SANS/GIAC
Practical Assignment v1.4b

GSEC Certification

Securing an OpenMosix Cluster

P. M. Campbell
[patrici023]

October 2003

Table of Content

Table of Figures

Abstract

Clustering of computers is becoming an increasingly widely used method to maximize the use of idle cycles and relatively inexpensive hardware. As their use increases securing clusters becomes important. OpenMosix allows a number of such computers behave as a supercomputer. The Internet and networks allow such clusters to be geographically located almost anywhere. The use of open source software further increases the cost effectiveness therefore use of and perhaps geographical distribution of such clusters. Hence the focus of this paper is on Open Source clustering and Open Source security.

While securing a cluster of geographically local computers is important, the need for security is even greater if the cluster is composed of non-specialized, widely geographically distributed computers. As with all security what and how it is implemented depends on the needs of the cluster and its data. Security versus performance tradeoff decisions must be made.

Description of Clustering and OpenMosix

OpenMosix is an open source Linux kernel software package that allows you to create a computer cluster from a network of heterogeneous commodity computers. By commodity computers I mean computers made from components that are easily available or off the shelf such as Intel x86 based computers. The open source movement is essentially a philosophy for software creation, licensing and distribution in which the development process is often done by volunteers and the source code is made available for copy and modification within certain bounds laid out in an open source license.[i] The open source movement is not discussed further in this document. A computer cluster consists of two or more computers connected together and configured in such a way that they can be working together in some fashion; sharing the processing load amongst disparate machines.

Cluster communications between member nodes are carried via a networking medium. A member node or just node is the basic unit of a cluster, a single computer. The networking medium may be any medium supported by the kernel and the clustering software. Most clusters are connected using commodity Ethernet networking hardware, 100Mbps being the most cost effective for its speed.

Today computer clusters fall into two different categories: HAC high availability computing and HPC high performance computing. The most widely deployed type of cluster is HAC, these clusters are used to perform load balancing and fail over amongst their member nodes. HAC clusters allow industry to ensure 24 hour 7 day a week availability. Load balancing is the dynamic allocation or assignment of servers to service an incoming request amongst the nodes. For example a HAC web server farm with load balancing receives a request for a web application, the cluster will check for the least busy node and send the request to it. Load balancing can also be done in a round robin fashion; in that case the next node in the circular list would be sent the request. Fail over is the use of a heartbeat between multiple nodes, as soon as the heartbeat stops or is not detected for one of the nodes the others try to take over the load for the missing node.

The HPC cluster is an effort to provide high speed computing through the use of clustering software and inexpensive commodity computers. The HPC cluster is the poor mans supercomputer. In the computer industry faster is better. It is the goal of most computer manufacturers to build faster units but cost is also a factor. The fastest components and interconnection media in terms of I/O and CPU power are also the most *costly. True supercomputers are produced by only a handful of companies for specific purposes and it is very costly to buy even time on a supercomputer. Universities and research departments in certain industries are the ones who have a need for the speed of supercomputers. HPC clusters are built to provide parallel processing or SSI single system image. Parallel processing is the use of multiple nodes to simultaneously solve a problem. SSI is the outward appearance of one computer while the processing is done by two or more computers. With HPC an SSI implementation gives the appearance of a single very fast, if not super, computer.*

While some of the security needs for HAC clusters are the same as those for HPC clusters they are very different entities. This paper discusses security specifically as it pertains to OpenMosix SSI HPC clusters.

Introduction

OpenMosix implements an HPC cluster through a kernel extension this turns those inexpensive commodity computers into a supercomputer that will run most standard Linux software. OpenMosix is being used for such diverse clustering purposes as gaming, molecular modeling, and number crunching; as a server cluster for Linux terminal server diskless clients, in Universities and in industry.

The purpose of an HPC cluster is performance speed; implementing security can impact speed so the development of OpenMosix to date has not included security considerations. The need for security in general has increased over the last few years and clusters are no exception. This paper will address securing the clusters through physical means and through software installed on the cluster. It does not address the many basic methods to harden Linux computers, all OM nodes are assumed to have been hardened before being added to the cluster. The details of security configuration and implementation are also not covered. This paper is an overview of what security can be applied, to that end it considers four physical topologies used for OpenMosix cluster implementation:

- Isolated cluster, private network.
- Semi-isolated cluster, one dual homed node all other nodes on a private network.
- Un-isolated cluster on a LAN.
- Distributed cluster over Internet connections.

The next sections address security needs for each of the four topologies.

For the purposes of an OpenMosix (hereafter OM) cluster we will consider the following security subdivisions as they apply to each of the cluster topologies:

- physical access security: isolation/lockdown, physical access to the cluster nodes.
- console/logon access security: access to the cluster node(s) at the console(s).
- network/logon access security: access to the cluster node(s) over a network.
- Inter-node communications security: access to the cluster inter node communication.

Isolated Cluster

To paraphrase an adage; the only secure computer is a dead computer (one that never does anything or connects to anything). Since OpenMosix implements an SSI cluster, the cluster will appear to be one computer. So the same is true of OpenMosix security; a cluster that is physically and electronically isolated is the safest you can get.

An isolated cluster is a single private network with only OM nodes connected to the networking device. Hence physical security considerations are similar to that of any server room. It is necessary to have locked and limited key or keypad access to the room containing the cluster nodes and networking device(s). Access to the cluster room is given only to those who work with the cluster. If the OM cluster shares a room with other servers it may also be necessary to physically lock down the machines and their keyboards, floppy and CD bays.

Isolated Cluster
Private Network

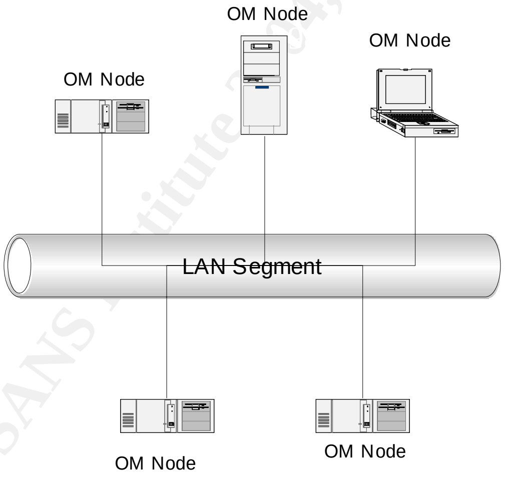


Figure 1 Sample Topology of an Isolated Cluster

It is possible to allow console access to any machine in the cluster. Since OpenMosix is an SSI cluster access to any node gives access to the whole system image; there is no node designated master. For administration purposes access to each node is sometimes necessary. So it is best to allow the administrator console access to all nodes. With access to each *node possible certain basic security should be in place such as BIOS password, boot loader / boot time password, and disable CTL-ALT-DEL keyboard reboot.*

The next barrier to be erected is logon or user authentication security. As with any UNIX computer, the use of root userid must be limited to those who need it, preferably one or two people. The current default authentication system for Linux uses one way DES (Data Encryption Standard). While this is better than two-way DES and some looser algorithms there are easily available programs that can "guess" passwords through brute force attacking. Stronger authentication is available through using PAM (pluggable authentication modules) to choose a different authentication method for example SmartCards or CryptoCards. However these solutions are more expensive and require some user training, whether or not they are to be used depends on the threat level and the organizational needs. Services such as NIS and Kerberos require servers to authenticate so are not usable on an isolated cluster.

All computers connected to the networking device must be OM nodes, that is members of the cluster, no other computers are connected. This means that network security may be lessened somewhat; all other nodes are "trusted". Unfortunately should the physical security be breached it will then be possible to attach another computer to the network or to compromise one node to gain network access to the cluster. If console access to each computer is available, through a KVM switch or physically, then all common network services can be shutdown completely to remove vulnerable points of entry. This would mean that the only traffic on the network was the OpenMosix intra-node communications.

As the network is isolated no packet filtering is necessary; packet filtering adds another layer of time to any network communications. However if console access is not available on all nodes then logon access to the other nodes must be over the private network from one node to another. To ensure secure access over the network and when security is highest priority then it is best to use ssh (secure shell) or stelnet (secure telnet) for an encrypted connection. There are several commercial and open source implementations of ssh. One of the most widely used is OpenSSH, it provides support for both SSH protocol version 1 and SSH protocol version 2 and it will interact with PAM. SSH1 supports 3DES and Blowfish encryption; SSH2 supports 3DES, Blowfish, CAST128, Arcfour and AES encryption algorithms. stelnet is not as widely used as is ssh; its encryption is done through SSLeay a free implementation of Secure Socket's layer.

Finally intra-node SSI communications are on a private network which is intended to be physically inaccessible. This should mean that traffic cannot be intercepted and decoded. Due to this consideration it may be decided to leave the intra-node SSI communications unencrypted and unauthenticated. Encrypting cluster communications adds significant communication latency and it was not considered in the implementation of OpenMosix. As the cluster is isolated, possibly in order to remove the need for such encryption, it may not be done. However it is possible to encrypt intra-node communications if the added latency is an acceptable price for security. This will be discussed in the next section.

Semi-Isolated Cluster

The isolated cluster has its drawbacks, for example the data produced have to be transported through some removable media as no other network devices have access to the cluster. Also in order to monitor the cluster itself and to run jobs on it the operator must be physically present with the cluster. This can be a cumbersome impediment to using the cluster should there be large data files to be moved. It is also possible to have a secure server room geographically far away from the users of the cluster. A semi-isolated cluster can solve these problems.

A semi-isolated cluster is a single private network with only OM nodes connected to the networking device and one OM node dual homed. A dual homed computer is one that has two network interfaces. In this case one interface is connected to the private OM network and the second network interface is connected to a non-private LAN. I will refer to the dual homed OM node as the OM bastion node, the interface connected to the private OM network as the internal interface and the one connected to the private LAN as the external interface. All OM nodes can be maintained in a server room as with an isolated cluster, so considerations for physical security and console access security remain the same as with the isolated cluster.

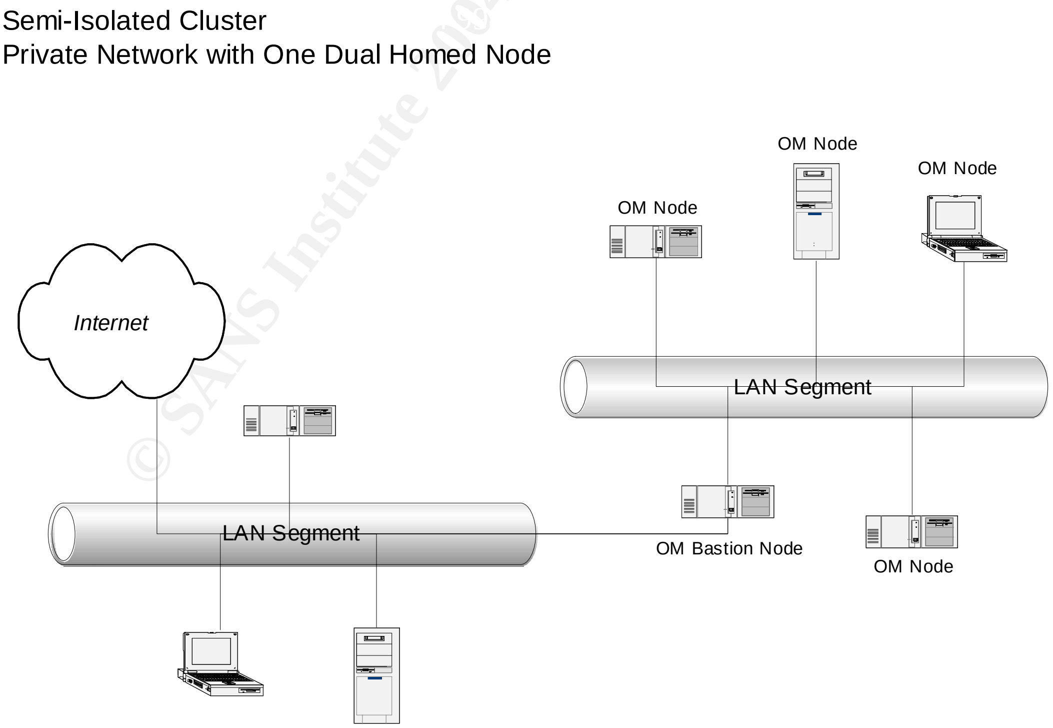

Semi-Isolated Cluster
Private Network with One Dual Homed Node

Figure 2 Sample Topology of a Semi-Isolated Cluster

The next barrier is logon or user authentication security. For the OM nodes only on the private network logon security is the same as for an isolated cluster. For the OM bastion node logon access may be its purpose but also its weakest point for security. The external interface is connected to a networking device that connects it to the non-private network hence any other device on the non private network may have access.

For the external interface the same authentication solutions may be used or a higher level of access restrictions may be implemented than that which is used for the private OM nodes. Services such as NIS+ and Kerberos may be used for user authentication to secure the OM bastion host should such servers exist on the network. NIS+ is a client server authentication service that maintains a database of user information. The + (plus) version has support for data encryption and authentication over secure RPC. Kerberos is also a client server authentication service. Once the user is authenticated to the server it provides a means to ensure identification through the issuance of tickets and encryption through the use of secret keys. The tickets and keys are like a passport and prove the identity of the client to other Kerberos enabled hosts.

It is also possible to secure the bastion with a SmartCard or a Cryptocard if that is the organization's implemented security standard. Both cards are physical devices that maintain or generate a security key to authenticate the user. They are computer chips that come imbedded in many devices. They may be used in conjunction with encryption in ssh or Kerberos and other protocols.

One more option is to use SMB (Server Message Block) for authentication. SMB is the primary protocol used for Microsoft Windows 9x and NT; there is a PAM module for SMB. Many organizations rely on Windows servers, hence this existing infrastructure can be used for authentication.

There are many methods of user authentication available for network devices. The choice of which to use is often based on the organizations standards for the whole network, so it may be necessary to implement further security to isolate the cluster.

Most common network services can be shutdown completely on all OM nodes to remove vulnerable points of entry. Those that are necessary for access to the OM bastion would be run only on the bastion. For example both sftp (secure ftp over ssh) and ssh may be necessary, perhaps even stelnet. These services can be controlled by inetd or xinetd configuration and PAM to limit times of access and even the nodes from which access is allowed. Should the majority of workstations on the non-

private network be Windows based computers then clients for ssh, sftp and stelnet are readily available, putty[ii] *is one of the easiest to use and it is open source. Though the OM bastion is a gateway of sorts forwarding should be disabled to limit access to the cluster only through authenticated access on the bastion node itself.*

The external interface is not isolated so packet filtering is necessary for that interface. Only those services, examples stated in the previous paragraph, necessary to obtain and use data should be allowed through the filter. It is possible that the OM bastion will be used as an X Client to display data, in this case all X communications should be tunneled through ssh or done over a VPN (Virtual Private Network).

Intra-node SSI communication security considerations are the same as those for an isolated cluster. However communications between the OM bastion and the non-private network may contain guarded information. Because this traffic can be intercepted and decoded perhaps encryption will be needed for that connection. If communications with the OM bastion and the non-private network are not encrypted through a higher level protocol then an encryption of all IP traffic may be necessary.

IPSec (Internet Protocol Security) is one of the more widely used cryptography systems, it provides both encryption and authentication services. IPSec runs at the network layer of the OSI Model (Open Systems Interconnect). It can protect many TCP/IP protocols running on many physical interfaces without the need to modify the protocols as it runs at the IP layer, below most protocols. The encryption and decryption are done at each end of the IPSec VPN. So each end must be using an IPSec system.

While there are a few open source IPSec packages for Linux, such as NST, USAGI and PIPSEC, the most widely used is FreeS/WAN[iii]*. The latest version of FreeS/WAN supports OE (opportunistic encryption) this removes the need to coordinate both ends of the VPN tunnel. OE allows "spontaneous" encrypted communications between two computers running IPSec supporting OE. OE relieves a lot of the administration and configuration necessary in current implementations of IPSec as OE nodes do not need to first be configured to recognize each other. Currently OE is only available in FreeS/WAN.*

It is not necessary to use OE, FreeS/WAN can encrypt data streams between itself and another FreeS/WAN, Windows 2000 PPTP, CISCO PIX firewall, or PGPnet amongst others. In order to set up these communication sessions the administrators of each VPN end point must first configure their end to recognize the other. IPSec runs in two modes *transport and tunnel. In transport mode only the data or payload of the packet is encrypted. The packet header and footer, the IP datagram portion, are left as created. In tunnel mode the whole packet is encrypted, a new IP datagram is generated to deliver the encrypted packet to the network. The overhead for tunneling is greater than that for transport but the security is also greater.*

When transport mode is used then both ends of the VPN are the actual nodes that need to communicate. In this case the OM bastion must be one end of the IPSec VPN. Since IPSec runs at the IP layer the software must be included in the kernel code. So the OpenMosix kernel will have to be compiled with the IPSec modules. This means that a compiler and the kernel code must be available to install IPSec. It is best if this kernel modification is done on a secure non OM node. Once the compile is complete the kernel binary and configuration files can be copied to the OM bastion node. Changing the kernel code requires the OpenMosix kernel source code and FreeS/WAN source code. This code should not be left on the secure OM bastion The other end of the VPN may be any node needs to communicate with the bastion and that can implement IPSec. Each end of the VPN must be configured to recognize the other end.

IPSec Communications Transport

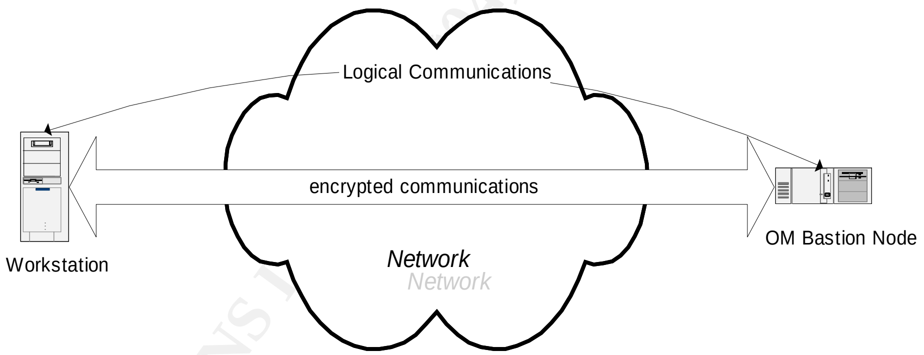


Figure 3 IPSec in Transport Mode

When tunnel mode is used both ends of the VPN act as gateways, they encrypt traffic between the two networks for which they are gateways. That is they create an essentially transparent encrypted tunnel between two disparate networks. This solution will lessen the load on the OM bastion host somewhat versus the transport mode. However since the OM bastion is not directly a part of the VPN any communications between the OM bastion and the VPN node on the local network will be on the non-private LAN. It is possible to use a VPN or ssh tunneling between the OM bastion and the local network VPN end point if that traffic absolutely must be encrypted but this is even more overhead.

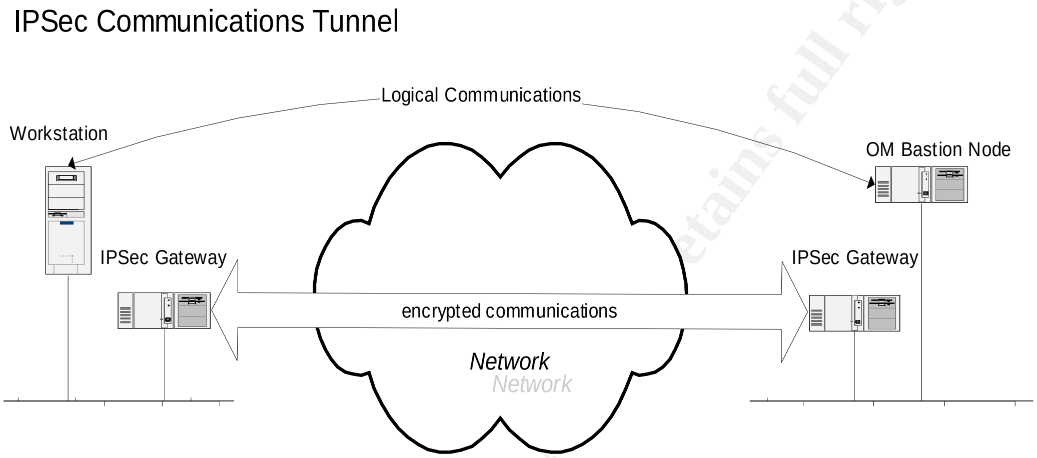


Figure 4 IPSec in Tunnel Mode

Un-Isolated Cluster

The isolated and semi-isolated clusters are only possible in those organizations that have both the hardware and the server room space available to dedicate to an HPC cluster. Their implementation also means that the hardware that is dedicated to cluster processing will be idle should the cluster be used infrequently. Some organizations may need HPC occasionally but not constantly. Some perhaps do not have the money to devote to hardware used part time. Some may wish to use the idle cycles on a computer that is used for jobs that are not CPU or I/O intensive. In those cases an un-isolated cluster is a better solution.

An un-isolated cluster is a series of OM nodes randomly distributed across an organizations network, LAN or WAN. The OM cluster shares the non-private network with other computers that may be acting as servers, desktop workstations, gateways etc.

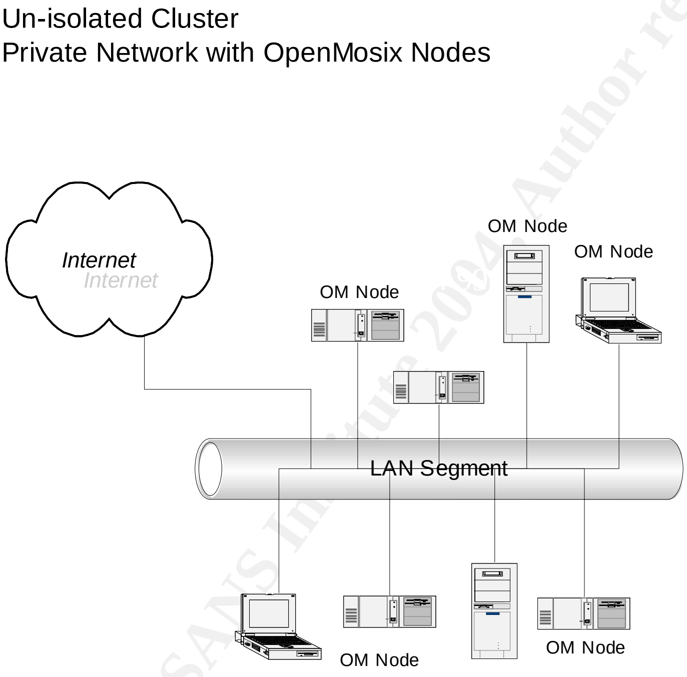


Figure 5 Sample Topology of an Un-Isolated Cluster

Since the un-isolated cluster shares a non-private network with other devices its security considerations are different to those for the isolated and semi-isolated networks. The nodes may be used for daily activities as well as for their cluster functionality. If this is the case then physical security is the equivalent to that which is used for desktop computers in any organization. They will be in offices and accessible to anyone within the office. As physical access is not really protected then any action *performed at the console will have to be limited: BIOS password, boot loader / boot time password and disable CTL-ALT-DEL keyboard reboot.*

The ease of physical access means that the next level of security chosen for console/logon will have to be the one of the most secure means detailed in the previous sections. The console/logon security options are the same as those for the OM bastion node in the semi-isolated cluster.

The network/logon security has similar options to those for the OM bastion node in the semi-isolated cluster. However any network services that are necessary for the day to day functioning of the computer must also be kept running. In this case there may be many more points of entry to the node so that tight packet filtering of all services is necessary. As with the OM bastion ssh may be necessary for secure access.

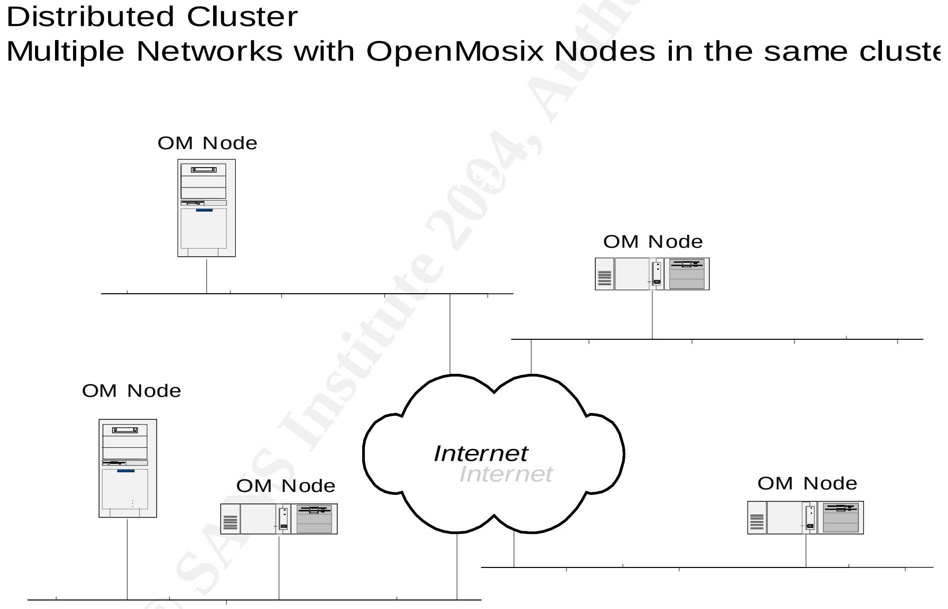
All OM nodes in an un-isolated cluster have intra-node communications over the non-private LAN. If the cluster data absolutely must be protected from any access then it is not acceptable to have clear intra-node authentication and communications. In this case all OM nodes must implement IPSec in transport mode. If OE is not used then the administration overhead is large as well as the communications overhead being large. Each OM node must have installed the OpenMosix kernel compiled with the IPSec modules. Each must also be configured to recognize all of the other OM IPSec nodes on the LAN. This is a considerable communications overhead for each of the OM nodes.

Some security that was optional for the Isolated or Semi-isolated cluster becomes obligatory for the un-isolated cluster.

Distributed Cluster

The distributed cluster is an un-isolated cluster. Where the nodes of an un-isolated cluster are spread across a LAN, those of a distributed cluster are spread across the Internet. The dynamic quality of an OM cluster allows nodes to be brought in and out of the cluster at will. It also allows processes to migrate seemingly at will. The use of the internet as the communications medium means that the number of nodes in a cluster is theoretically limitless. The current implementation of OpenMosix probably has limits; I have been unable to ascertain them at this time. The potential power of an internet wide cluster of many computers is mind boggling, even given the latency attributable to the potentially many network devices, security consideration and distances between nodes.

A distributed cluster is a series of OM nodes randomly distributed across different internet connected networks. There may be one or more OM nodes on each network.



Distributed Cluster
Multiple Networks with OpenMosix Nodes in the same cluste

Figure 6 Sample Topology of a Distributed Cluster

The OM nodes in a distributed cluster may be any internet connected computer. They could be any combination of single nodes connected to the internet, single or multiple nodes as in an un-isolated cluster on an internet connected LAN, or semi-isolated clusters whose OM bastion node is on an internet connected LAN.

The physical security of the nodes depends on their individual deployment and is more difficult to ensure consistency in the nodes than in any of the other topologies. Again the nodes may be used for daily activities as well as for their cluster functionality. Some minimum level of console/logon security would have to be agreed upon amongst the administrators of the OM nodes. Unless security is maintained at a high level a breach of any node could be catastrophic for the cluster and for the organizations at which the nodes reside.

The network/logon security has similar options to those for the OM bastion node in the semi-isolated cluster and for the nodes in an un-isolated cluster. In this case ssh or some form of public key authentication for access to the server is mandatory.

All OM nodes in a distributed cluster have intra-node communications over the Internet. Apart from any network services running on the OM nodes the intra-node cluster communications are the weakest point and absolutely must be protected from any interception. In this case all OM nodes must implement IPSec in transport mode or be behind an IPSec gateway in tunnel mode. If transport mode is used then it is the same as for an un-isolated cluster. If tunnel mode is used then all communications over the internet must be through the tunnel(s). Each tunnel must also be configured to recognize all of the other tunnels that guard the other OM nodes. This is a considerable human communications and network communications overhead. Administrators of each of the OM node containing LANs must be in communication with each other for this configuration. Depending on the number of OM nodes on un-shared networks the human communications may be the slowest. However given the easy accessibility and wide openness of the Internet it is absolutely needed to protect data integrity and node authenticity using IPSec is the best method to date. If all administrators can be made to agree to implement FreeS/WAN with OE then the administrative headache is taken away. However getting 5 or more people to agree on such an implementation may be more difficult than the configuration.

Due to the dog eat dog nature of the internet security is a major concern in a distributed cluster, perhaps more important than with any other cluster topology. Security that was optional for the isolated, semi-isolated and un-isolated cluster becomes obligatory for the un-isolated cluster. As we have less and less isolation and physical security we must implement more and more software security.

Conclusion

The isolated cluster comes first, it is the most secure and securable type of OM cluster. Its isolation is its most formidable line of defense. Second is the semi-isolated cluster this has one point of entry, the network interface. The third is the un-isolated it has the dubious security of being on a LAN which may or may not be secured properly. Finally the distributed cluster is the least secure.

As we have less and less isolation and physical security we must implement more and more software security. Note that performance decreases as security increases. With any cluster all nodes must maintain security in order to ensure that the cluster is inviolate. With an HPC SSI cluster this is especially important as the whole cluster image is available to each node. One un-secure node is a point of illicit entry.

While all types of security for a distributed cluster are required to be maintained at their highest level each of the other cluster topologies will be vulnerable if any of their barriers are breached. This would indicate that for all topologies it is best to have in place all levels of security.

As with any consideration of security its cost must be weighed against the value of that which is protected. With OM security the cost is in bandwidth and node performance (CPU and memory). It is a mistake to choose performance over security when data is extremely valuable but not so on re-creatable or less valuable data. If performance and security are tantamount and money is available then the isolated cluster is the best choice. Which security to implement must be decided on a cluster by cluster basis. For each cluster performance and security needs must be evaluated, compared and taken into consideration with the goal of the cluster.

Bibliography

Buytaert, Kris et al The OpenMosix HowTo. [Online]
<<www.openmosix.org>>

Fenzi, Kevin and Wreski, Dave Linux Security HowTo. [Online]
<<www.tldp.org>>

FreeS/WAN Project Home Page [Online] <<www.freeswan.org>>

Free OnLine Dictionary of Computing [Online] <<www.foldoc.org>>

Frisch, AEleen Essential System Administration 2nd Edition, O'Reilly and Associates, Inc., 1995. ISBN:1-56592-127-5

Jaspan, Barry Kerberos Users' Frequently Asked Questions [Online]
<<ftp://aeneas.mit.edu/pub/kerberos/doc>>

Mulas, Giacomo Turning a group of independent GNU/Linux workstations into an OpenMosix cluster in an untrusted networking environment and surviving the experience... INAF Osservatorio Astronomico di Cagliari November 2002

Nackad, Ted Securing and Optimizing Linux: The Ultimate Solution(v2.0), [downloaded from <<www.tldp.org>>] Open Network Architecture, Inc., ISBN:0-9688793-0-6

Seigfreid, Kurt Linux Administrator's Security Guide[Online].
<<seifried.org/lasg/>>

Spector, David HM Building Linux Clusters, O'Reilly and Associates, Inc., 2000. ISBN:1-56592-625-0

OpenMosix Development Web Site [Online]. <<www.openmosix.org>>

[i] *Raymond, Eric S. The Cathedral and the Bazzar*
[ii] *Tatham, Simon. PuTTY: A Free Win32 Telnet/SSH Client <<www.chiark.greenend.org.uk/~sgtatham/putty/>>*
[iii] *FreeS/WAN Project Home Page <<www.freeswan.org>>*