...
|  309    Exchange Server     iqn.1995-04.com.example.sw:cd.87...
|
+-------------------------------------------------
```

This gives the storage administrator a better idea of who is connected to their targets. Of course, one could always do a reverse DNS lookup of the incoming IP address to determine a host name, but simpler devices really don't do well with that particular feature due to blocking problems, and it won't always work if there is a firewall or iSCSI gateway involved.

Again, these are purely informational and optional and require a management application.

Aliases are extremely easy to implement. Targets just send a TargetAlias whenever they send a TargetName. Initiators just send an InitiatorAlias whenever they send an InitiatorName. If an alias is received that does not fit, or seems invalid in any way, it is ignored.

3. iSCSI Discovery

The goal of iSCSI discovery is to allow an initiator to find the targets to which it has access, and at least one address at which each target may be accessed. This should generally be done using as little configuration as possible. This section defines the discovery mechanism only; no attempt is made to specify central management of iSCSI devices within this document. Moreover, the iSCSI discovery mechanisms listed here only deal with target discovery and one still needs to use the SCSI protocol for LUN discovery.

In order for an iSCSI initiator to establish an iSCSI session with an iSCSI target, the initiator needs the IP address, TCP port number and iSCSI target name information. The goal of iSCSI discovery mechanisms are to provide low overhead support for small iSCSI setups, and scalable discovery solutions for large enterprise setups. Thus, there are several methods that may be used to find targets ranging from configuring a list of targets and addresses on each initiator and doing no discovery at all, to configuring nothing on each initiator, and allowing the initiator to discover targets dynamically. The various discovery mechanisms differ in their assumptions about what information is already available to the initiators and what information needs to be still discovered.

iSCSI supports the following discovery mechanisms:

a. Static Configuration: This mechanism assumes that the IP address, TCP port and the iSCSI target name information are already available to the initiator. The initiators need to perform no discovery in this approach. The initiator uses the IP address and the TCP port information to establish a TCP connection, and it uses the iSCSI target name information to establish an iSCSI session. This discovery option is convenient for small iSCSI setups.

b. SendTargets: This mechanism assumes that the target’s IP address and TCP port information are already available to the initiator. The initiator then uses this information to establish a discovery session to the Network Entity. The initiator then subsequently issues the SendTargets text command to query information about the iSCSI targets available at the particular Network Entity (IP address). SendTargets command details can be found in the iSCSI document [RFC3720]. This discovery option is convenient for iSCSI gateways and routers.

c. Zero-Configuration: This mechanism assumes that the initiator does not have any information about the target. In this option, the initiator can either multicast discovery messages directly to the

targets or it can send discovery messages to storage name servers. Currently, there are many general purpose discovery frameworks available such as Salutation [John], Jini [John], UPnP [John], SLP [RFC2608] and iSNS [iSNS]. However, with respect to iSCSI, SLP can clearly perform the needed discovery functions [iSCSI-SLP], while iSNS [iSNS] can be used to provide related management functions including notification, access management, configuration, and discovery management. iSCSI equipment that need discovery functions beyond SendTargets should at least implement SLP, and then consider iSNS when extended discovery management capabilities are required such as in larger storage networks. It should be noted that since iSNS will support SLP, iSNS can be used to help manage the discovery information returned by SLP.

4. Security Considerations

Most security issues relating to iSCSI naming are discussed in the main iSCSI document [RFC3720] and the iSCSI security document [RFC3723].

In addition, Appendix B discusses naming and discovery issues when gateways, proxies, and firewalls are used to solve security or discovery issues in some situations where iSCSI is deployed.

iSCSI allows several different authentication methods to be used. For many of these methods, an authentication identifier is used, which may be different from the iSCSI node name of the entity being authenticated. This is discussed in more detail in Appendix C.

5. References

5.1. Normative References

[RFC3720] Satran, J., Meth, K., Sapuntzakis, C. Chadalapaka, M. and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004.

[EUI64] EUI - "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

[SAM2] R. Weber et al, INCITS T10 Project 1157-D revision 24, "SCSI Architectural Model - 2 (SAM-2)", Section 4.7.6 "SCSI device name", September 2002.

5.2. Informative References

[RFC2608] Guttman, E., Perkins, C., Veizades, J. and M. Day, "SLP Version 2", RFC 2608, June 1999.

[RFC2732] Hinden, R., Carpenter, B. and L. Masinter, "Format for Literal IPv6 Addresses in URL's", RFC 2732, December 1999.

[RFC2979] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

[RFC3303] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A. and A. Rayhan, "Middlebox Communication Architecture and Framework", RFC 3303, August 2002.

[RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 Addressing Architecture", RFC 3513, April 2003.

[RFC3723] Aboba, B., Tseng, J., Walker, J., Rangan, V. and F. Travostino, "Securing Block Storage Protocols over IP", RFC 3723, April 2004.

[iSCSI-SLP] Bakke, M., et al., "Finding iSCSI Targets and Name Servers using SLP", Work in Progress, March 2003.

[iSNS] Tseng, J., et al., "Internet Storage Name Service (iSNS)", Work in Progress, January 2003.

[John] R. John, "UPnP, Jini and Salutation- A look at some popular coordination frameworks for future networked devices", http://www.cswl.com/whiteppr/tech/upnp.html", June 17, 1999.

6. Acknowledgements

Joe Czap (IBM), Howard Hall (Pirus), Jack Harwood (EMC), Yaron Klein (SANRAD), Larry Lamers (Adaptec), Josh Tseng (Nishan Systems), and Todd Sperry (Adaptec) have participated and made contributions during development of this document.

Appendix A: iSCSI Naming Notes

Some iSCSI Name Examples for Targets

- Assign to a target based on controller serial number

  iqn.2001-04.com.example:diskarray.sn.8675309

- Assign to a target based on serial number

  iqn.2001-04.com.example:diskarray.sn.8675309.oracle-db-1

Where oracle-db-1 might be a target label assigned by a user.

This would be useful for a controller that can present different logical targets to different hosts.

Obviously, any naming authority may come up with its own scheme and hierarchy for these names, and be just as valid.

A target iSCSI Name should never be assigned based on interface hardware, or other hardware that can be swapped and moved to other devices.

Some iSCSI Name Examples for Initiators

- Assign to the OS image by fully qualified host name

  iqn.2001-04.com.example.os:dns.com.customer1.host-four

Note the use of two FQDNs - that of the naming authority and also that of the host that is being named. This can cause problems, due to limitations imposed on the size of the iSCSI Name.

- Assign to the OS image by OS install serial number

  iqn.2001-04.com.example.os:newos5.12345-OEM-0067890-23456

Note that this breaks if an install CD is used more than once. Depending on the O/S vendor’s philosophy, this might be a feature.

- Assign to the Raid Array by a service provider

  iqn.2001-04.com.example.myssp:users.mbakke05657

Appendix B: Interaction with Proxies and Firewalls

iSCSI has been designed to allow SCSI initiators and targets to communicate over an arbitrary IP network. This means that in theory, making some assumptions about authentication and security, the whole internet could be used as one giant storage network.

However, there are many access and scaling problems that would come up when this is attempted.

1. Most iSCSI targets may only be meant to be accessed by one or a few initiators. Discovering everything would be unnecessary.

2. The initiator and target may be owned by separate entities, each with their own directory services, authentication, and other schemes. An iSCSI-aware proxy may be required to map between these things.

3. Many environments use non-routable IP addresses, such as the "10." network.

For these and other reasons, various types of firewalls [RFC2979] and proxies will be deployed for iSCSI, similar in nature to those already handling protocols such as HTTP and FTP.

B.1. Port Redirector

A port redirector is a stateless device that is not aware of iSCSI. It is used to do Network Address Translation (NAT), which can map IP addresses between routable and non-routable domains, as well as map TCP ports. While devices providing these capabilities can often filter based on IP addresses and TCP ports, they generally do not provide meaningful security, and are used instead to resolve internal network routing issues.

Since it is entirely possible that these devices are used as routers and/or aggregators between a firewall and an iSCSI initiator or target, iSCSI connections must be operable through them.

Effects on iSCSI:

- iSCSI-level data integrity checks must not include information from the TCP or IP headers, as these may be changed in between the initiator and target.

- iSCSI messages that specify a particular initiator or target, such as login requests and third party requests, should specify the initiator or target in a location-independent manner. This is accomplished using the iSCSI Name.

- When an iSCSI discovery connection is to be used through a port redirector, a target will have to be configured to return a domain name instead of an IP address in a SendTargets response, since the port redirector will not be able to map the IP address(es) returned in the iSCSI message. It is a good practice to do this anyway.

B.2. SOCKS server

A SOCKS server can be used to map TCP connections from one network domain to another. It is aware of the state of each TCP connection.

The SOCKS server provides authenticated firewall traversal for applications that are not firewall-aware. Conceptually, SOCKS is a "shim-layer" that exists between the application (i.e., iSCSI) and TCP.

To use SOCKS, the iSCSI initiator must be modified to use the encapsulation routines in the SOCKS library. The initiator then opens up a TCP connection to the SOCKS server, typically on the canonical SOCKS port 1080. A sub-negotiation then occurs, during which the initiator is either authenticated or denied the connection request. If authenticated, the SOCKS server then opens a TCP connection to the iSCSI target using addressing information sent to it by the initiator in the SOCKS shim. The SOCKS server then forwards iSCSI commands, data, and responses between the iSCSI initiator and target.

Use of the SOCKS server requires special modifications to the iSCSI initiator. No modifications are required to the iSCSI target.

As a SOCKS server can map most of the addresses and information contained within the IP and TCP headers, including sequence numbers, its effects on iSCSI are identical to those in the port redirector.

B.3. SCSI gateway

This gateway presents logical targets (iSCSI Names) to the initiators, and maps them to SCSI targets as it chooses. The initiator sees this gateway as a real iSCSI target, and is unaware of any proxy or gateway behavior. The gateway may manufacture its own iSCSI Names, or map the iSCSI names using information provided by the physical SCSI devices. It is the responsibility of the gateway to

ensure the uniqueness of any iSCSI name it manufactures. The gateway may have to account for multiple gateways having access to a single physical device. This type of gateway is used to present parallel SCSI, Fibre Channel, SSA, or other devices as iSCSI devices.

Effects on iSCSI:

- Since the initiator is unaware of any addresses beyond the gateway, the gateway's own address is for all practical purposes the real address of a target. Only the iSCSI Name needs to be passed. This is already done in iSCSI, so there are no further requirements to support SCSI gateways.

B.4. iSCSI Proxy

An iSCSI proxy is a gateway that terminates the iSCSI protocol on both sides, rather than translate between iSCSI and some other transport. The proxy functionality is aware that both sides are iSCSI, and can take advantage of optimizations, such as the preservation of data integrity checks. Since an iSCSI initiator's discovery or configuration of a set of targets makes use of address-independent iSCSI names, iSCSI does not have the same proxy addressing problems as HTTP, which includes address information into its URLs. If a proxy is to provide services to an initiator on behalf of a target, the proxy allows the initiator to discover its address for the target, and the actual target device is discovered only by the proxy. Neither the initiator nor the iSCSI protocol needs to be aware of the existence of the proxy. Note that a SCSI gateway may also provide iSCSI proxy functionality when mapping targets between two iSCSI interfaces.

Effects on iSCSI:

- Same as a SCSI gateway. The only other effect is that iSCSI must separate data integrity checking on iSCSI headers and iSCSI data, to allow the data integrity check on the data to be propagated end-to-end through the proxy.

B.5. Stateful Inspection Firewall (stealth iSCSI firewall)

The stealth model would exist as an iSCSI-aware firewall, that is invisible to the initiator, but provides capabilities found in the iSCSI proxy.

Effects on iSCSI:

- Since this is invisible, there are no additional requirements on the iSCSI protocol for this one.

This one is more difficult in some ways to implement, simply because it has to be part of a standard firewall product, rather than part of an iSCSI-type product.

Also note that this type of firewall is only effective in the outbound direction (allowing an initiator behind the firewall to connect to an outside target), unless the iSCSI target is located in a DMZ (De-Militarized Zone) [RFC3303]. It does not provide adequate security otherwise.

Appendix C: iSCSI Names and Security Identifiers

This document has described the creation and use of iSCSI Node Names. There will be trusted environments where this is a sufficient form of identification. In these environments the iSCSI Target may have an Access Control List (ACL), which will contain a list of authorized entities that are permitted to access a restricted resource (in this case a Target Storage Controller). The iSCSI Target will then use that ACL to permit (or not) certain iSCSI Initiators to access the storage at the iSCSI Target Node. This form of ACL is used to prevent trusted initiators from making a mistake and connecting to the wrong storage controller.

It is also possible that the ACL and the iSCSI Initiator Node Name can be used in conjunction with the SCSI layer for the appropriate SCSI association of LUNs with the Initiator. The SCSI layer’s use of the ACL will not be discussed further in this document.

There will be situations where the iSCSI Nodes exist in untrusted environments. That is, some iSCSI Initiator Nodes may be authorized to access an iSCSI Target Node, however, because of the untrusted environment, nodes on the network cannot be trusted to give the correct iSCSI Initiator Node Names.

In untrusted environments an additional type of identification is required to assure the target that it really knows the identity of the requesting entity.

The authentication and authorization in the iSCSI layer is independent of anything that IPSec might handle, underneath or around the TCP layer. This means that the initiator node needs to pass some type of security related identification information (e.g., userid) to a security authentication process such as SRP, CHAP, Kerberos etc. (These authentication processes will not be discussed in this document.)

Upon the completion of the iSCSI security authentication, the installation knows "who" sent the request for access. The installation must then check to ensure that such a request, from the identified entity, is permitted/authorized. This form of Authorization is generally accomplished via an Access Control List (ACL) as described above. Using this authorization process, the iSCSI target will know that the entity is authorized to access the iSCSI Target Node.

It may be possible for an installation to set a rule that the security identification information (e.g., UserID) be equal to the iSCSI Initiator Node Name. In that case, the ACL approach described above should be all the authorization that is needed.

If, however, the iSCSI Initiator Node Name is not used as the security identifier there is a need for more elaborate ACL functionality. This means that the target requires a mechanism to map the security identifier (e.g., UserID) information to the iSCSI Initiator Node Name. That is, the target must be sure that the entity requesting access is authorized to use the name, which was specified with the Login Keyword "InitiatorName=". For example, if security identifier 'Frank' is authorized to access the target via iSCSI InitiatorName=xxxx, but 'Frank' tries to access the target via iSCSI InitiatorName=yyyy, then this login should be rejected.

On the other hand, it is possible that 'Frank' is a roaming user (or a Storage Administrator) that "owns" several different systems, and thus, could be authorized to access the target via multiple different iSCSI initiators. In this case, the ACL needs to have the names of all the initiators through which 'Frank' can access the target.

There may be other more elaborate ACL approaches, which can also be deployed to provide the installation/user with even more security with flexibility.

The above discussion is trying to inform the reader that, not only is there a need for access control dealing with iSCSI Initiator Node Names, but in certain iSCSI environments there might also be a need for other complementary security identifiers.

Authors' Addresses

Kaladhar Voruganti
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120

EMail: kaladhar@us.ibm.com

Mark Bakke
Cisco Systems, Inc.
6450 Wedgwood Road
Maple Grove, MN 55311

Phone: +1 763 398-1054
EMail: mbakke@cisco.com

Jim Hafner
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120

Phone: +1 408 927-1892
EMail: hafner@almaden.ibm.com

John L. Hufferd
IBM Storage Systems Group
5600 Cottle Road
San Jose, CA 95193

Phone: +1 408 256-0403
EMail: hufferd@us.ibm.com

Marjorie Krueger
Hewlett-Packard Corporation
8000 Foothills Blvd
Roseville, CA 95747-5668, USA

Phone: +1 916 785-2656
EMail: marjorie_krueger@hp.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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