Proposal: Production Code Enhancements for Robust Lustre 2.x WAN Security

Steve Simms
Matt Link
Craig Stewart

Indiana University

27 June 2011

Executive summary

Indiana University has a made a significant investment in creating and maintaining a centralized data repository using the Lustre file system over the wide area network (Lustre-WAN). This has included the creation of user ID and group ID (UID and GID) mapping code that enables proper management of files by individuals and groups across different namespaces and authentication realms. This code has been contributed to Lustre 1.8x and is the basis of the Lustre-WAN service that IU has operated in production for the TeraGrid and beyond since 2008. National and international collaborators who currently use this environment appreciate the functionality of Lustre-WAN. So far, the security of IU's Lustre-WAN implementation has depended on a combination of usage over private networks, use only of non-sensitive data, careful monitoring of systems, and the lack of interest by hackers in the Lustre protocols to date. This has enabled the IU Lustre-WAN installation to operate for more than three years without a security incident, but it is not a scalable model. Kerberos security is available for Lustre-WAN and in theory scalable. In practice it is not. The lack of expertise in Kerberos on the part of potential users of Lustre-WAN, and the longstanding difficulties in implementing cross-realm Kerberos for Lustre make a Kerberos-based approach to Lustre-WAN difficult at best.

Indiana University proposes to update and improve the UID/GID mapping code for Lustre 2.x as well as to create a flexible security model that utilizes the Generic Security Services Application Program Interface (GSSAPI). This security model would not utilize Kerberos. It would allow secure client server communication as well as data encryption on the wire if desired through use of a shared secret. This method is similar to other file systems and has proven successful. It would allow widespread adoption of Lustre 2.x for Wide Area Networks, provide robust security, and would be suitable for sensitive data. (Indeed, IU intends to vet it with the relevant security authorities for transmission of electronic Private Health Information (ePHI) at IU and thus provide a precedent for other institutions.)

Indiana University proposes a cost of $250,000 for a successful implementation of the UID/GID mapping code as well as a new more flexible security model. This project will take approximately 12 months and all code produced by Indiana University will be vetted by Whamcloud Inc. under their existing support contract with IU. This important addition to the Lustre 2.x code base will enable other institutions and centers to deploy Lustre-WAN installations, thus reaping the same sort of benefits in enabling data-centric science that IU and several of its partner institutions have been able to enjoy. This should lead to more institutions adopting Lustre and Lustre-WAN, increase the size of the community of Lustre users, and aid the development of a large and vibrant community of Lustre users and service providers. This proposal, then, serves the long-term interests of research in science and technology and the long-term interests of the Lustre community and OpenSFS.

Project description

Introduction

Demand for easy-to-use, ubiquitous access to data has increased drastically in the past few years, especially from scientists and technologists. Many financial, medical, military, and research organizations have deployed (or are interested in deploying) storage solutions that permit their users to access data from many locations, all the time. Whether the distance is a matter of blocks or across continents, users would like to be able to share data easily and in some cases execute workflows that harness distributed resources.

In 2006, Indiana University deployed a site-wide Lustre file system spanning Bloomington and Indianapolis (a distance of roughly 50 miles) and began experimenting with mounting the file system across much longer distances nationally (San Diego Supercomputing Center) and abroad (Technische Universitaet Dresden, Germany). From this point forward, Indiana University has led the development of Lustre-WAN technology and pioneered its use for collaboration, remote visualization, and advanced scientific workflows.

Lustre has shown that it can provide excellent performance across the wide area network (WAN). IU's 2007 Bandwidth Challenge win was an important step towards Lustre's acceptance as a high performance wide area file system [1]. Indiana University's Lustre-WAN instance has been the one and only globally accessible file system adopted and used by a majority of the resource provider sites within the NSF-funded TeraGrid [2-7]. It has been in use in production across the TeraGrid since 2008.

A user ID (UID) mapping scheme is required for any type of WAN file system across heterogeneous namespaces to ensure that files have proper ownership and permissions across all clients. In 2007, Indiana University developed a UID mapping scheme that has enabled Lustre to be deployed across heterogeneous namespaces for versions 1.4.X – 1.8.X. There is not at present UID mapping code for Lustre 2.x. This code must be developed in order for Lustre 2.x to be used in a wide area setting.

Security is essentially risk management. So far, the security of IU's Lustre-WAN implementation has depended on a combination of usage over private networks, use only of non-sensitive data, careful monitoring of systems, and the lack of interest by hackers in the Lustre protocols to date. This has enabled the IU Lustre-WAN installation to operate for more than three years without a security incident, but it is not a scalable model. However, the current 1.8x Lustre-WAN implementation is vulnerable to a "man-in-the-middle" attack. A bad actor with appropriate knowledge of the Lustre network (LNET) could craft packets between client and server and inject them into the data stream, corrupting user data and possibly the file system. This has never yet happened, but is technically possible. This risk has prevented IU from using Lustre-WAN for sensitive data, and is perhaps the single most important factor limiting adoption of Lustre-WAN for globally accessible file systems and wide area file transfer.

Currently the only form of strong digital security implemented for Lustre is based on Kerberos. Many sites, however, don't run Kerberos and would find it impractical to stand up a Kerberos service for the sole purpose of supporting a secure Lustre-WAN installation. A set of generic security service (GSS) wrappers (that don't depend on Kerberos) for Lustre would greatly increase the number of users that could run a secured Lustre-WAN. Such wrappers would prevent man-in-the-middle attacks and provide encryption on the wire for those who need it.

Using Lustre in a heterogeneous namespace requires that client UIDs and group IDs (GIDs) be mapped to a canonical or reference set on the server side. Mapping the UIDs and GIDs is a method of accounting, to track which individual remote user owns and has access to any

particular file. Mapping provides methods of file access control, and assures that people are accessing files to which they have rights over heterogeneous namespaces. It does not, however, provide a security system per se, and certainly not something consistent with and that fulfills the functions that the Generic Security Services Application Program Interface (GSSAPI) comprises. For example, on public networks, UID mapping is subject to a man-in-the-middle attack in which an attacker crafts a packet masquerading as a mapped client, effectively permitting the same attack as someone who has gained root privileges on a client system.

Proposed work

We propose here creation of UID mapping code for Lustre 2.X and the creation of a flexible security model that uses the GSSAPI, doesn't require Kerberos, and can be easily configured to provide encryption on the wire. Such security would be sufficient to satisfy Health Information Privacy and Accountability Act (HIPAA) [8] security requirements for transmission of electronic Personal Health Information (ePHI).

Authentication of client to server and server to client can be done via shared secret. This resembles the method that IBM's General Parallel File System (GPFS) has implemented successfully. Every Lustre client has a key that is shared with every Lustre server. Messages hash the data payload, a timestamp, and the shared key to produce a signature that authenticates both the sender and integrity of the message. The timestamp is used to ensure that playback attacks against the hashing are not possible. Once the integrity of the message has been verified, UID mapping can be done on the server. This method doesn't require an additional set of servers (or employees) because no Kerberos instance is required.

Conceptually, the easiest way to protect against a network attack is to encrypt the stream. Practically, encryption is computationally expensive. Encrypting the data stream simply to authenticate the client to the server and server to the client seems unnecessary at this point and counterproductive given strongly stated user requirements for performance. However, if encryption is required, the shared keys could be used to exchange session keys to encrypt the data payload.

The proposed method doesn't require an organization to distribute usernames and passwords to non-members. In fact, this model of operation does not require file system user authentication. User authentication would not provide any additional security beyond the host/server authentication that we will provide anyway. If an attacker tries to inject a message as a user into the data stream, it will be rejected because it will be represented by a change in the hash of the message.

This method, with the possibility of long-lived persistent UID mapping, overcomes the credential limitations of a Kerberos-based user authentication system. This model easily supports batch computing on clusters with long-lived jobs and variable length queues where start times (and assigned nodes) cannot be known in advance or guaranteed.

Robust Lustre 2.x security implementation

Keys

Keys can be stored with appropriate Unix file system protections locally to each machine (as srvtabs in kerberos or PKI certificates would be). The servers will naturally have keys for all clients in memory, indexed by network ID (NID) or by universally unique ID (UUID).

Upon mounting the file system resources, the key should be loaded into kernel memory (probably, but not necessarily keychain) to keep access cost for the key low.

Rotating keys is generally thought to be a function of the time it takes to compute a key using brute force. Since key values will be (relatively) large, key rotation should not be necessary. Revoking a key due to a compromise is easy, and requires the administrator of the compromised host only to reallocate keys after the compromise is resolved (with knowledge and permission of the file system administrator).

Message authentication code

When a client wishes to send data securely to a Lustre server, this method will use the data payload, the shared secret between the server and the client, and a timestamp to produce a 256-bit message authentication code (MAC).

The server can use the MAC to authenticate the message sender. When the message is received, the server computes the MAC against the data with the same values, and checks the output against the 256-bit MAC transferred with the data. If the MAC values do not match, the message is rejected. Similarly, when a server sends data securely to a client, the server generates a MAC from the payload with the shared secret and the timestamp so that the client can validate the sender. Code already exists in the kernel to perform this operation (hmac.c) and has been vetted, so no additional cryptographic code needs to be written. Consequently, a possible course to take would be to add support for a MAC buffer to LNET. However, the creation of a GSSAPI wrapper that handles the shared key authentication would use the GSSAPI buffer that is already a part of the LNET protocol.

Any implementation of this method should allow for the feature to be selectively turned on or off for groups of clients.

UID mapping

UID mapping would exist in a form slightly different than the code that IU has implemented for 1.8.X. Native data structures in the Lustre 2.x code (such as the idmap linked list) would be used to store mappings rather than an external kernel module.

The UID mapping scheme on the server side will be required to have knowledge of client cluster configuration (i.e., what client nodes belong to which clusters). This would be necessary for any UID mapping scheme in which users have access to a head node and run jobs on compute nodes that are otherwise inaccessible.

Alternatives considered

There are two alternative courses of action that could be considered. One is use of proprietary, licensed products – e.g., Panasas or GPFS. We take it as a given that there is value in use of an open source code in terms of ability of the Lustre community control its own destiny, and to be able to see the source code of the file systems to which we are entrusting our data. In addition, Lustre has beaten GPFS in the past in head-to-head performance competitions.

The other option is Kerberos-based security. Several sites we work with have told us unequivocally that they cannot and will not deploy or participate in a Kerberos-based security system. There is significant overhead associated with running a Kerberos service, which requires more hardware and considerable expertise to run and maintain properly. In addition, sites that currently operate a Kerberos-based security environment are often reluctant to provide credentials to individuals and machines outside of their home institution, or are prohibited from doing so by institutional policy. For example, IU runs a Kerberos service, but IU does not grant trust to *any other* Kerberos domain.

In addition, one of the major uses of Lustre-WAN is wide area access to data in high performance computing (HPC) environments. In such environments, the vast majority of work is done in a batch processing environment and jobs often wait hours to days. This is plainly

inconsistent with Kerberos implementations, which depend fundamentally on a ticket life of a few hours. There does not exist today functioning code for cross-realm authentication via Kerberos, and even if robust, well-engineered code existed today it would not lead to widespread adoption of Lustre 2.x in a wide area network setting because of the strong and unavoidable limitations created by a dependence on Kerberos.

Conclusion

The security enhancements proposed here will lead to a robust security model and implementation for Lustre 2.x over a wide area network. It will be sufficiently secure that it will successfully pass the risk analysis associated with transport of ePHI (electronic Personal Health Information). These important additions to Lustre 2.x code base will enable other institutions and centers to deploy Lustre-WAN installations, thus reaping the same sort of benefits in enabling data-centric science that IU and several of its partner institutions have been able to enjoy. This should lead to more institutions adopting Lustre and Lustre-WAN and increase the size of the community of Lustre users. In a recent workshop on sustainability of open source software [9], one of the critical factors identified that helps enable sustainable, high quality open source software is a large community of users. This proposal will aid the growth of the community of Lustre users and service providers. This proposal, then, serves the long-term interests of research in science and technology and the long-term interests of the Lustre community and OpenSFS.

Budget summary

Best practices in programming, when secure applications are being created, involve at least two people programming as a team and a third party reviewing and vetting code. Table 1 below presents the elements of the proposed work package. We believe these activities can be completed within one calendar year.

Task #	Description	Task dependencies	Person-months
0	Finalize code designs	None	1
1	Implement UID mapping	Code design	5
2	Implement security model for server and client authentication	Code design	5
3	Implement data transmission encryption	Basic security model	1
4	Testing / QC / Landing	Testing and QC as code is developed – Landing once complete	5
5	Writing Documentation	Ongoing from #1-#4	2

Table 1. Work package breakdown for proposed Lustre-WAN programming activities.

Table 2 presents a budget for these tasks as a one-year (12 month) budget.

Budget item	FTE years	Total
Direct costs		
IU Staff and Benefits	1.55	$189,394
Total Direct Costs		$189,394
Indirect costs		
32% Facilities and administration costs for IU activities (based on iU policies)		$60,606
Net funding requested	1.55	$250,000

Table 2. 12 month project budget.

References

[1] Indiana University. Team Led by IU Wins Supercomputing Bandwidth Competition. 2007. Available from: http://newsinfo.iu.edu/news/page/normal/6839.html [cited 24 Jun 2011]

[2] Simms, S.C., G.G. Pike and D. Balog. Wide Area Filesystem Performance Using Lustre on theTeraGrid. In: *Proceedings of Proceedings of the TeraGrid 2007 Conference.* (Madison, WI, 2007). Available from: http://pti.iu.edu/pubs/wide-area-filesystem-performance-using-lustre-teragrid [cited 15 Dec 2009]

[3] Stewart, C.A., M. Link, D.S. McCaulay, G. Rodgers, G. Turner, D. Hancock, P. Wang, F. Saied, M. Pierce, R. Aiken, M. Mueller, M. Jurenz, M. Lieber, J. Tillotson and B. Plale. "Implementation, performance, and science results from a 30.7 TFLOPS IBM BladeCenter cluster". *Concurrency and Computation: Practice and Experience*, 22(2), 157-174. 2010. Available from: http://dx.doi.org/10.1002/cpe.1539 [cited 6 Jan 2010]

[4] Simms, S., S. Tiege, G. Pike, B. Hammond, Y. Ma, C. Westneat, L.L. Simms and D. Balog. Empowering Distributed Workflow with the Data Capacitor: Maximizing Lustre Performance Across the Wide Area Network. In: *Proceedings of Proceedings of the Workshop on Service-Oriented Computing Performance: Aspects, Issues, and Approaches.* (Monterey, CA, 2007). Available from: http://portal.acm.org/citation.cfm?id=1272465 [cited 31 Jan 2011]

[5] Walgenbach, J., S.C. Simms, J.P. Miller and K. Westneat. Enabling Lustre WAN for Production Use on the TeraGrid: A Lightweight UID Mapping Scheme. Paper presented at: *TeraGrid'10* (Pittsburgh, PA, 2010). Available from: http://dx.doi.org/10.1145/1838574.1838593 [cited 28 Feb 2011]

[6] Stewart, C.A., S. Simms, B. Plale, M. Link, D. Hancock and G. Fox. What is Cyberinfrastructure? In: *Proceedings of SIGUCCS 2010.* (Norfolk, VA, 24-27 Oct, 2010). Available from: http://portal.acm.org/citation.cfm?doid=1878335.1878347 [cited 31 Jan 2011]

[7] Simms, S. *Indiana University's Lustre WAN: The TeraGrid and Beyond.* April 2009. Presentation at Lustre User Group Meeting, Sausalito, CA.

[8] U.S. Department of Health and Human Services. Health Information Privacy. Available from: http://www.hhs.gov/ocr/privacy/ [cited 24 Jun 2011]

[9] *Cyberinfrastructure Software Sustainability and Reusability Workshop Final Report*. C.A. Stewart, G.T. Almes, D.S. McCaulay and B.C. Wheeler, eds., 2010. Available from: http://hdl.handle.net/2022/6701 and as print-on-demand from https://http://www.createspace.com/3506064 [cited 27 Jun 2011]