CHAPTER 15

Configuring Solaris iSCSI Initiators (Tasks)

This chapter describes how to configure Solaris iSCSI initiators in the Solaris 10 7/05 time frame. For information on the procedures associated with configuring iSCSI initiators, see "Setting Up Solaris iSCSI Initiators (Task Map)" on page 243.

The iSCSI Technology (Overview)

iSCSI is an acronym for Internet SCSI (Small Computer System Interface), an Internet Protocol (IP)-based storage networking standard for linking data storage subsystems. This networking standard was developed by the Internet Engineering Task Force (IETF). For more information about the iSCSI technology, see RFC 3720:

```
http://www.ietf.org/rfc/rfc3720.txt
```

By carrying SCSI commands over IP networks, the iSCSI protocol enables you to access block devices from across the network as if they were connected to the local system.

If you want to use storage devices in your existing TCP/IP network, the following solutions are available:

- iSCSI block devices or tape – Translates SCSI commands and data from the block level into IP packets. The advantage of using iSCSI in your network is when you need to have block-level access between one system and the target device, such as a tape device or a database. Access to a block-level device is not locked so that you could not have multiple users or systems accessing a block-level device such as an iSCSI target device.
- NFS – Transfers file data over IP. The advantage of using NFS in your network is that you can share file data across many systems. Access to file data is locked appropriately when many users are accessing data that is available in an NFS environment.

Here are the benefits of using Solaris iSCSI initiators:

- The iSCSI protocol runs across existing Ethernet networks.
  - You can use any supported network interface card (NIC), Ethernet hub or switch.
  - One IP port can handle multiple iSCSI target devices.
  - You can use existing infrastructure and management tools for IP networks.
- There is no upper limit on the maximum number of configured iSCSI target devices.
- The protocol an be used to connect to Fibre Channel or iSCSI Storage Area Network (SAN) environments with the appropriate hardware.

Here are the current limitations or restrictions of using the Solaris iSCSI initiator software:

- No support for iSCSI devices that use SLP or iSNS is currently available.
- No boot support for iSCSI devices is currently available.
- Do not configure iSCSI targets as dump devices.
- iSCSI supports multiple connections per session, but the current Solaris implementation only supports a single connection per session.

  For more information, see RFC 3720.
- You should consider the impact of transferring large amounts of data over your existing network.

iSCSI Software and Hardware Requirements

- iSCSI target software and devices
- The Solaris 10 7/05 release, the Solaris Express 02/05, or a later release
- The following software packages:
  - `SUNWiscsiu` – Sun iSCSI Device Driver (`root`)
  - `SUNWiscsir` – Sun iSCSI Management Utilities (`usr`)

Note – The Solaris iSCSI technology includes the iSCSI initiator software only.

- Any supported NIC

Setting Up Solaris iSCSI Initiators (Task Map)

Task	Description	For Instructions
1. Identify the iSCSI software and hardware requirements.	Identify the software and hardware requirements for setting up an iSCSI-based storage network.	"iSCSI Software and Hardware Requirements" on page 242
2. Set up your iSCSI target devices.	Connect and set up your iSCSI target devices.	See your vendor's iSCSI target device documentation for setup instructions
3. (Optional) Set up authentication in your Solaris iSCSI configuration.	Decide whether you want to use authentication in your Solaris iSCSI configuration: Consider using unidirectional CHAP or bidirectional CHAP Consider using a third-party RADIUS server to simplify CHAP management	 "How to Configure CHAP Authentication for Your iSCSI Configuration" on page 246 "How to Configure RADIUS for Your iSCSI Configuration" on page 247
4. Configure the iSCSI target discovery.	Configure the iSCSI target discovery method.	"How to Prepare for a Solaris iSCSI Configuration" on page 245
5. (Optional) Remove discovered iSCSI targets .	You might need to remove a discovered iSCSI target.	"How to Remove Discovered iSCSI Targets" on page 249
6. Monitor your iSCSI configuration.	Monitor your iSCSI configuration with the `iscsiadm` command.	"Monitoring Your iSCSI Configuration" on page 250
7. (Optional) Modify your iSCSI configuration.	You might want to change your iSCSI target settings such as the header and data digest parameters.	"How to Modify iSCSI Initiator and Target Parameters" on page 252

Configuring Solaris iSCSI Initiators

Basically, the steps for configuring your Solaris iSCSI initiators involves the following steps:

- Identifying the hardware and software requirements

- Configuring your IP network
- Connecting and setting up your iSCSI target device
- (Optional) Configure iSCSI authentication between the iSCSI initiator and the iSCSI target, if necessary
- Configuring the iSCSI target discovery method
- Creating file systems on your iSCSI disks
- Monitoring your iSCSI configuration

The iSCSI configuration information is stored in the `/etc/iscsi` directory. This information requires no administration.

iSCSI Terminology

Review the following terminology before configuring iSCSI initiators.

Term	Description
Initiator	The driver that initiates SCSI requests from the iSCSI target.
Target device	Represents the iSCSI storage component.
Discovery	Discovery is the process that presents the initiator with a list of available targets
Discovery method	Describes the way in which the iSCSI targets can be found. Two discovery methods are currently available: ■ SendTargets – Potential targets are discovered by using a *discovery-address.* ■ Static – Static target address is configured.

Configuring Dynamic or Static Target Discovery

Determine whether you want to configure the dynamic iSCSI SendTargets feature or use static iSCSI initiator targets to perform device discovery.

- **Dynamic device discovery** – If an iSCSI node exposes a large number of targets, such as an iSCSI to Fibre-Channel bridge, you can supply the iSCSI node IP address/port combination and allow the iSCSI initiator to use the SendTargets features to perform the device discovery.
- **Static device discovery** – If an iSCSI node has a small number of targets or if you want to restrict the targets that the initiator attempts to access, you can statically configure the *target-name* by using the following static target address naming convention:

  target-name,*target-address*[:*port-number*]

  You can also determine the static target address from the array's management tool.

The preferred method for target discovery is SendTargets discovery.

Note – Do not configure an iSCSI target to be discovered by both static and dynamic device discovery methods. The consequence of using redundant discovery methods might be slow performance when communicating with the iSCSI target device.

▼ How to Prepare for a Solaris iSCSI Configuration

Steps

1. **Become superuser.**
2. **Verify that the iSCSI software packages are installed.**

```
# pkginfo SUNWisciu SUNWiscsir
system      SUNWiscsiu Sun iSCSI Device Driver (root)
system      SUNWiscsir Sun iSCSI Management Utilities (usr)
```

3. **Verify that you are running a Solaris release that supports the iSCSI protocol.**
   - **Solaris Express 2/05 release**

```
% cat /etc/release
                            Nevada nv_07 X86
           Copyright 2005 Sun Microsystems, Inc.  All Rights Reserved.
                        Use is subject to license terms.
                           Assembled 25 January 2005
```

   - **Solaris 10 7/05 release**

```
% cat /etc/release
                       Solaris 10 7/05 X86
           Copyright 2005 Sun Microsystems, Inc.  All Rights Reserved.
                        Use is subject to license terms.
                            Assembled 03 March 2005
```

   - **Solaris 10 release with the iSCSI patch**

     On a SPARC system:

```
# showrev -p | grep 119090
```

     On an x86 system:

```
# showrev -p | grep 119091
```

4. **Confirm that your TCP/IP network is setup.**
5. **Connect your third-party iSCSI target devices and confirm that they are configured.**

   For example, determine if the iSCSI target device is reachable by using the `telnet` command to connect to the iSCSI target device using port 3260. If the connection is refused, see "Troubleshooting iSCSI Configuration Problems" on page 254.

For information about connecting your third-party iSCSI target devices, see your third-party hardware documentation.

Configuring Authentication in Your iSCSI-Based Storage Network

Setting up authentication for your iSCSI devices is optional.

In a secure environment, authentication is not required because only trusted initiators can access the targets.

In a less secure environment, the target cannot determine if a connection request is truly from a given host. In that case, the target can authenticate an initiator by using the Challenge-Handshake Authentication Protocol (CHAP).

CHAP authentication uses the notion of a challenge and response, which means that the target challenges the initiator to prove its identity. For the challenge/response method to work, the target must know the initiator's secret key and the initiator must be set up to respond to a challenge. Refer to the array vendor's documentation for instructions on setting up the secret key on the array.

iSCSI supports unidirectional and bidirectional authentication:

- *Unidirectional authentication* enables the target to authenticate the identity of the initiator.
- *Bidirectional authentication* adds a second level of security by providing a means for the initiator to authenticate the identity of the target.

▼ How to Configure CHAP Authentication for Your iSCSI Configuration

This procedure assumes that you are logged in to the local system where you want to securely access the configured iSCSI target device.

Steps

1. **Become superuser.**
2. **Determine whether you want to configure unidirectional or bidirectional CHAP.**
   - Unidirectional authentication enables the target to validate the initiator. This method is the default method. Complete steps 3–4 only.
   - Bidirectional authentication adds a second level of security by providing a means for the initiator to authenticate the target. Complete steps 5–6 only.
3. **Unidirectional CHAP – Set the secret key on the initiator.**

For example, the following command initiates a dialogue to define the CHAP secret key.

```
# iscsiadm modify initiator-node --CHAP-Secret
```

Note – The CHAP secret length must be 16 or more characters.

---

4. **Unidirectional CHAP – Enable CHAP authentication on the initiator after the secret has been set.**

   ```
   # iscsiadm modify initiator-node --authentication CHAP
   ```

5. **Bidirectional CHAP – Set the target device secret key on the initiator.**

   For example, the following command initiates a dialogue to define the CHAP secret key.

   ```
   # iscsiadm modify target-param --CHAP-Secret eui.5000ABCD78945E2B
   ```

6. **Bidirectional CHAP – Enable bidirectional authentication parameters on the target.**

   For example:

   ```
   # iscsiadm modify target-param -B enable eui.5000ABCD78945E2B
   ```

Using a Third-Party Radius Server to Simplify CHAP Management in Your iSCSI Configuration

You can use a third-party RADIUS server to simplify CHAP secret management. A RADIUS server is a centralized authentication service. While you must still specify the initiator's CHAP secret, you are no longer required to specify each target's CHAP secret on each initiator when using bidirectional authentication with a RADIUS server.

For more information, see RFC 1994 (CHAP) and RFC 2865 (RADIUS).

▼ How to Configure RADIUS for Your iSCSI Configuration

Steps

1. **Become superuser.**

2. **Configure the initiator node with the IP address and port (the default port is 1812) of the RADIUS server.**

   For example:

   ```
   # iscsiadm modify initiator-node --radius-server 10.0.0.72:1812
   ```

3. **Configure the initiator node with the shared secret of the RADIUS server.**

   ```
   # iscsiadm modify initiator-node --radius-shared-secret
   ```

   Note – The Solaris iSCSI implementation requires that the RADIUS server is configured with a shared secret before the Solaris iSCSI software can interact with the RADIUS server.

   ---

4. **Enable the RADIUS server.**

   ```
   # iscsiadm modify initiator-node --radius-access enable
   ```

Solaris iSCSI and RADIUS Server Error Messages

This section describes the messages that are related to a Solaris iSCSI and RADIUS server configuration with potential solutions for recovery.

`empty RADIUS shared secret`
Cause: The RADIUS server is enabled on the initiator but the RADIUS shared secret is not set.

Solution: Configure the initiator with RADIUS shared secret. For more information, see "How to Configure RADIUS for Your iSCSI Configuration" on page 247.

`WARNING: RADIUS packet authentication failed`
Cause: The initiator failed to authenticate the RADIUS data packet. This error can occur if the shared secret configured on the initiator-node is different from the shared secret on the RADIUS server.

Solution: Reconfigure the initiator with the correct RADIUS shared secret. For more information, see "How to Configure RADIUS for Your iSCSI Configuration" on page 247.

▼ How to Configure iSCSI Target Discovery

This procedure assumes that you are logged in to the local system where you want to configure access to an iSCSI target device.

Steps

1. **Become superuser.**
2. **Configure the SendTargets device discovery method or the static discovery method:**
   - Configure the iSCSI device discovery method.

For example:

```
# iscsiadm add discovery-address 10.0.0.1:3260
```

The iSCSI connection is not initiated until the discovery method is enabled. See the next step.

- Configure the static discovery method.

  For example:

  ```
  # iscsiadm add static-config eui.5000ABCD78945E2B,10.0.0.1
  ```

  The iSCSI connection is not initiated until the discovery method is enabled. See the next step.

3. **Enable the iSCSI target discovery method using one of the following:**
   - If you have configured the SendTargets method of discovery, enable SendTargets discovery.

     ```
     # iscsiadm modify discovery --send-targets enable
     ```

   - If you have configured static targets, enable the static target discovery method.

     ```
     # iscsiadm modify discovery --static enable
     ```

4. **Create the iSCSI device links for the local system.**

   ```
   # devfsadm -i iscsi
   ```

▼ How to Remove Discovered iSCSI Targets

This optional procedure assumes that you are logged in to the local system where access to an iSCSI target device has already been configured.

Steps

1. **Become superuser.**
2. **(Optional) Disable an iSCSI target discovery method using one of the following:**
   - If you need to disable the SendTargets method of discovery, use the following command:

     ```
     # iscsiadm modify discovery --send-targets disable
     ```

   - If you need to disable the static targets, use the following command:

     ```
     # iscsiadm modify discovery --static disable
     ```

3. **Remove an iSCSI device discovery entry:**
   - Remove an iSCSI SendTargets discovery entry.

     For example:

     ```
     # iscsiadm remove discovery-address 10.0.0.1:3260
     ```

- Remove a static iSCSI initiator entry.

  For example:

  ```
  # iscsiadm remove static-config eui.5000ABCD78945E2B,10.0.0.1
  ```

4. **Reboot the system if you want to remove the iSCSI target device.**

   The iSCSI target device is still available until the system is rebooted.

Accessing iSCSI Disks

If you want to make the iSCSI drive available on reboot, create the file system, and add an entry to the `/etc/vfstab` file as you would with a UFS file system on a SCSI device.

After the devices have been discovered by the Solaris iSCSI initiator, the login negotiation occurs automatically. The Solaris iSCSI driver determines that the number of LUNs available and creates the device nodes. Then, the iSCSI devices can be treated as any other SCSI device.

You can view the iSCSI disks on the local system with the `format` utility.

In the following `format` output, disks 1,2, and 3 are iSCSI LUNs that are not under MPxIO control. Disks 21 and 22 are iSCSI LUNs under MPxIO control.

```
# format
AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500000e010685cf1,0
       1. c0t2d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w500000e0106e3ba1,0
       2. c3t0d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>
          /iscsi/disk@0000iqn.2001-05.com.abcstorage%3A6-8a0900-477d70401-
           b0fff044352423a2-hostname-020000,0
       3. c3t1d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>
           /iscsi/disk@0000iqn.2001-05.com.abcstorage%3A6-8a0900-3fcd70401
           -085ff04434f423a2-hostname-010000,0
.
.
.
      21. c4t60A98000686F694B2F59775733426B77d0 <ABCSTORAGE-LUN-0.2 cyl
          4606 alt 2 hd 16 sec 256>
          /scsi_vhci/ssd@g60a98000686f694b2f59775733426b77
      22. c4t60A98000686F694B2F59775733434C41d0 <ABCSTORAGE-LUN-0.2 cyl
          4606 alt 2 hd 16 sec 256>
          /scsi_vhci/ssd@g60a98000686f694b2f59775733434c41
```

▼ Monitoring Your iSCSI Configuration

You can display information about the iSCSI initiator and target devices by using the `iscsiadm list` command.

Steps

1. **Become superuser.**
2. **Display information about the iSCSI initiator.**

   For example:

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
        Login Parameters (Default/Configured):
                Header Digest: NONE/-
                Data Digest: NONE/-
        Authentication Type: NONE
        RADIUS Server: NONE
        RADIUS access: unknown
```

3. **Display information about which discovery methods are in use.**

   For example:

```
# iscsiadm list discovery
Discovery:
     Static: enabled
     Send Targets: disabled
```

Example 15–1 Listing Information About a Specific iSCSI Target

The following example shows how to list information about a specific iSCSI target.

```
# iscsiadm list target-param iqn.1992-08.com.abcstorage:sn.33592219
        Target: iqn.1992-08.com.abcstorage:sn.33592219
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                Data Sequence In Order: yes/-
                Data PDU In Order: yes/-
                Default Time To Retain: 20/-
                Default Time To Wait: 2/-
                Error Recovery Level: 0/-
                First Burst Length: 65536/-
                Immediate Data: yes/-
                Initial Ready To Transfer (R2T): yes/-
                Max Burst Length: 262144/-
                Max Outstanding R2T: 1/-
                Max Receive Data Segment Length: 65536/-
                Max Connections: 1/-
                Header Digest: NONE/-
                Data Digest: NONE/-
```

Modifying iSCSI Initiator and Target Parameters

You can modify parameters on both the iSCSI initiator and the iSCSI target device. However, the only parameters that can be modified on the iSCSI initiator are the following:

- Header digest – The value can be none, the default value, or CRC32.
- Data digest – The value can be none, the default value, or CRC32.
- Authentication and CHAP secret – For more information about setting up authentication, see "How to Configure CHAP Authentication for Your iSCSI Configuration" on page 246.

The iSCSI driver provides default values for the iSCSI initiator and iSCSI target device parameters. If you modify the parameters of the iSCSI initiator, the modified parameters are inherited by the iSCSI target device, unless the iSCSI target device is already set to a different value.

Caution – Ensure that the target software supports the parameter to be modified. Otherwise, you might be unable to log in to the iSCSI target device. See your array documentation for a list of supported parameters.

Modifying iSCSI parameters should be done when I/O between the initiator and the target is complete. The iSCSI driver reconnects the session after the changes are made with the `iscsiadm modify` command.

▼ How to Modify iSCSI Initiator and Target Parameters

The first part of this procedure illustrates how modifying parameters of the iSCSI initiator are inherited by the iSCSI target device. The second part of this procedure shows how to actually modify parameters on the iSCSI target device.

Steps

1. **Become superuser.**
2. **List the current parameters of the iSCSI initiator and target device.**
   a. **List the current parameters of the iSCSI initiator. For example:**

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
        Login Parameters (Default/Configured):
                Header Digest: NONE/-
                Data Digest: NONE/-
        Authentication Type: NONE
        RADIUS Server: NONE
        RADIUS access: unknown
```

   b. **List the current parameters of the iSCSI target device. For example:**

```
# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
        Alias: -
```

```
Bi-directional Authentication: disabled
Authentication Type: NONE
Login Parameters (Default/Configured):
        Data Sequence In Order: yes/-
        Data PDU In Order: yes/-
        Default Time To Retain: 20/-
        Default Time To Wait: 2/-
        Error Recovery Level: 0/-
        First Burst Length: 65536/-
        Immediate Data: yes/-
        Initial Ready To Transfer (R2T): yes/-
        Max Burst Length: 262144/-
        Max Outstanding R2T: 1/-
        Max Receive Data Segment Length: 65536/-
        Max Connections: 1/-
        Header Digest: NONE/-
        Data Digest: NONE/-
```

Note that both header digest and data digest parameters are currently set to NONE for both the iSCSI initiator and the iSCSI target device.

To review the default parameters of the iSCSI target device, see the `iscsiadm list target-param` output in Example 15–1.

3. **Modify the parameter of the iSCSI initiator.**

   For example, set header digest to CRC32.

   ```
   # iscsiadm modify initiator-node -h CRC32
   ```

4. **Verify that the parameter was modified.**

   a. **Display the updated parameter information for the iSCSI initiator. For example:**

   ```
   # iscsiadm list initiator-node
   Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
   Initiator node alias: zzr1200
           Login Parameters (Default/Configured):
                   Header Digest: NONE/CRC32
                   Data Digest: NONE/-
           Authentication Type: NONE
           RADIUS Server: NONE
           RADIUS access: unknown
   ```

   Note that the header digest is now set to CRC32.

   b. **Display the updated parameter information for the iSCSI target device. For example:**

   ```
   # iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
   Target: iqn.1992-08.com.abcstorage:sn.84186266
           Alias: -
           Bi-directional Authentication: disabled
           Authentication Type: NONE
           Login Parameters (Default/Configured):
                   Data Sequence In Order: yes/-
   ```

```
Data PDU In Order: yes/-
Default Time To Retain: 20/-
Default Time To Wait: 2/-
Error Recovery Level: 0/-
First Burst Length: 65536/-
Immediate Data: yes/-
Initial Ready To Transfer (R2T): yes/-
Max Burst Length: 262144/-
Max Outstanding R2T: 1/-
Max Receive Data Segment Length: 65536/-
Max Connections: 1/-
Header Digest: CRC32/-
Data Digest: NONE/-
```

Note that the header digest is now set to CRC32.

5. **Verify that the iSCSI initiator has reconnected to the iSCSI target.**

```
# iscsiadm list target -v iqn.1992-08.com.abcstorage:sn.84186266
Target: iqn.1992-08.com.abcstorage:sn.84186266
        Target Portal Group Tag: 2
        Connections: 1
                CID: 0
                  IP address (Local): nnn.nn.nn.nnn:64369
                  IP address (Peer): nnn.nn.nn.nnn:3260
                  Discovery Method: SendTargets
                  Login Parameters (Negotiated):
                        .
                        .
                        .
                        Header Digest: CRC32
                        Data Digest: NONE
```

6. **Unset an iSCSI initiator parameter or an iSCSI target device parameter.**

   You can unset a parameter by either setting it to none with the `iscsiadm modify` command. Or, you can use the `iscsiadm remove` command to reset all target properties to the default settings.

   The following example shows how to reset the header digest to none:

```
# iscsiadm modify target-param -h none iqn.1992-08.com.abcstorage:sn...
```

   For information about using the `iscsiadm remove target-param` command, see iscsiadm.1m.

Troubleshooting iSCSI Configuration Problems

The following tools are available to troubleshoot general iSCSI configuration problems:

- `snoop` – This tool has been updated to support iSCSI packets.
- `ethereal` – This freeware product is available from `http://www.ethereal.com`.

Both tools can filter iSCSI packets on port 3260.

The following sections describe various iSCSI troubleshooting and error message resolution scenarios.

No Connections to the iSCSI Target From the Local System

▼ How to Troubleshoot iSCSI Connection Problems

Steps

1. **Become superuser.**
2. **List your iSCSI target information.**

   For example:

   ```
   # iscsiadm list target
   Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df
   -zzr1200-01
               Target Portal Group Tag: default
               Connections: 0
   ```

3. **If no connections are listed in the `iscsiadm list target` output, check the `/var/adm/messages` file for possible reasons why the connection failed.**

   You can also verify whether the connection is accessible by using the `ping` command or by connecting to the storage device's iSCSI port with the `telnet` command to ensure the iSCSI service is available. The default port is 3260.

4. **If your target is not listed in the `iscsiadm list target` output, check the `/var/adm/messages` file for possible causes.**

   If you are using SendTargets as the discovery method, try listing the *discovery-address* using the `-v` option to ensure that the expected targets are visible to the host. For example:

   ```
   # iscsiadm list discovery-address -v 10.0.0.1
   Discovery Address: 10.0.0.1:3260
       Target name: eui.210000203787dfc0
           Target address:        10.0.0.1:11824
       Target name: eui.210000203787e07b
           Target address:        10.0.0.1:11824
   ```

iSCSI Device or Disk Is Not Available on the Local System

▼ How to Troubleshoot iSCSI Device or Disk Unavailability

Steps

1. **Become superuser.**
2. **Identify the LUNs that were discovered on this target during enumeration.**

   For example:

```
# iscsiadm list target -S
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
    Target Portal Group Tag: default
    Connections: 1
        LUN: 0
            Vendor:  ABCSTOR
            Product: 0010
            OS Device Name: /dev/rdsk/c3t34d0s2
```

   The `-S` option shows which LUNs where discovered on this target during enumeration. If you think a LUN should be listed but it is not, review the `/var/adm/messages` file to see if an error was reported. Check the storage device's log files for errors. Also, ensure that any storage device LUN masking is properly configured.

General iSCSI Error Messages

This section describes the iSCSI messages that might be found in the `/var/adm/messages` file and potential solutions for recovery.

The message format is as follows:

`iscsi` *TYPE* (*OID*) *STRING* (*STATUS-CLASS#/STATUS-DETAIL#*)

TYPE	Is either connection or session.
OID	Is the object ID of the connection or session. This ID is unique for an OS instance.
STRING	Is a description of the condition.
<STATUS-CLASS#>/<STATUS-DETAIL#>	These values are returned in an iSCSI login response as defined by RFC 3270.

`iscsi connection(`*OID*`) login failed - Miscellaneous iSCSI initiator errors.`

Cause: The device login failed due to some form of initiator error.

`iscsi connection(`*OID*`) login failed - Initiator could not be successfully authenticated.`

Cause: The device could not successfully authenticate the initiator.

Solution: If applicable, verify that the passwords or RADIUS information are accurate.

`iscsi connection(`*OID*`) login failed - Initiator is not allowed access to the given target.`

Cause: The device will not allow the initiator access to the iSCSI target device.

Solution: Verify your initiator name and confirm that it is properly masked or provisioned by the storage device.

`iscsi connection(`*OID*`) login failed - Requested ITN does not exist at this address.`

Cause: The device does not provide access to the iSCSI target name (ITN) that you are requesting.

Solution: Verify the initiator discovery information is entered properly and that the storage device is configured properly.

`iscsi connection(`*OID*`) login failed - Requested ITN has been removed and no forwarding address is provided.`

Cause: The device can no longer provide access to the iSCSI target name (ITN) that you are requesting.

Solution: Verify that the initiator discovery information has been specified properly and the storage device has been configured properly.

`iscsi connection(`*OID*`) login failed - Requested iSCSI version range is not supported by the target.`

Cause: The initiator's iSCSI version is not supported by the storage device.

`iscsi connection(`*OID*`) login failed - No more connections can be accepted on this Session ID (SSID).`

Cause: The storage device cannot accept another connection for this initiator node to the iSCSI target device.

`iscsi connection(`*OID*`) login failed - Missing parameters (e.g., iSCSI initiator and/or target name).`

Cause: The storage device is reporting that the initiator or target name has not been properly specified.

Solution: Properly specify the iSCSI initiator or target name.

`iscsi connection(`*OID*`) login failed - Target hardware or software error.`

Cause: The storage device encountered a hardware or software error.

Solution: Consult the storage documentation or contact the storage vendor for further assistance.

`iscsi connection(`*OID*`) login failed - iSCSI service or target is not currently operational.`

Cause: The storage device is currently not operational.

Solution: Consult the storage documentation or contact the storage vendor for further assistance.

`iscsi connection(`*OID*`) login failed - Target has insufficient session, connection or other resources.`

Cause: The storage device has insufficient resources.

Solution: Consult the storage documentation or contact the storage vendor for further assistance.

`iscsi connection(`*OID*`) login failed - unable to initialize authentication`

`iscsi connection(`*OID*`) login failed - unable to set authentication`

`iscsi connection(`*OID*`) login failed - unable to set username`

`iscsi connection(`*OID*`) login failed - unable to set password`

`iscsi connection(`*OID*`) login failed - unable to set ipsec`

`iscsi connection(`*OID*`) login failed - unable to set remote authentication`

Cause: The initiator was unable to initialize or set authentication properly.

Solution: Verify that your initiator settings for authentication are properly configured.

`iscsi connection(`*OID*`) login failed - unable to make login pdu`

Cause: The initiator was unable to make a login payload data unit (PDU) based on the initiator or storage device settings.

Solution: Try resetting any target login parameters or other nondefault settings.

`iscsi connection(`*OID*`) login failed - failed to transfer login`

`iscsi connection(`*OID*`) login failed - failed to receive login response`

Cause: The initiator failed to transfer or receive a login payload data unit (PDU) across the network connection.

Solution: Verify that the network connection is reachable.

`iscsi connection(`*OID*`) login failed - received invalid login response (`*OP CODE*`)`

Cause: The storage device has responded to a login with an unexpected response.

`iscsi connection(`*OID*`) login failed - login failed to authenticate with target`

Cause: The initiator was unable to authenticate the storage device.

Solution: Verify that your initiator settings for authentication are properly configured.

`iscsi connection(`*OID*`) login failed - initiator name is required`

Cause: An initiator name must be configured to perform all actions.

Solution: Verify that the initiator name is configured.

`iscsi connection(`*OID*`) login failed - authentication receive failed`

`iscsi connection(`*OID*`) login failed - authentication transmit failed`

Cause: The initiator was unable to transmit or receive authentication information.

Solution: Verify the network connectivity with storage device or the RADIUS server as applicable.

`iscsi connection(`*OID*`) login failed - login redirection invalid`

Cause: The storage device attempted to redirect the initiator to an invalid destination.

Solution: Consult the storage documentation or contact the storage vendor for further assistance.

`iscsi connection(`*OID*`) login failed - target protocol group tag mismatch, expected <TPGT>, received <TPGT>`

Cause: The initiator and target had a TPGT (target portal group tag) mismatch.

Solution: Verify your TPGT discovery settings on the initiator or the storage device.

`iscsi connection(`*OID*`) login failed - can't accept` *PARAMETER* `in security stage`

Cause: The device responded with an unsupported login parameter during the security phase of login.

Solution: The parameter name is noted for reference. Consult the storage documentation or contact the storage vendor for further assistance.

`iscsi connection(`*OID*`) login failed - HeaderDigest=CRC32 is required, can't accept` *VALUE*

`iscsi connection(`*OID*`) login failed - DataDigest=CRC32 is required, can't accept` *VALUE*

Cause: The initiator is only configured to accept `HeaderDigest` or `DataDigest` that is set to CRC32 for this target. The device returned the value of *VALUE*.

Solution: Verify that the initiator and device digest settings are compatible.

`iscsi connection(`*OID*`) login failed - HeaderDigest=None is required, can't accept` *VALUE*

`iscsi connection(`*OID*`) login failed - DataDigest=None is required, can't accept` *VALUE*

Cause: The initiator is only configured to accept `HeaderDigest` or `DataDigest` that is set to none for this target. The device returned the value of *VALUE*.

Solution: Verify that the initiator and device digest settings are compatible.

`iscsi connection(`*OID*`) login failed - can't accept` *PARAMETER*

Cause: The initiator does not support this parameter.

`iscsi connection(`*OID*`) login failed - can't accept MaxOutstandingR2T` *VALUE*

Cause: The initiator does not accept `MaxOutstandingR2T` of the noted *VALUE*.

`iscsi connection(`*OID*`) login failed - can't accept MaxConnections` *VALUE*

Cause: The initiator does not accept the maximum connections of the noted *VALUE*.

`iscsi connection(`*OID*`) login failed - can't accept ErrorRecoveryLevel` *VALUE*

Cause: The initiator does not accept an error recovery level of the noted *VALUE*.

`iscsi session(`*OID*`)` *NAME* `offline`

Cause: All connections for this target *NAME* have been removed or failed.

`iscsi connection(`*OID*`) failure - unable to schedule enumeration`

Cause: The initiator was unable to enumerate the LUNs on this target.

Solution: You can force LUN enumeration by running the `devfsadm -i iscsi` command. For more information, see `devfsadm`(1M).

`iscsi connection(`*OID*`) unable to connect to target` *NAME* `(errno:`*ERRNO*`)`

Cause: The initiator failed to establish a network connection.

Solution: For information about the specific *ERRNO* on the connection failure, see the `/usr/include/sys/errno.h` file.