Demilitarizing your backups and restores.

Edmond van As (Competa IT)
Xander Soldaat (Competa IT)

Introduction

No one knows the importance of backups more than the system administrator who has felt the hot breath of his IT manager in his neck after a system crash. The importance of the availability of data has become greater than ever. This increased need for fast data backup and recovery on the corporate data centres has lead to interesting developments in the storage technology world. However, in today's Internet driven world, data is no longer confined to the relative security of a well controlled LAN, but must be made available to the world at large.
The purpose of this presentation is to give the audience a better idea of what to expect when designing a backup solution which will have to deal with piping data from a DMZ or otherwise firewalled environment to the backup network. Furthermore, we will present the results of several tests that we have performed using various firewall and backup solutions in different network layouts.

Backup methodologies

One of the scopes of this paper is to give insight in how network based backup solutions make use of your network. Although you may think that the methods used by the different software manufacturers are quite similar, in practice they tend to differ greatly.

To get a good picture of how backup products make use of TCP and/or UDP ports, we tested the following products:

- Legato NetWorker 6.1
- Arkeia Backup

The general setup for networked backups is commonly a backup server with locally attached storage devices. This host can also be a machine with enough overnight idle time to carry out this task. The existing network infrastructure is used to transfer data from the backup client hosts to the backup server host. Some companies use a separate network for backups to offload backup data from the corporate LAN.

Backups are often done by means of locally attached storage devices on every individual machine. This approach is safe in terms of separating backup data between all machines. On the other hand, the extra work involved in swapping tape cartridges and configuring and maintaining the backup software of your choice can be extremely time, and therefore money consuming.

Centralization seems to be the keyword here. Many companies have already invested in hardware such as tape robots and tape stackers. When it is possible to take up ‘DMZ’ed’ machines in an already existing and configured backup environment, why not do so? Using a backup mechanism as described above saves a lot of time in administrating only one tool instead of several. In both cases, backups could still be subject to malicious crackers that compromise the machines (and backups) from the Internet. To reduce the chance of backups being damaged or destroyed, countermeasures must be taken.

It may seem a bit paranoid to try to protect even backups made within a DMZ, but if we assume that a cracker is out to cause damage to your infrastructure, we can safely state that every chance to cause even more damage will be used.

Firewalls

As much as many vendors would like their customers to believe, firewalls are not the end solution. Firewalls are part of an overall security policy which must be enforced not just by good system administration but also user education.
There are three main types of firewalls; stateless, stateful and application proxies.

OSI Model

To fully appreciate the level at which different kind of filtering can take place, one must first look at the OSI Layer Reference Model. The OSI Reference is made up of seven layers. Here you will find an overview of their names and their function.

OSI Layer	**Description**
7 – Application	The seventh layer supports application and end-user processes. Communication partners and quality of service are identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services.
6 – Presentation	The sixth layer, sometimes also known as the syntax layer, provides data representation independence. This is done by translating the way an application represents data to something that can be used by the network and vice versa.
5 – Session	Connections between applications are provided by the fifth layer. Conversations are setup, coordinated and terminated at this level. It allows applications to exchange messages to each other.
4 – Transport	To ensure end-to-end error recovery, the fourth layer is used to provide transparent and complete transfer of data between end systems.
3 – Network	The third layer provides ways to perform circuit-switching and routing. This creates logical paths, which are also known as virtual circuits. These are used for sending data between nodes. Routing, forwarding, addressing, internetworking, error handling, congestion control and packet sequencing are done at this layer.
2 – Data Link	At the second layer, data packets are encoded and decoded into bits. It provides transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is consists of two sub layer, the first one is the Media Access Control layer (MAC) and the second one is the Logical Link Control layer (LLC). The function of the MAC layer is to control the way a computer on the network can gain access to the data and permission to send it. The LLC layer keeps track of frame synchronisation and deals with flow control and error checking.

1 – Physical	The first layer carries the bit stream - electrical impulse, light or radio signal -- through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier. It also defines cables, cards and physical aspects. Fast Ethernet, RS232, and ATM are protocols with physical layer components.

OSI Model vs. TCP/IP Model

How does the TCP/IP model fit into the OSI Reference model? For one thing, the TCP/IP model has fewer layers. This is due to the fact that some of the functionality described in the OSI model is done in one and the same TCP/IP model layer.
Layer 1 and 2 in the OSI Reference model are called the Network Access layer in the TCP/IP model. Protocols such as Ethernet, ISDN, ARP en RARP do their work in the Network Access layer.
The Internet layer is the same as layer 3 in the OSI model. This is the realm of IP. Routing between networks and host is done here. IP provides error detection, but does nothing to correct the problem. Packets that are deemed to be faulty are simply discarded by the host or router along the way.
The Host to Host Transport layer handles error correction. If packets are dropped along the way, it is up to this layer to ask the remote peer to retransmit the lost data. It is also responsible for creating and tearing down connections. TCP is the protocol used here.
The Application layer is where the higher level protocols that make use of TCP do their work. Applications communicate with each other on this layer. Examples of this would be HTTP, FTP and SMTP.

OSI Reference	**TCP/IP Model**
7 – Application	Application
6 – Presentation	
5 – Session	Host to Host Transport
4 – Transport	
3 – Network	Internet
2 – Data Link	Network Access
1 – Physical	

Filtering and Firewalling

Each layer in the TCP/IP model uses a different method for filtering unwanted traffic.

TCP/IP Model layer	Filtering method
Application	Traffic on this layer is filtered using application level proxies. It requires an intimate knowledge of the higher level protocol that is being monitored and filtered. An example of this would the Squid web proxy. Some proxies merely act as transparent tunnels for applications that wish to be able to be accessed from beyond the firewall. SOCKS proxies do this, Dante is an example.
Host to Host Transport	Firewalls are generally used on this layer. They can be both stateful and stateless, depending on the safety requirements. Traffic can be monitored and filtered on a port and IP address level. Examples of this would be Packet Filter on OpenBSD, iptables under Linux 2.4.x and extended ACLs in Cisco IOS.
Internet	Routers are most often used to filter traffic on this level. Usually at the border of a network. They monitor only access to and from certain network addresses.
Network Access	Switches reside on this level. Ethernet switches can filter based on MAC addresses, effectively locking down a port to allow traffic coming from one or a set of hosts. This document will not cover this kind of filtering.

Stateless Firewalls.

Stateless firewalls are the simplest form of firewall. They work like sieves, filtering out traffic based on ports and network addresses. They know nothing about the TCP sessions or UDP streams which pass through them. They usually only look at whether the packet is to or from a permitted network range. These firewalls often come in the shape of routers and form the first line of defence on a network. They regulate traffic that should under no circumstance reach the hinterland and allowing traffic that is deemed harmless or necessary without spending too much thought on the internal workings of the protocols it is filtering. In most cases no effort is made to filter on anything but the Network Layer (IP)

Stateful firewalls.

Stateful firewalls are more complex than their stateless counter parts and are generally considered a more desirable form of packet filtering.
To understand how a stateful firewall does its job, one needs to understand how a TCP connection is established. The connection is established using a 3-way hand shaking mechanism. When host A wishes to make connection to host B, it sends a packet with the SYN flag set in the TCP header to host B (Figure 1) together with an initial sequence number (ISN). Upon receiving the SYN packet, host B responds with a SYN packet, acknowledging (ACK) host A's ISN together with its own ISN (figure 2). Host A then responds to this packet with an ACK of host B's ISN and the first block of data. (Figure 3). This completes the three-way handshake is the beginning of the actual data stream

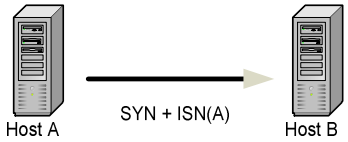


Figure 1: Initial contact by host A to host B.

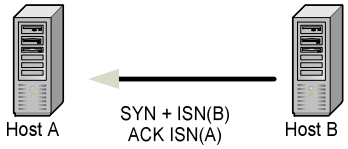


Figure 2: Host B responds to initial request

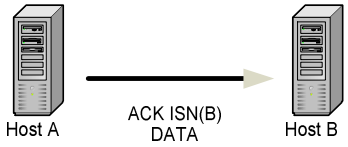


Figure 3: Final phase of handshake

So what exactly does a stateful firewall do? A stateful firewall uses a technique called connection tracking. It listens for the initial SYN packet, makes a note of it in its state tracking table and waits for the other side to respond with a SYN+ACK packet, when this is followed by the final packet handshake packet, it considers the connection as "established" and marks thus in its state table. State tracking can be done on various levels. The simplest type would be to become "transparent" after the handshake to the packets which pass between the two hosts on the pre-determined port until the connection is closed. A more sophisticated method would involve keeping track of the sequence numbers of the packets passing between the hosts. This would make certain types of attacks more difficult to realize.

Proxies

Proxies make it possible for the system administrator to protect the applications from certain kinds of content within the TCP packet. Firewalls generally do not look at what's inside these packets and are unable to know whether they pose a danger or not to the application at which they are directed. Proxies must be knowledgeable of the protocol they are to proxy. Web proxies, for example can not only analyse the HTTP commands which are sent from the client to the server, but also inspect the server's responses. This makes content-based web filtering possible.
Proxies can also check if a client is making an illegal request to the server it is trying to contact. Some server applications are sensitive to certain types of buffer overflows when the client passes data to it that it is not expecting. A proxy could intercept this and prevent possible disaster.

Test setups

The firewall used in the tests was Check Point Firewall-1 4.1.3 with DES running under NT4 SP6a on a PIII-600 with 256MB RAM. It was equipped with 3 NICs, making it possible to easily switch between various network configurations. Our main backup server was a 40MHz Sun Sparc Station LX running Solaris 8, equipped with a DLT4 drive (performance is outside the scope of this paper ☺). We ran a total of 3 scenarios using 2 different network setups.

In our first test network we configured the firewall for two Network Interface Cards (NICs). The "outside" NIC represented the DMZ, the "inside" NIC the LAN. This kind of setup works well for backup solutions in which the ability to control separate storage nodes is not required or possible. Traffic passes directly between the client on the DMZ and the backup server and can be NATd if so desired or needed.

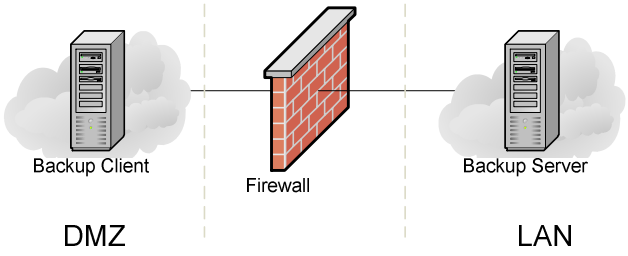


Figure 1. Test network for simple backup solutions.

The second setup involved a separate storage network on which the storage nodes reside. In theory, this should prevent direct access from the DMZ to the controlling backup server. The backup server controls the storage node and the storage node gets its data from the client.

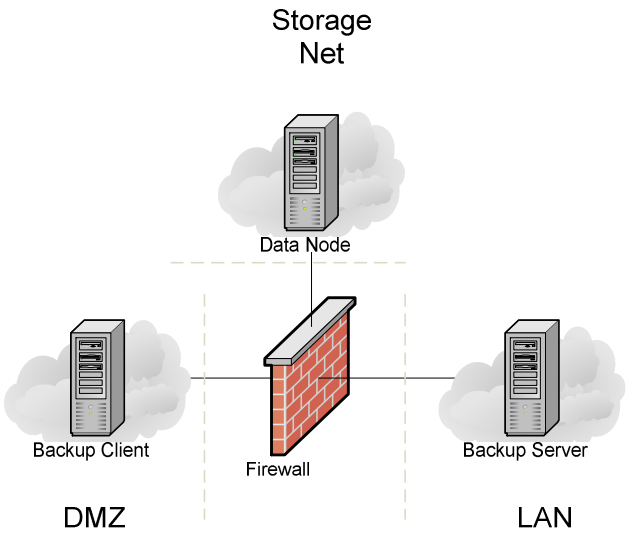
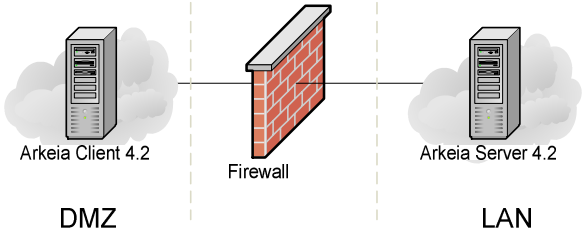


Figure 2. Test network for more complex backup solutions.

Scenario 1

In this we tested Arkeia using the simple network layout (see diagram below). The client was a PIII-733 running SuSE 7.2 with a 2.4.13 kernel. The version of Arkeia used was 4.2.

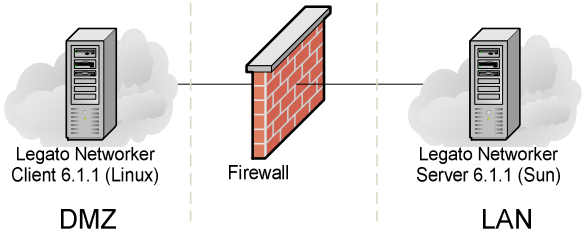


Network setup for scenario 1.

We found that Arkeia only needed one port, namely 617/TCP, in order to operate properly. Even better was the fact that this connection only needed to be made by the server to the client. This means that only one state tracking firewall rule needs to be added to the firewall rule set and that the backup server will only be exposed to traffic coming from connections it initiates itself. Please keep in mind that Arkeia will try to setup a connection on other ports than 617 when communication fails during a backup. Arkeia will try to sequentially reconnect to the client on free ports starting from port 1024.

Scenario 2

The second scenario consisted of the same network layout (see below), only this time the client was running Legato NetWorker Client version 6.1.1 and the backup server was running Legato NetWorker Server version 6.1.1.



Network setup for scenario 2.

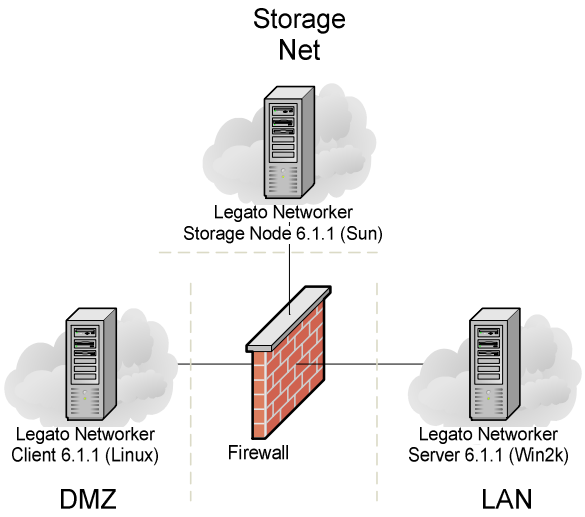
Legato uses two port ranges; service ports, communication ports. In addition to these ranges, Legato also uses two ports for nsrexecd and its own portmapper. These are all well documented on the Legato website, including instructions on how to configure these to more restricted, user defined ranges. A quick overview of the default port ranges can be found in the table below.

Range Type	Default settings
Nsrexecd	7937TCP
Legato portmapper	7938/TCP
Service ports	7939-9936/TCP
Connection ports	10001-30000/TCP

Running a backup with extensive logging enabled on the firewall showed that connections were made both from the client to the server and from the server to the client. This was also mentioned in the Legato documentation. The port ranges mentioned in the table above need to be accepted by the firewall in both directions with the exception of the service port range, which is not used by the client. This means that the server can be exposed to traffic from the client, even if it has not initiated a backup operation.

Scenario 3

The third and final scenario made use of a more complex network layout. The idea was to setup a separate network on which the data from the DMZ would be stored. This storage node would then be controlled by the main backup server residing on the LAN (see below). The underlying idea of this setup was to prevent direct communication between the backup server and the client, other than commands issued by the server to initiate the backup operation.



Network setup for scenario 3.

Test results showed that traffic did not flow as we had expected. Both the client and backup server communicated with each other, thus defeating our original purpose of having a separate storage network. The only reason a setup like below would be used would be for performance reasons, as backup data does not pass onto the LAN, but is confined to the DMZ and storage network.

Conclusions

Arkeia seems to be the most favourable of the backup suites we tested when it comes down to the number of firewall rules needed to make it work.
Legato uses fairly large port ranges by default, however, it has facilities to change these ranges used by both the client and the server. This would make the risk of exposure smaller. The instructions needed to calculate these ranges are well documented on the Legato website.
The amount of work involved in configuring firewalls when used in conjunction with a backup solution should not be underestimated. Care must be taken that the backup servers and nodes are not exposed to any more open ports than absolutely necessary.
A good way to determine the ports used and the direction of connection initiations is to setup very generic rules which pass all involved port ranges with extensive logging enabled.
Firewalls may pose a serious bottle neck in the overall backup performance, especially if the firewall has a large and complex rule set. This needs to be taken into account when choosing a platform for the firewall.
Due to time restrictions we were unable to run more tests using other backup products, such as Veritas NetBackup and Amanda in conjunction with other firewall products such as OpenBSD's Packet Filter and Linux's NetFilter. We will cover these in our presentation.

References

Legato technical bulletin 354 (http://www.legato.com/resources/bulletins/354.html)
Phoneboy for general Firewall-1 configuration information (http://www.phoneboy.com)
Arkeia - How to backup across a firewall (http://www.arkeia.com/faq/faq.67.html)