Internet Draft
<draft-ietf-ips-iscsi-slp-09.txt>
Expires February 2005

Mark Bakke
Cisco

John Hufferd
Kaladhar Voruganti
IBM

Marjorie Krueger
HP

Todd Sperry
Adaptec

August 2004

**Finding Internet Small Computer Systems Interface (iSCSI) Targets** and Name Servers using Service Location Protocol version 2 (SLPv2)

Status of this Memo



Copyright Notice

Abstract

The iSCSI protocol provides a way for hosts to access SCSI devices over an IP network. This document defines the use of the Service Location Protocol (SLP) by iSCSI hosts, devices, and management services, along with the SLP service type templates that describe the services they provide.

Acknowledgements

This draft was produced by the iSCSI Naming and Discovery team, including Joe Czap, Jim Hafner, John Hufferd, and Kaladhar Voruganti (IBM), Howard Hall (Pirus), Jack Harwood (EMC), Yaron Klein (Sanrad), Marjorie Krueger (HP), Lawrence Lamers (San Valley), Todd Sperry (Adaptec), and Joshua Tseng (Nishan). Thanks also to Julian Satran (IBM) for suggesting the use of SLP for iSCSI discovery, and to Matt Peterson (Caldera) and James Kempf (Sun) for reviewing the document from an SLP perspective.

Table of Contents



## 1. Introduction

iSCSI [RFC3720] is a protocol used to transport SCSI [SAM2] commands, data, and status across an IP network. This protocol is connection-oriented, and is currently defined over TCP. iSCSI uses a client-server relationship. The client end of the connection is an initiator, and sends SCSI commands; the server end of the connection is called a target, and receives and executes the commands.

There are several methods an iSCSI initiator can use to find the targets to which it should connect. Two of these methods can be accomplished without the use of SLP:

- Each target and its address can be statically configured on the initiator.

- Each address providing targets can be configured on the initiator; iSCSI provides a mechanism by which the initiator can query the address for a list of targets.

The above methods are further defined in "iSCSI Naming and Discovery Requirements" [RFC3721].

Each of the above methods requires a small amount of configuration to be done on each initiator. The ability to discover targets and name services without having to configure initiators is a desirable feature. The Service Location Protocol (SLP) [RFC2608] is an IETF standards track protocol that provides several features that will simplify locating iSCSI services. This document describes how SLP can be used in iSCSI environments to discover targets, addresses providing targets, and storage management servers.

## 2. Notation Conventions

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 [RFC2119].

## 3. Terminology

Here are some definitions that may aid readers that are unfamiliar with either SLP, SCSI, or iSCSI. Some of these definitions have been reproduced from [RFC2608] and "Finding an RSIP Server with SLP" [RFC3105].

| | |
|---|---|
| User Agent (UA) | A process working on the client's behalf to establish contact with some service. The UA retrieves service information from the Service Agents or Directory Agents. |
| Service Agent (SA) | A process working on behalf of one or more services to advertise the services and their capabilities. |
| Directory Agent (DA) | A process which collects service advertisements. There can only be one DA present per given host. |

| | |
|---|---|
| Scope | A named set of services, typically making up a logical administrative group. |
| Service Advertisement | A URL, attributes, and a lifetime (indicating how long the advertisement is valid), providing service access information and capabilities description for a particular service. |
| Initiator | A logical entity, typically within a host, that sends SCSI commands to targets to be executed. An initiator is usually present in the form of a device driver. |
| Target | A logical entity, typically within a storage controller or gateway, that receives SCSI commands from an initiator and executes them. A target includes one or more Logical Units (LUs); each LU is a SCSI device, such as a disk or tape drive. |
| iSCSI Name | A UTF-8 character string which serves as a unique identifier for iSCSI initiators and targets. Its format and usage is further defined in [RFC3721]. |
| iSCSI Client | A logical entity, typically a host, which includes at least one iSCSI Initiator. |
| iSCSI Server | A logical entity, typically a storage controller or gateway, which includes at least one iSCSI Target. |
| Storage Management Server | An addressable entity that provides management services that benefit an iSCSI environment. "Storage management server" is used as a generic term, rather than a specific protocol or service. |

## 4. Using SLP for iSCSI Service Discovery

Two entities are involved in iSCSI discovery. The end result is that an iSCSI initiator (e.g. a host) discovers iSCSI targets, usually provided by storage controllers or gateways.

iSCSI targets are registered with SLP as a set of service URLs, one for each address on which the target may be accessed. Initiators

discover these targets using SLP service requests. Targets that do not directly support SLP, or are under the control of a management service, may be registered by a proxy service agent as part of the software providing this service.

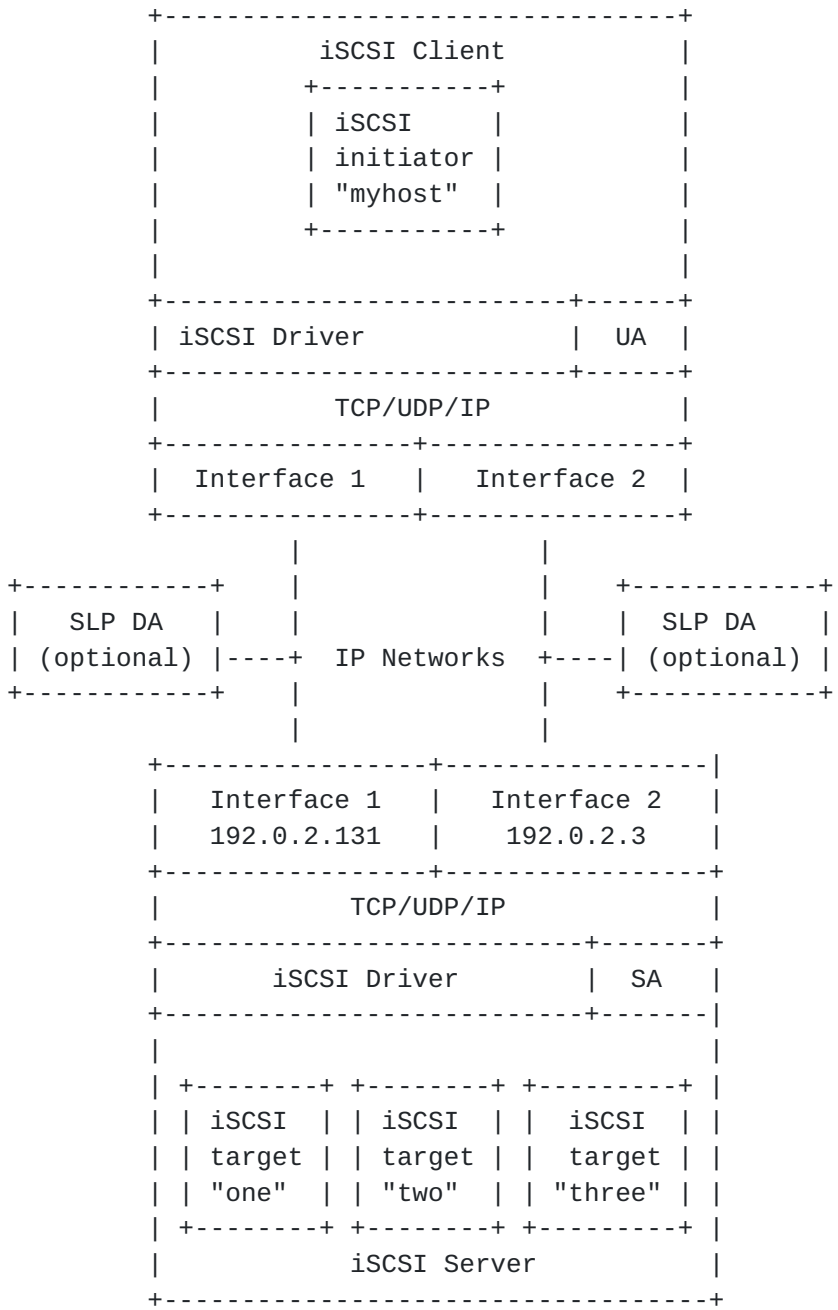
iSCSI entities may also use SLP to discover higher-level management services where needed.

This section first describes the use of SLP for discovery of targets by iSCSI initiators, and then describes the use of SLP to discover storage management servers.

This document assumes that SLPv2 will be used when discovering iSCSI-related services; no attempt is made to include support for SLPv1.

### 4.1. Discovering iSCSI Targets using SLP

The following diagram shows the relationship between iSCSI clients, servers, initiators, and targets. An iSCSI client includes at least one iSCSI initiator, and an SLP user agent (UA). An iSCSI server includes at least one iSCSI target, and an SLP service agent (SA). Some entities, such as extended copy engines, include both initiators and targets. These include both an SA, for its targets to be discovered, and a UA, for its initiator(s) to discover other targets.



In the above drawing, the iSCSI server has three iSCSI targets that the client could discover, named "one", "two" and "three". The iSCSI client has an iSCSI initiator with the name "myhost". The iSCSI client may use the initiator name in its SLP Service Requests as a filter to discover only targets that are configured to accept iSCSI connections from "myhost".

Each iSCSI target and initiator has a unique name, called an iSCSI Name. This identifier is the same regardless of the network path (through adapter cards, networks, interfaces on the storage device) over which the target is discovered and accessed. For this example, the iSCSI names "one" and "two", and "three" are used for the targets; the initiator uses the name "myhost". An actual iSCSI name would incorporate more structure, including a naming authority, and is not described here.

Each of the iSCSI targets in the drawing can appear at two addresses, since two network interfaces are present. Each target would have two service URLs, unless a single service URL included a DNS host name mapping to both addresses.

An iSCSI target URL consists of its fully qualified host name or IP address, the TCP port on which it is listening, and its iSCSI name. An iSCSI server must register each of its individual targets at each of its network addresses.

The iSCSI server constructs a service advertisement of the type "service:iscsi:target" for each of the service URLs it wishes to register. The advertisement contains a lifetime, along with other attributes which are defined in the service template.

If the server in the above drawing is listening at TCP port 3260 for both network addresses, the service URLs registered would be:

- 192.0.2.131:3260/one

- 192.0.2.131:3260/two

- 192.0.2.131:3260/three

- 192.0.2.3:3260/one

- 192.0.2.3:3260/two

- 192.0.2.3:3260/three

The remainder of the discovery procedure is identical to that used by any client/server pair implementing SLP:

1. If an SLP DA is found, the SA contacts the DA and registers the service advertisement. Whether or not one or more SLPv2 DAs are discovered, the SA maintains the advertisement itself and answers multicast UA queries directly.

2. When the iSCSI initiator requires contact information for an iSCSI target, the UA either contacts the DA using unicast or the SA using multicast. If a UA is configured with the address of the SA, it may avoid multicast and contact an SA using unicast. The UA includes a query based on the attributes to indicate the characteristics of the target(s) it requires.

3. Once the UA has the host name or address of the iSCSI server as well as the port number and iSCSI Target Name, it can begin the normal iSCSI login to the target.

As information contained in the iSCSI target template may exceed common network datagram sizes, the SLP implementation for both UAs and SAs supporting this template MUST implement SLP over TCP.

### 4.1.1. Finding Targets Based on Initiator Credentials

To be allowed access to an iSCSI target, an initiator must be authenticated. The initiator may be required by the target to produce one or more of the following credentials:

- An iSCSI Initiator Name

- An IP address

- A CHAP, SRP, or Kerberos credential

- Any combination of the above

Most iSCSI targets allow access to only one or two initiators. In the ideal discovery scenario, an initiator would send an SLP request, and receive responses ONLY for those targets to which the initiator is guaranteed a successful login. To achieve this goal, the iSCSI target template contains the following attributes, each of which allows a list of values:

1. auth-name - This attribute contains the list of initiator names allowed to access this target, or the value "any", indicating that no specific initiator name is required.

2. auth-addr - This attribute contains the list of host names and/or IP addresses which will be allowed access to this target, or the value "any", indicating that no specific address or host name is required. If a large number of addresses is to be allowed (perhaps a subnet), this attribute may contain the value "any".

3. auth-cred - This attribute contains a list of "method/identifier" credentials that will be allowed access to the target, provided they can produce the correct password or other verifier during the login process. If no specific credentials are required, the value "any" is used.

The list of valid method strings for auth-cred are defined in [RFC3720], section 11.1 "AuthMethod". The identifier used after the "/" is defined by the specific AuthMethod, also in [RFC3720]. Examples showing initiator searches based on auth-xxxx attributes are shown in the target-specific template section below.

Also note that the auth-xxxx attributes are considered to be security policy information. If these attributes are distributed, IPsec MUST be implemented as specified in the Security Implementation section below.

### 4.1.2. Supporting Access by Multiple Identities to the Same Target

If a target is to allow access to multiple host identities, more than one combination of auth-xxxx attributes will need to be allowed. In some of these cases, it is not possible to express the entire set of valid combinations of auth-xxxx attributes within a single registered service URL. For example, if a target can be addressed by:

```
auth-name=myhost1 AND auth-cred=CHAP/user1      (identity1)

   OR

auth-name-myhost2 AND auth-cred=CHAP/user2      (identity2)
```

the above cannot be specified in a single registered service URL, since (auth-name=myhost1, auth-name=myhost2, auth-cred=CHAP/user1, auth-cred=CHAP/user2) would allow either auth-name to be used with either auth-cred. This necessitates the ability to register a target and address under more than one service URL; one for (identity1) and one for (identity2).

Since service URLs must be unique, (identity1) and (identity2) must each be registered under its own unique service URL.

For systems that support the configuration of multiple identities to access a target, the service URL must contain an additional, opaque string defining the identity. This appears after the iSCSI name in the URL string, and is separated by a "/". Each registered (target-address, target-name, initiator-identity) tuple can then register its own set of auth-xxxx attributes.

### 4.1.3. Using SLP in a Non-Multicast Environment

In some networks, the use of multicast for discovery purposes is either unavailable or not allowed. Such networks include public or service-provider networks that are placed in between an iSCSI client and server; these are probably most common between two iSCSI gateways, one at a storage service provider site, and one at a customer site.

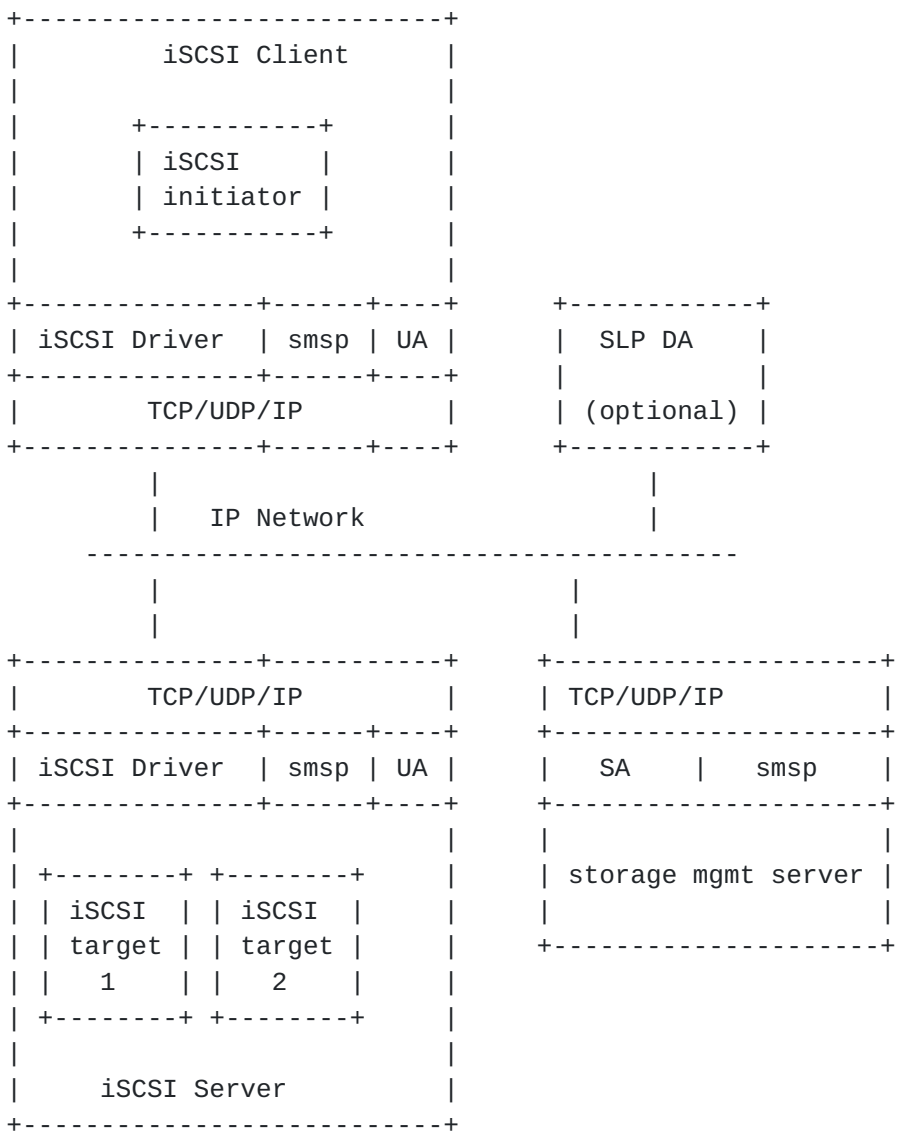
In these networks, an initiator may, instead or in addition to its DA configuration, allow the addresses of one or more SAs to be configured. The initiator would then make unicast SLP service requests directly to these SAs, without the use of multicast to first discover them.

This functionality is well within the scope of the current SLP protocol. The main consequence for implementors is that an initiator configured to make direct, unicast requests to an SA will have to add this to the SLP API, if it is following the service location API defined in [RFC2614]. This capability is being added to the next revision of the API, in [2614BIS].

## 4.2. Discovering Storage Management Services using SLP

Storage management servers can be built to manage and control access to targets in a variety of ways. They can also provide extended services beyond discovery, which could include storage allocation and management. None of these services are defined here; the intent of this document is to allow these services to be discovered by both clients and servers, in addition to the target discovery already being performed.

The following drawing shows an iSCSI client, an iSCSI server, and a storage management server. To simplify the drawing, the second IP network is not shown, but is assumed to exist. The storage management server would use its own protocol (smsp) to provide capabilities to iSCSI clients and servers; these clients and servers can both use SLP to discover the storage management server.




Note the difference between the storage management server model and the previously-defined target discovery model. When target discovery was used, the iSCSI Server implemented an SA, to be discovered by the initiator's UA. In the storage management server model, the iSCSI clients and servers both implement UAs, and the management server implements the SA.

A storage management server's URL contains the domain name or IP address and TCP or UDP port number. No other information is required.

The storage management server constructs a service advertisement of the type "service:iscsi:sms" for each of the addresses at which it appears. The advertisement contains the URL, a lifetime, along with other attributes which are defined in the service template.

The remainder of the discovery procedure is identical to that used to discover iSCSI targets, except that both initiators and targets would normally be "clients" of the storage management service.

Targets that support a storage management service implement a UA in addition to the SA. A target may alternatively just implement the UA, and allow the storage management service to advertise its targets appropriately by providing an SA and registering the appropriate service:iscsi:target registrations on the target's behalf; the target device would not have to advertise its own targets. This has no impact on the initiator.

This allows the initiators' discovery of targets to be completely interoperable regardless of which storage management service is used, or whether one is used at all, or whether the target registrations are provided directly by the target or by the management service.

### 4.3. Internationalization Considerations

SLP allows internationalized strings to be registered and retrieved. Attributes in the template that are not marked with an 'L' (literal) will be registered in a localized manner. An "en" (English) localization MUST be registered, and others MAY be registered.

Attributes that include non-ASCII characters will be encoded using UTF-8, as discussed in [RFC3722] and [RFC3491].

## 5. iSCSI SLP Templates

Three templates are provided: an iSCSI target template, a management service template, and an abstract template to encapsulate the two.

### 5.1. The iSCSI Abstract Service Type Template

This template defines the abstract service "service:iscsi". It is used as a top-level service to encapsulate all other iSCSI-related services.

Name of submitter: Mark Bakke
Language of service template: en
Security Considerations: see section 6.

Template Text:
```
------------------------template begins here----------------------
template-type=iscsi
```

```
template-version=0.1

template-description=
  This is an abstract service type.  The purpose of the iscsi
  service type is to encompass all of the services used to support
  the iSCSI protocol.

template-url-syntax=
  url-path=  ;  Depends on the concrete service type.

--------------------------template ends here-----------------------
```

## 5.2. The iSCSI Target Concrete Service Type Template

This template defines the service "service:iscsi:target". An entity containing iSCSI targets that wishes them discovered via SLP would register each of them, with each of their addresses, as this service type.

Initiators (and perhaps management services) wishing to discover targets in this way will generally use one of the following queries:

1. Find a specific target, given its iSCSI Target Name:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
Query:   (iscsi-name=iqn.2001-04.com.example.sn.456)
```

2. Find all of the iSCSI Target Names that may allow access to a given initiator:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
Query:   (auth-name=iqn.1998-03.com.example.hostid.045A7B)
```

3. Find all of the iSCSI Target Names that may allow access to any initiator:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
Query:   (auth-name=any)
```

4. Find all of the iSCSI Target Names that may allow access to this initiator, or that will allow access to any initiator:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
```

```
Query:   &(auth-name=iqn.1998-03.com.example.hostid.045A7B)
          (auth-name=any)
```

5. Find all of the iSCSI Target Names that may allow access to a given CHAP user name:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
Query:   (auth-cred=chap/my-user-name)
```

6. Find all of the iSCSI Target Names that may allow access to a given initiator that supports two IP addresses, a CHAP credential and an SRP credential, and an initiator name:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
Query:   &(|(auth-name=iqn.com.example:host47)(auth-name=any)
|(auth-addr=192.0.2.3)(auth-addr=192.0.2.131)(auth-addr=any)
|(auth-cred=chap/foo)(auth-cred=srp/my-user-name)
 (auth-cred=any))
```

7. Find the iSCSI Target Names from which the given initiator is allowed to boot:

```
Service: service:iscsi:target
Scope:   initiator-scope-list
Query:   (boot-list=iqn.1998-03.com.example.hostid.045A7B)
```

8. In addition, a management service may wish to discover all targets:

```
Service: service:iscsi:target
Scope:   management-server-scope-list
Query:   <empty-string>
```

More details on booting from an iSCSI target are defined in [BOOT].

Name of submitter: Mark Bakke
Language of service template: en
Security Considerations: see section 6.

Template Text:

```
-------------------------template begins here----------------------
template-type=iscsi:target

template-version=0.1

template-description=
```

This is a concrete service type. The iscsi:target service type is used to register individual target addresses to be discovered by others. UAs will generally search for these by including one of the following:

- the iSCSI target name
- iSCSI initiator identifiers (iSCSI name, credential, IP address)
- the service URL

```
template-url-syntax=
  url-path    = hostport "/" iscsi-name [ "/" identity ]
  hostport    = host [ ":" port ]
  host        = hostname / hostnumber  ; DNS name or IP address
  hostname    = *( domainlabel "." ) toplabel
  alphanum    = ALPHA / DIGIT
  domainlabel = alphanum / alphanum *[alphanum / "-"] alphanum
  toplabel    = ALPHA / ALPHA *[ alphanum / "-" ] alphanum
  hostnumber  = ipv4-number / ipv6-addr  ; IPv4 or IPv6 address
  ipv4-number = 1*3DIGIT 3("." 1*3DIGIT)
  ipv6-addr   = "[" ipv6-number "]"
  ipv6-number =                              6( h16 ":" ) ls32
                /                       "::" 5( h16 ":" ) ls32
                / [               h16 ] "::" 4( h16 ":" ) ls32
                / [ *1( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32
                / [ *2( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
                / [ *3( h16 ":" ) h16 ] "::"    h16 ":"   ls32
                / [ *4( h16 ":" ) h16 ] "::"              ls32
                / [ *5( h16 ":" ) h16 ] "::"              h16
                / [ *6( h16 ":" ) h16 ] "::"
  ls32        = ( h16 ":" h16 ) / ipv4-number
                ; least-significant 32 bits of ipv6 address
  h16         = 1*4HEXDIG
  port        = 1*DIGIT
  iscsi-name  = iscsi-char ; iSCSI target name
  identity    = iscsi-char ; optional identity string
  iscsi-char  = ALPHA / DIGIT / escaped / ":" / "-" / "."
                ; Intended to allow UTF-8 encoded strings
  escaped     = 1*(`' HEXDIG HEXDIG)
  ;
  ; The iscsi-name part of the URL is required and must be the iSCSI
  ; name of the target being registered.
  ; A device representing multiple targets must individually
  ; register each target/address combination with SLP.
  ; The identity part of the URL is optional, and is used to
  ; indicate an identity that is allowed to access this target.
  ;
  ; Example (split into two lines for clarity):
  ; service:iscsi:target://192.0.2.3:3260/
```

```
    ;                     iqn.2001-04.com.example.sn.45678
    ;
    ; IPv6 addresses are also supported; they use the notation specified
    ; above and in [RFC3513], section 2.2

  iscsi-name = string
  # The iSCSI Name of this target.
  # This must match the iscsi-name in the url-path.

  portal-group = integer
  # The iSCSI portal group tag for this address.  Addresses sharing
  # the same iscsi-name and portal-group tag can be used within the
  # same iSCSI session.  Portal groups are described in [RFC3720].

  transports = string M L
  tcp
    # This is a list of transport protocols that the registered
    # entity supports.  iSCSI is currently supported over TCP,
    # but it is anticipated that it could be supported over other
    # transports, such as SCTP, in the future.
  tcp

  mgmt-entity = string O
  # The fully qualified domain name, or IP address in dotted-decimal
  # notation, of the management interface of the entity containing
  # this target.
  #

  alias = string O
  # The alias string contains a descriptive name of the target.

  auth-name = string M X
  # A list of iSCSI Initiator Names that can access this target.
  # Normal iSCSI names will be 80 characters or less; max length
  # is 255.
  # Normally, only one or a few values will be in the list.
  # Using the equivalence search on this will evaluate to "true"
  # if any one of the items in this list matches the query.
  # If this list contains the default name "any", any initiator
  # is allowed to access this target, provided it matches the
  # other auth-xxx attributes.
  #
  # This attribute contains security policy information.  If this
  # attribute is distributed via an Attribute Reply message,
  # IPsec MUST be implemented.

  auth-addr = string M X
  # A list of initiator IP addresses (or host names) which will
```

```
   # be allowed access to this target.  If this list contains the
   # default name "any", any IP address is allowed access to this
   # target, provided it matches the other auth-xxx attributes.
   #
   # This attribute contains security policy information.  If this
   # attribute is distributed via an Attribute Reply message,
   # IPsec MUST be implemented.

   auth-cred = string M X
   # A list of credentials which will be allowed access to the target
   # (provided they can provide the correct password or other
   # authenticator).  Entries in this list are of the form
   # "method/identifier", where the currently defined methods are
   # "chap" and "srp", both of which take usernames as their
   # identifiers.
   #
   # This attribute contains security policy information.  If this
   # attribute is distributed via an Attribute Reply message,
   # IPsec MUST be implemented.

   boot-list = string M O
   # A list of iSCSI Initiator Names that can boot from this target.
   # This list works precisely like the auth-name attribute.  A name
   # appearing in this list must either appear in the access-list,
   # or the access-list must contain the initiator name "iscsi".
   # Otherwise, an initiator will be unable to find its boot target.
   # If boot-list contains the name "iscsi", any host can boot from it,
   # but I am not sure if this is useful to anyone.
   # If this attribute is not registered, this target is not "bootable".
   #
   # Note that the LUN the host boots from is not specified here; a
   # host will generally attempt to boot from LUN 0.
   #
   # It is quite possible that other attributes will need to be defined
   # here for booting as well.
   #
   # This attribute contains security policy information.  If this
   # attribute is distributed via an Attribute Reply message,
   # IPsec MUST be implemented.

   --------------------------template ends here-----------------------
```

### 5.3. iSCSI Storage Management Service Templates

This template defines the service "service:iscsi:sms". An entity supporting one or more iSCSI management service protocols may register itself with SLP as this service type.

iSCSI clients and servers wishing to discover storage management services using SLP will usually search for them by the protocol(s) they support:

```
     Service: service:iscsi:sms
     Scope:   initiator-scope-list
     Query:   (protocols=isns)
```

Name of submitter: Mark Bakke
Language of service template: en
Security Considerations: see section 6.

Template Text:

```
-------------------------template begins here----------------------
template-type=iscsi:sms

template-version=0.1

template-description=
  This is a concrete service type.  The iscsi:sms service type
  provides the capability for entities supporting iSCSI to discover
  appropriate management services.

template-url-syntax=
  url-path   = ; The URL of the management service [RFC2608].

protocols = string M
# The list of protocols supported by this name service.  This
# list may be expanded in the future.  There is no default.
#
# "isns"  - This management service supports the use of the iSNS
#           protocol for access management, health monitoring, and
#           discovery management services.  This protocol is defined
#           in [ISNS].
isns

transports = string M L
tcp
# This is a list of transport protocols that the registered
# entity supports.
tcp, udp

server-priority = integer
# The priority a client should give this server, when choosing
# between multiple servers with the same protocol type.
# When multiple servers are discovered for a given protocol type,
# this parameter indicates their relative precedence. Server
# precedence is protocol-specific; for some protocols, the primary
```

```
# server may have the highest server-priority value, while for
# others it may have the lowest. For example, with iSNS, the primary
# server has the lowest value (value 0).

--------------------------template ends here-----------------------
```

## 6. Security Considerations

The SLPv2 security model as specified in [RFC2608] does not provide confidentiality, but does provide an authentication mechanism for UAs to assure that service advertisements only come from trusted SAs with the exception that it does not provide a mechanism to authenticate "zero-result responses". See [RFC3723] for a discussion of the SLPv2 [RFC2608] security model.

Once a target or management server is discovered, authentication and authorization are handled by the iSCSI protocol, or by the management server's protocol. It is the responsibility of the providers of these services to ensure that an inappropriately advertised or discovered service does not compromise their security.

When no security is used for SLPv2, there is a risk of distribution of false discovery information. The primary countermeasure for this risk is authentication. When this risk is a significant concern, IPsec SAs and iSCSI in-band authentication SHOULD be used for iSCSI traffic subject to this risk to ensure that iSCSI traffic only flows between endpoints that have participated in IKE authentication and iSCSI in-band authentication. For example, if an attacker distributes discovery information falsely claiming that it is an iSCSI target, it will lack the secret information necessary to successfully complete IKE authentication or iSCSI in-band authentication, and hence will be prevented from falsely sending or receiving iSCSI traffic.

There remains a risk of a denial of service attack based on repeated use of false discovery information that will cause initiation of IKE negotiation. The countermeasures for this are administrative configuration of each iSCSI Target to limit the peers that it is willing to communicate with (i.e., by IP address range and/or DNS domain), and maintenance of a negative authentication cache to avoid repeatedly contacting an iSCSI Target that fails to authenticate. These three measures (i.e., IP address range limits, DNS domain limits, negative authentication cache) MUST be implemented.

The auth-name, auth-addr, auth-cred, and boot-list attributes comprise security policy information. When these are distributed, IPsec MUST be implemented.

### 6.1. Security Implementation

Security for SLPv2 in an IP storage environment is specified in [RFC3723].

IPsec SHOULD be implemented for SLPv2 as specified in [RFC3723]; this includes ESP with a non-null transform to provide both authentication and confidentiality.

When SLPv2 can be used to distribute auth-name, auth-addr, auth-cred, boot-list information (see Section 5.2 above), IPsec MUST be implemented, as these items are considered to be sensitive security policy information. If IPsec is not implemented, auth-name, auth-addr, auth-cred, and boot-list information MUST NOT be distributed via SLPv2, and MUST NOT be used if discovered via SLPv2.

SLPv2 authentication is OPTIONAL to implement and use, and SLPv2 authentication SHOULD be implemented when IPsec is not supported.

## 7. IANA Considerations

This document describes three SLP Templates. When they have been reviewed and approved by the IESG, they should be registered in the IANA "SVRLOC Templates" registry. This process is described in the IANA Considerations section of [RFC2609].

## 8. Summary

This document describes how SLP can be used by iSCSI initiators to find iSCSI targets and storage management servers. Service type templates for iSCSI targets and storage management servers are presented.

## 9. Normative References

[RFC2608] Guttman, E., Perkins, C., Veizades, J. and M. Day, "Service Location Protocol, version 2", RFC 2608, June 1999.

[RFC2609] Guttman, E., Perkins, C. and J. Kempf, "Service Templates and service: Schemes", RFC 2609, June 1999.

[RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names", RFC 3491, March 2003.

[RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

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

[RFC3722] Bakke, M., "String Profile for iSCSI Names", RFC 3722, March 2004.

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

## 10. Informative References

[RFC2614] Kempf, J. and E. Guttman, "An API for Service Location", RFC 2614, June 1999.

[2614BIS] Kempf, J. and E. Guttman, "An API for Service Location", draft-kempf-svrloc-rfc2614bis-00.txt, February 2002.

[SAM2] ANSI T10. "SCSI Architectural Model 2", March 2000.

[RFC3721] Bakke, M., Hafner, J., Hufferd, J., Voruganti, K., and M. Krueger, "Internet Small Computer Systems Interface (iSCSI) Naming and Discovery", RFC 3721, March 2004.

[ISNS] Tseng, J., Gibbons, K., Travostino, F., Du Laney, C. and J. Souza, "Internet Storage Name Service", Work in Progress, draft-ietf-ips-isns-22.txt, February 2004.

[BOOT] Sarkar, P., Missimer, D. and C. Sapuntzakis, "A Standard for Bootstrapping Clients using the iSCSI Protocol", Work in Progress, draft-ietf-ips-iscsi-boot-12.txt, March 2004.

[RFC3105] Kempf, J. and G. Montenegro, "Finding an RSIP Server with SLP", RFC 3105, October 2001.

## 11. Authors' Addresses

Mark Bakke
Cisco Systems, Inc.
6450 Wedgwood Road
Maple Grove, MN 55311
Voice: +1 763-398-1000
EMail: mbakke@cisco.com

Kaladhar Voruganti
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
Email: kaladhar@us.ibm.com

John L. Hufferd
IBM Storage Systems Group
5600 Cottle Road
San Jose, CA 95193
Voice: +1 408 256-0403
Email: hufferd@us.ibm.com

Marjorie Krueger
Hewlett-Packard Corporation
8000 Foothills Blvd
Roseville, CA 95747-5668, USA
Voice: +1 916 785-2656
Email: marjorie_krueger@hp.com

Todd Sperry
Adaptec, Inc.
691 South Milpitas Boulevard
Milpitas, Ca. 95035
Voice: +1 408 957-4980
Email: todd_sperry@adaptec.com

## 12. Full Copyright Notice

## Acknowledgement

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