

Internet Engineering Task Force (IETF)
Request for Comments: 6688
Updates: 5663
Category: Standards Track
ISSN: 2070-1721

D. Black, Ed.
EMC Corporation
J. Glasgow
Google
S. Faibish
EMC Corporation
July 2012

Parallel NFS (pNFS) Block Disk Protection

Abstract

Parallel NFS (pNFS) extends the Network File System version 4 (NFSv4) to enable direct client access to file data on storage devices and bypass the NFSv4 server. This can increase both performance and parallelism, but it requires additional client functionality, some of which depends upon the type of storage used. The pNFS specification for block storage (RFC 5663) describes how clients can identify the volumes used for pNFS, but this mechanism requires communication with the NFSv4 server. This document updates RFC 5663 to add a mechanism that enables identification of block storage devices used by pNFS file systems without communicating with the server. This enables clients to control access to pNFS block devices when the client initially boots, as opposed to waiting until the client can communicate with the NFSv4 server.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6688>.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	4
3. GPT Partition Table Entry	4
4. Security Considerations	5
5. References	5
5.1. Normative References	5
5.2. Informative References	6
6. Acknowledgements.....	6

1. Introduction

Figure 1 shows the overall architecture of a Parallel NFS (pNFS) system:

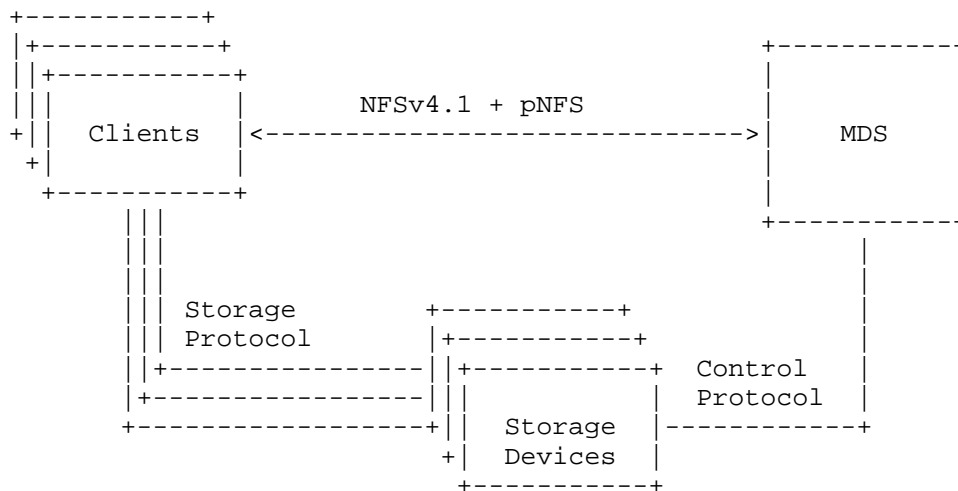


Figure 1. pNFS Architecture

In this document, "storage device" is used as a general term for a data server and/or storage server for any pNFS layout type. The MetaData Server (MDS) is the NFSv4 server that provides pNFS layouts to clients and handles operations on file metadata (e.g., names and attributes).

For the pNFS block protocol as specified in [RFC5663], client identification of pNFS storage devices requires contacting the MDS to obtain device signature information. It is not possible for a pNFS client to reliably identify pNFS block storage devices without contacting the MDS, because the device signature location and contents may vary among devices and servers; both device signature location and contents are determined by the MDS, not the client.

Typical operating system (OS) boot functionality scans and activates block devices (e.g., Small Computer System Interface (SCSI)) before activating the NFS client (including pNFS functionality). This sequence of operations creates a window of time during which the client OS may modify a pNFS block device without contacting the server (e.g., by attempting to mount or initialize a local physical filesystem). This document specifies an identification mechanism for pNFS block storage devices that can be used by an OS implementation to remove this window of vulnerability.

Many storage area network (SAN) storage systems provide quasi-static access control mechanisms (e.g., Logical Unit Number (LUN) mapping and/or masking) that operate at the granularity of individual hosts. While it is feasible to use such mechanisms to remove this window (e.g., by only enabling a client to access pNFS block storage devices after the client has contacted the responsible MDS), such usage is undesirable and potentially problematic. This is because the storage access control mechanisms are quasi-static; they are typically configured once to allow client access to the block pNFS storage devices and not reconfigured dynamically (e.g., based on crashes and reboots). Block storage access controls can be changed to respond to unusual circumstances (e.g., to fence [remove access from] an uncooperative pNFS client), but should not be used as part of routine client operations (e.g., reboot). A different mechanism is needed.

This document specifies an entry in the GUID (Globally Unique Identifier) partition table (GPT) that can be used by a pNFS server to label pNFS storage devices. This GPT entry is intended for shared pNFS storage devices that are accessible to pNFS clients and servers, and that may be accessible to other hosts or systems. This entry enables pNFS clients, as well as other hosts and systems, to avoid accessing pNFS storage devices via means other than pNFS.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. GPT Partition Table Entry

The following mechanism enables pNFS clients to identify pNFS block storage devices without contacting the server:

- Each block storage device dedicated to pNFS includes a GUID partition table (GPT) [GPT].
- The pNFS block storage partitions are identified in the GPT with GUID e5b72a69-23e5-4b4d-b176-16532674fc34, which has been generated for this purpose. GPT GUID usage is well understood and implemented. This document provides a definition for this GUID and its usage. A central registration mechanism does not exist for GPT GUIDs, or GUIDs in general, by design; see [RFC4122].

This mechanism enables an operating system to prevent non-pNFS access to pNFS block storage immediately upon boot. Servers that support pNFS block layouts SHOULD use the GPT and this GUID for all pNFS block storage devices.

A pNFS client operating system that supports block layouts SHOULD recognize this GUID and SHOULD use its presence to prevent data access to pNFS block devices until a layout that includes the device is received from the MDS.

Data stored on pNFS block layout storage devices can be better protected by incorporating checks for this GUID into other hosts and systems that do not support pNFS block layouts. If pNFS block storage devices are presented to such hosts or systems by mistake, the check for presence of this GUID can be used to prevent writes that could otherwise corrupt stored pNFS data.

Many current operating system versions support the GPT [GPT-W].

4. Security Considerations

The pNFS block layout security considerations in [RFC5663] apply to this document.

The security considerations in [RFC4122] apply to the GUID specified in this document.

5. References

5.1. Normative References

- [GPT] Unified EFI Forum, "Unified Extensible Firmware Interface Specification", Version 2.3.1, Errata A, Section 5.3, September 2011, available from <http://www.uefi.org>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5663] Black, D., Fridella, S., and J. Glasgow, "Parallel NFS (pNFS) Block/Volume Layout", RFC 5663, January 2010.

5.2. Informative References

- [GPT-W] Wikipedia, "GUID Partition Table", July 2012,
http://en.wikipedia.org/w/index.php?title=GUID_Partition_Table&oldid=502098731.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005.

6. Acknowledgements

This document was produced by the IETF NFSv4 Working Group. Review comments from members of the working group improved this document and are gratefully acknowledged. The authors would like to thank Tom Talpey, and members of the IESG for helpful comments on this document, and also Alex Burlyga for providing an appropriate reference for the format of the GPT.

Authors' Addresses

David L. Black (editor)
EMC Corporation
176 South Street
Hopkinton, MA 01748
USA
Phone: +1 (508) 293-7953
EMail: david.black@emc.com

Jason Glasgow
Google
5 Cambridge Center, Floors 3-6
Cambridge, MA 02142
USA
Phone: +1 (617) 575-1599
EMail: jglasgow@google.com

Sorin Faibish
EMC Corporation
228 South Street
Hopkinton, MA 01748
USA
Phone: +1 (508) 305-8545
EMail: sfaibish@emc.com