Network Security, Filters and Firewalls

by Darren Bolding

This article is a general introduction to network security issues and solutions in the Internet; emphasis is placed on route filters and firewalls. It is not intended as a guide to setting up a secure network; its purpose is merely as an overview. Some knowledge of IP networking is assumed, although not crucial.

In the last decade, the number of computers in use has exploded. For quite some time now, computers have been a crucial element in how we entertain and educate ourselves, and most importantly, how we do business. It seems obvious in retrospect that a natural result of the explosive growth in computer use would be an even more explosive (although delayed) growth in the desire and need for computers to talk with each other. The growth of this industry has been driven by two separate forces which until recently have had different goals and end products. The first factor has been research interests and laboratories; these groups have always needed to share files, email and other information across wide areas. The research labs developed several protocols and methods for this data transfer, most notably TCP/IP. Business interests are the second factor in network growth. For quite some time, businesses were primarily interested in sharing data within an office or campus environment, this led to the development of various protocols suited specifically to this task.

Within the last five years, businesses have begun to need to share data across wide areas. This has prompted efforts to convert principally LAN-based protocols into WAN-friendly protocols. The result has spawned an entire industry of consultants who know how to manipulate routers, gateways and networks to force principally broadcast protocols across point-to-point links (two very different methods of transmitting packets across networks). Recently (within the last 2 or 3 years) more and more companies have realized that they need to settle on a common networking protocol. Frequently the protocol of choice has been TCP/IP, which is also the primary protocol run on the Internet. The emerging ubiquitousness of TCP/IP allows companies to interconnect with each other via private networks as well as through public networks.

This is a very rosy picture: businesses, governments and individuals communicating with each other across the world. While reality is rapidly approaching this utopian picture, several relatively minor issues have changed status from low priority to extreme importance. Security is probably the most well known of these problems. When businesses send private information across the net, they place a high value on it getting to its destination intact and without being intercepted by someone other than the intended recipient. Individuals sending private communications obviously desire secure

communications. Finally, connecting a system to a network can open the system itself up to attacks. If a system is compromised, the risk of data loss is high.

It can be useful to break network security into two general classes:

- methods used to secure data as it transits a network
- methods which regulate what packets may transit the network

While both significantly effect the traffic going to and from a site, their objectives are quite different.

Transit Security

Currently, there are no systems in wide use that will keep data secure as it transits a public network. Several methods are available to encrypt traffic between a few coordinated sites. Unfortunately, none of the current solutions scale particularly well. Two general approaches dominate this area:

Virtual Private Networks: This is the concept of creating a private network by using TCP/IP to provide the lower levels of a second TCP/IP stack. This can be a confusing concept, and is best understood by comparing it to the way TCP/IP is normally implemented. In a nutshell, IP traffic is sent across various forms of physical networks. Each system that connects to the physical network implements a standard for sending IP messages across that link. Standards for IP transmission across various types of links exist, the most common are for Ethernet and Point to Point links (PPP and SLIP). Once an IP packet is received, it is passed up to higher layers of the TCP/IP stack as appropriate (UDP, TCP and eventually the application). When a virtual private network is implemented, the lowest levels of the TCP/IP protocol are implemented using an existing TCP/IP connection. There are a number of ways to accomplish this which tradeoff between abstraction and efficiency. The advantage this gives you in terms of secure data transfer is only a single step further away. Because a VPN gives you complete control over the physical layer, it is entirely within the network designers power to encrypt the connection at the physical (virtual) layer. By doing this, all traffic of any sort over the VPN will be encrypted, whether it be at the application layer (such as Mail or News) or at the lowest layers of the stack (IP, ICMP). The primary advantages of VPNs are: they allow private address space (you can have more machines on a network), and they allow the packet encryption/translation overhead to be done on dedicated systems, decreasing the load placed on production machines.

Packet Level Encryption: Another approach is to encrypt traffic at a higher layer in the TCP/IP stack. Several methods exist for the secure authentication and encryption of telnet and rlogin sessions (Kerberos, S/Key and DESlogin) which are examples of encryption at the highest level of the stack (the application layer). The advantages to encrypting traffic at the higher layer are that the processor overhead of dealing with a VPN is eliminated, inter-operability with current applications is not affected, and it is much easier to compile a client program that supports application layer encryption than to build a VPN. It is possible to encrypt traffic at essentially any of the layers in the IP stack. Particularly

promising is encryption that is done at the TCP level which provides fairly transparent encryption to most network applications.

It is important to note that both of these methods can have performance impacts on the hosts that implement the protocols, and on the networks which connect those hosts. The relatively simple act of encapsulating or converting a packet into a new form requires CPU-time and uses additional network capacity. Encryption can be a very CPU-intensive process and encrypted packets may need to be padded to uniform length to guarantee the robustness of some algorithms. Further, both methods have impacts on other areas (security related and otherwise- such as address allocation, fault tolerance and load balancing) that need to be considered before any choice is made as to which is best for a particular case.

Traffic Regulation

The most common form of network security on the Internet today is to closely regulate which types of packets can move between networks. If a packet which may do something malicious to a remote host never gets there, the remote host will be unaffected. Traffic regulation provides this screen between hosts and remote sites. This typically happens at three basic areas of the network: routers, firewalls and hosts. Each provides similar service at different points in the network. In fact the line between them is somewhat ill-defined and arbitrary. In this article, I will use the following definitions:

> **Router traffic regulation:** Any traffic regulation that occurs on a router or terminal server (hosts whose primary purpose is to forward the packets of other hosts) and is based on packet characteristics. This does not include application gateways but does include address translation.
>
> **Firewall traffic regulation:** Traffic regulation or filtering that is performed via application gateways or proxies.
>
> **Host traffic regulation:** Traffic regulation that is performed at the destination of a packet. Hosts are playing a smaller and smaller role in traffic regulation with the advent of filtering routers and firewalls.

Filters and access lists

Regulating which packets can go between two sites is a fairly simple concept on the surface- it shouldn't be and isn't difficult for any router or firewall to decide simply not to forward all packets from a particular site. Unfortunately, the reason most people connect to the Internet is so that they may exchange packets with remote sites. Developing a plan that allows the right packets through at the right time and denies the malicious packets is a thorny task which is far beyond this article's scope. A few basic techniques are worth discussing, however.

- Restricting access in, but not out: Almost all packets (besides those at the lowest levels which deal with network reachability) are sent to destination sockets of either UDP or TCP. Typically, packets from remote hosts will attempt to reach one of what are known as the well known ports. These ports are monitored by applications which provide services such as Mail Transfer and Delivery, Usenet News, the time, Domain Name Service, and various login protocols. It is trivial for modern routers or firewalls only to allow these types of packets through to the specific machine that provides a given service. Attempts to send any other type of packet will not be forwarded. This protects the internal hosts, but still allows all packets to get out. Unfortunately this isn't the panacea that it might seem.

- The problem of returning packets: Let's pretend that you don't want to let remote users log into your systems unless they use a secure, encrypting application such as S/Key. However, you are willing to allow your users to attempt to connect to remote sites with telnet or ftp. At first glance, this looks simple: you merely restrict remote connections to one type of packet and allow any type of outgoing connection. Unfortunately, due to the nature of interactive protocols, they must negotiate a unique port number to use once a connection is established. If they didn't, at any given time, there could only be one of each type of interactive session between any given two machines. This results in a dilemma: all of a sudden, a remote site is going to try to send packets destined for a seemingly random port. Normally, these packets would be dropped. However, modern routers and firewalls now support the ability to dynamically open a small window for these packets to pass through if packets have been recently transmitted from an internal host to the external host on the same port. This allows connections that are initiated internally to connect, yet still denies external connection attempts unless they are desired.

- Dynamic route filters: A relatively recent technique is the ability to dynamically add entire sets of route filters for a remote site when a particular set of circumstances occur. With these techniques, it is possible to have a router automatically detect suspicious activity (such as ISS or SATAN) and deny a machine or entire site access for a short time. In many cases this will thwart any sort of automated attack on a site.

Filters and access lists are typically placed on all three types of systems, although they are most common on routers.

Address Translation: Another advancement has been to have a router modify outgoing packets to contain their own IP number. This prevents an external site from knowing any information about the internal network, it also allows for certain tricks to be played which provide for a tremendous number of additional internal hosts with a small allocated address space. The router maintains a table which maps an external IP number and socket with an internal number and socket. Whenever an internal packet is destined for the outside, it is simply forwarded with the routers IP number in the source field of the IP header. When an external packet arrives, it is analyzed for its destination port and re-mapped before it is sent on to the internal host. The procedure does have its pitfalls; checksums have to be recalculated because they are based in part on IP numbers, and some upper layer protocols encode/depend on the IP number. These protocols will not work through simple address translation routers.

Application gateways and proxies: The primary difference between firewalls and routers is that firewalls actually run applications. These applications frequently include mail daemons, ftp servers and web servers. Firewalls also usually run what are known as application gateways or proxies. These are best described as programs which understand a protocol's syntax, but do not implement any of the functionality of the protocol. Rather, after verifying that a message from an external site is appropriate, they send the message on to the real daemon which processes the data. This provides security for those applications that are particularly susceptible to interactive attacks. One advantage of using a firewall for these services is that it makes it very easy to monitor all activity, and very easy to quickly control what gets in and out of a network.

Conclusion

There are two basic types of network security, transit security and traffic regulation, which when combined can help guarantee that the right information is securely delivered to the right place. It should be apparent that there is also a need for ensuring that the hosts that receive the information will properly process it, this raises the entire specter of host security: a wide area which varies tremendously for each type of system. With the growth in business use of the Internet, network security is rapidly becoming crucial to the development of the Internet. Soon, security will be an integral part of our day to day use of the Internet and other networks.

Darren Bolding is a "hired geek" with Internet Partners of America. Currently designing and implementing several Internet Service Providers, he has wide experience with Routers, Terminal Servers and telecom equipment. If you have any questions, corrections or comments, Darren may be reached at darren@bolding.org or http://www.bolding.org/~darren.