Internet-Draft
<draft-satran-iscsi-00.txt>
Expires 14 August 2000

J. Satran
D. Smith
K. Meth
IBM
C. Sapuntzakis
Cisco Systems
M. Toledano
P. Sarkar
C. Fuente
IBM
E. Zeidner
SanGate

February 2000

SCSI/TCP (SCSI over TCP)

Status of this Memo

Table of Contents

1. Abstract

The Small Computer Systems Interface (SCSI) is a popular family of protocols for communicating with I/O devices, especially storage devices.

This memo describes a transport protocol for SCSI that operates on top of TCP.

The SCSI/TCP protocol aims to be fully compliant with the requirements laid out in the SCSI Architecture Model - 2 [SAM2] document.

2. Overview

2.1. SCSI Concepts

The endpoint of most SCSI commands is a "logical unit" (LUN). Examples of logical units include hard drives, tape drives, CD and DVD drives, even printers and processors.

A "target" is a collection of logical units and is directly addressable on the network. The target corresponds to the server in the client-server model.

An "initiator" creates and sends SCSI commands to the target. The initiator corresponds to the client in the client-server model.

A "task" is a linked set of SCSI commands. Some LUNs support multiple simultaneous tasks. The target uses a "task tag" to distinguish between simultaneous tasks. Only one command in a task can be outstanding at any given time.

A SCSI command results in a data phase and a response phase. In the data phase, information travels either from the initiator to the target, as in a WRITE command, or from target to initiator, as in a READ command. In the response phase, the target returns the final status of the operation, including any errors. A response terminates a SCSI command.

2.2. SCSI/TCP Functional Overview

Communication between initiator and target occurs over one or more TCP connections. The first TCP connection opened is designated a control connection and used for sending control messages, SCSI commands, and parameters. Additional connections may be opened for sending data from the SCSI data phases.

2.3. SCSI/TCP Login

The purpose of SCSI/TCP login is to create a connection, authenticate the parties, and authorize the initiator to send SCSI commands.

The targets listen on a well-known TCP port for incoming connections. The initiator begins the login process by connecting to that well-known TCP port.

As part of the login process, the initiator and target MAY wish to authenticate each other. This can occur in many different ways. For example, the endpoints may wish to check the IP address of the other party. If the TCP connection uses transport layer security [TLS], certificates may be used to identify the endpoints. Also, SCSI/TCP includes commands for identifying the initiator and passing an authenticator to the target (see Appendix B).

Once suitable authentication has occured, the target MAY authorize the initiator to send SCSI commands. How the target chooses to authorize an initiator is beyond the scope of this document.

The target indicates a succesful authentication and authorization by sending a login response with "accept login".

After authentication and authorization, other parameters may be negotiated using the highly extensible Text Command message that allows arbitrary key:value pairs to be passed.

Finally, if any other TCP control or data connections between the initiator and target are currently open, they will be forced closed (TCP RST), flushing unacknowledged data.

2.4. SCSI/TCP Full Feature Phase

Once the initiator is authorized to do so, the connection is in SCSI/TCP full feature phase. The initiator may send SCSI commands to the various LUNs on the target.

SCSI commands are encapsulated in messages that go over the control connection.

Data phases associated with SCSI commands go over separate data connections. Initiators may explicitly request the establishment of data connections to targets using the "Open Data Connections" message. A Target responds by granting some number of data connections, (to be established using the well known SCSI/TCP data port), and by providing a cookie for the initiator to produce upon establishment of its data connections.

The targets listen on another well-known TCP port for incoming SCSI/TCP data connections. The initiator connects to the well-known SCSI/TCP data connection port and provides the cookie it received in the "Open Data Connections" response. The cookie occupies the first 8 bytes of data sent by the initiator through the data connection. The target uses the cookie to match a newly established data channel with its corresponding control channel.

2.5. SCSI/TCP Connection Termination

Graceful connection shutdowns are done by sending TCP FINs. Graceful connection shutdowns MUST only occur when there are no outstanding tasks on the connection. A target SHOULD respond rapidly to a FIN from the initiator by closing its half of the connection.

Usually, the initiator will initiate the closing of data channels when it no longer needs them for its data transfer operations. Similarly, an initiator may initiate the closing of its control channel when it has finished all operations with the target device.

The closing of one data channel has no effect on other data channels connecting the initiator and the target.

A target may wish to close a TCP data connection. Once an initiator has received the FIN, it SHOULD not add any more data to be sent onto that connection and should close its half of the connection when it is done sending the pending data.

In the case where a control channel is closed, the target should clean up all of its state associated with the corresponding initiator; all outstanding tasks are cancelled and all resources that were allocated for the initiator can be freed. Any open data connections should be forcibly closed (using TCP RST).

2.6. Naming

Domain names, not IPv4 addresses, identify initiator and target interfaces.

In order to express an address that is to be resolved locally (without a DNS server), standard conventions are to be used. For example, a domain name of the form d.c.b.a.in-addr.arpa. might represent the IPv4 address a.b.c.d.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE