RPANetwork

Internet Firewall Tutorial

A White Paper

July 2002

What a firewall does

Computer networks are generally designed to do one thing above all others: allow any computer connected to the network to freely exchange information with any other computer also connected to the same network.

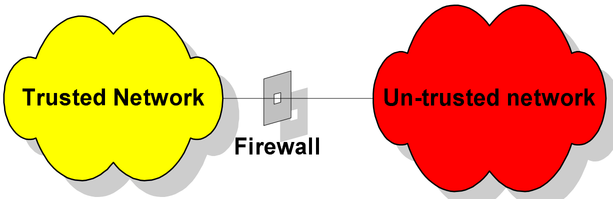
In an ideal world,this is a perfect way for a network to operate facilitating universal communications between connected systems. Individual computers are then free to decide who they want to communicate with, what information they want to allow access to and which services they will make available. This way of operating is called "host based security", because individual computers or hosts, implement security mechanisms.The Internet is designed in this way, as is the network in your office.

In practice individual computers on say, an office network, are not terribly good at defining and securely enforcing a consistent security policy. They run very complex, and therefore by definition error prone software systems, and it is very difficult to ensure that they are consistently kept secure, much less that their users obey basic advice like choosing difficult to guess passwords etc.

This situation may be adequate where individual users on a network have a similar level of trust such that there is little chance or motive for a user to subvert host security, such as a small company network where everyone with physical access is trusted (e.g employee etc).

Once that network is connected to other networks where the trust relationships simply do not exist in the same way, then other mechanisms need to be put in place to provide adequate security by protecting resources on the trusted network from potential access by attackers on the un-trusted part of the network.

The way this is done is by partially breaking connectivity at the network level so that nodes on the trusted and untrusted parts of the network can no longer freely exchange information in an unfettered way. The device which does this is called a "Firewall", by reference to the analogue in American automobile engineering, where the Firewall is a thick steel plate barrier between engine and passenger compartments which prevents a fire in the former spreading to the latter. I suppose that if this particular piece of technology had been invented on the English side of the Atlantic, it would have been called a "bulkhead" instead!



About the Author

Rob Pickering's involvement in the field of Internet security started in the 1980s when he was involved in developing implementations of the Internet TCP/IP protocol suite. He also designed and implemented one of the earliest commercial Internet connections in the UK, at a time before the widespread availability of commercial firewalls.

He has worked on Internet security strategy and implementation for major organisations like 3Com, as well as lecturing widely on computer security issues.

How it Works

A Firewall disrupts free communication between trusted and un-trusted networks, attempting to manage the information flow and restrict dangerous free access.

There are numerous mechanisms employed to do this, each one being somewhere between completely preventing packets flowing, which would be equivalent to completely disconnected networks, and allowing free exchange of data, which would be equivalent to having no Firewall.

In order to understand how each of these works, it is first necessary to understand the basics of how data moves across the Internet.

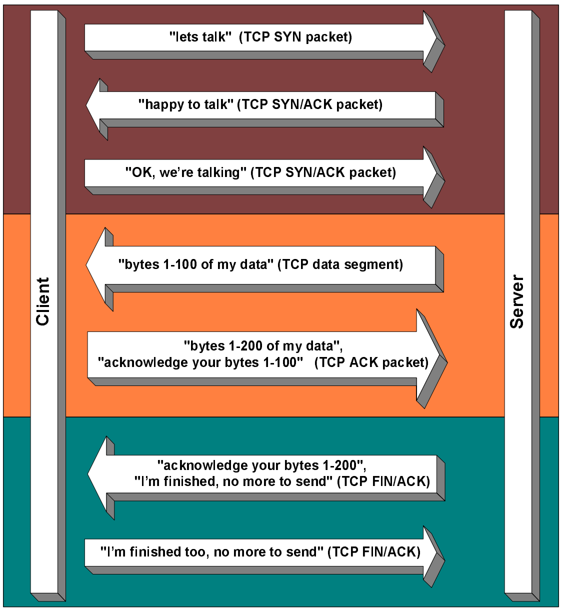
Protocols: TCP/IP

The underlying way that data moves across the Internet is in individual packets called Internet Protocol (IP) datagrams. Each packet is completely self contained, and has the unique address of the originating computer (source-address), and intended recipient computer (destination address). On it's journey between the source and destination, the packet is forwarded by routers which simply forward it on, one hop at a time to it's destination. In a non-Firewall environment these packets flow freely between the two machines.

TCP

To have a complete conversation in order to *e.g.* send an e-mail, or view a web page, a sequence of packets are grouped together using something called Transmission Control Protocol (the TCP bit of TCP/IP).

Under TCP, a complete conversation looks something like this:


Client
Server
"lets talk" (TCP SYN packet)
"happy to talk" (TCP SYN/ACK packet)
"OK, we're talking" (TCP SYN/ACK packet)
"bytes 1-100 of my data" (TCP data segment)
"bytes 1-200 of my data",
"acknowledge your bytes 1-100" (TCP ACK packet)
"acknowledge your bytes 1-200",
"I'm finished, no more to send" (TCP FIN/ACK)
"I'm finished too, no more to send" (TCP FIN/ACK)

The data part above would contain the higher level protocol which actually sends and e-mail, or requests, and gets the contents of a web page.

In order to connect to the right service on a particular host, a special identifier called a "port number" is used which routes the exchange through to the correct application program on the server end of the connection. For example, by convention, web-requests are directed at port 80, and incoming e-mails involve a connection to port 25.

Simpler Requests: UDP

TCP is a bit cumbersome for simple requests, so a streamlined protocol called User Datagram Protocol also exists. This doesn't have the same connection setup overhead and tends to be used for simpler conversations which perhaps only involve a simple information exchange, which may be repeated if packets are lost and things go wrong.

A domain name service request, used to get an IP address for a host name, is an example of a UDP exchange:

```
From A to N: UDP: Q: foo.bar.com
From N to A: UDP: A: 1.2.3.4
```

A similar port mechanism is used in UDP to route packets to the appropriate application on the host.

Determining Conversation Details

If asked to write down a security policy that we would like our Firewall to implement in English, it would probably look something like:

> *"Allow internal users to access external www servers, but not allow external users to access our Intranet server".*

In order to implement this policy, our Firewall needs to be able to examine packets and determine if they belong to either a conversation which should be allowed, or one which should be blocked.

To do this, it basically needs to know two things:

- The application being connected to.
- The direction of the conversation.

The first one of these can be guessed from the port number on the receiving end of the connection. For example, by convention, WWW servers run by default on port 80, e-mail servers run on port 25 etc.

Somewhat harder (and crucial to the above), is to determine the direction of conversation. Whilst each packet flowing through the Firewall is a self contained unit, by examining the sequence it is possible to see what the overall direction of the conversation is (*ie* who initiated it).

From the TCP transaction diagram it can be seen that the initial "Lets talk" TCP SYN packet is always seen coming from the originator of the connection, to the destination service.

Our Firewall then could implement the above security policy by translating to the following network level operations:

> *If packet is a TCP SYN from any inside address to any outside address, port 80, allow through.*
>
> *If packet is a TCP SYN from any outside address to any inside address, port 80, block.*
>
> *Allow through all other packets.*

As we will see later, this trivial algorithm isn't ideal, but it is at least a faithful implementation of the security policy shown earlier (the bug is in the security policy, not the implementation!).

Types of Firewall

There are a number of different kinds of technique which may be employed by a Firewall in order to correctly identify a conversation and act on it.

The techniques used by a particular Firewall have an impact on the accuracy with which it can identify traffic, the level of sophistication of the checks it can implement, but also it's complexity and therefore cost and likelihood that it incorporates bugs.

Packet Filter

The network level operations corresponding to the security policy above were actually an example of a simple packet filter.

A Firewall implementing a packet filter looks at one packet at a time, and considers it in isolation in order to make a forwarding decision.

Because of the way that a packet filtering Firewall works, it can implement a restricted range of filtering decisions. The principal limitations of packet filtering are:

- TCP connections can be filtered on port and direction in order to implement simple directional traffic rules keyed on port number only.
- It is not possible to completely filter TCP packets which aren't valid, or don't form part of an active connection.
- It is not possible to fully filter UDP connections to ensure that they are part of a valid conversation.

The latter restriction is a fairly serious drawback of packet filtering. It means the Firewall implementor is left with the choice of either completely blocking UDP transactions, or accepting that packets may

Glossary

BSD: *Berkeley Software Distribution - a derivative of the UNIX operating system developed under a contract from the US Department of Defense, and made publically available when the project was wound up in the late 80s. Used very widely as a secure operating system, and as the basis of many commercial firewalls and security products.*

DMZ: *Demilitarised Zone - a special network which is used for computers which need to be connected to from the Internet.*

IP*: Internet Protocol - lowest level packet format of the Internet, contains basic details about where the data came from and where it is going to.*

NAT: *Network Address Translation - technique that allows internal network to use private self managed IP addresses but still talk to other Internet systems with real addresses.*

Port number: *an identifier carried in a TCP or UDP packet which identifies which process or application within a host the conversation is addressed to.*

SMTP*: Simple Mail Transport Protocol - the application layer protocol used to send mail over the Internet.*

TCP*: Transmission Control Protocol - key reliable Internet protocol over which applications like web, e-mail etc are all carried.*

traverse the Firewall which should not be allowed through.

In the face of this, the only safe option is to block external to internal UDP transactions when using a packet filtering Firewall.

Although the above drawbacks may seem significant, there are also some quite strong advantages to a basic packet filtering Firewall:

- It is simple to implement, which means that it is much more unlikely that exploitable bugs exist in the Firewall code.
- The same simplicity means that rule sets tend to be less complex, and again are less likely to contain unintentional access routes.
- It can be implemented on relatively inexpensive hardware,meaning that simple,cheap boxes can do packet filtering for very large numbers of user connections.

Stateful Inspection

Stateful inspection takes the basic principles of packet filtering and adds the concept of history, so that the Firewall considers the packets in the context of previous packets.

So for example, it records when it sees a TCP SYN packet in an internal table, and in many implementations will only allow TCP packets that match an existing conversation to be forwarded to the network.

This has a number of advantages over simpler packet filtering:

- It is possible to build up Firewall rules for protocols which cannot be properly controlled by packet filtering (e.g. UDP based protocols).
- More complete control of traffic is possible.

Equally, there are some disadvantages to a stateful inspection solution, in that the implementation is necessarily more complex and therefore more likely to be buggy.

It also requires a device with more memory and a more powerful CPU etc for a given traffic load, as information has to be stored about each and every traffic flow seen over a period of time.

Network Address Translation

This is not really a Firewall technology at all, but is often confused with one! NAT is a pragmatic solution to the issue of IP address limitations.

When a network is connected to the Internet, the computers on that network need to be given addresses so that other computers on the Internet can send packets to them.

Because IP addresses are a somewhat limited resource, and have to be unique across the globe, they are assigned hierarchically by a central authority and passed down in blocks to service providers who then make them available to their customers.

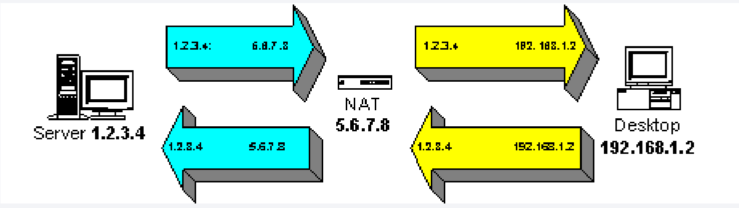
As an end customer this has some implications if you are to apply for and get sufficient IP addresses for your network:

- You need to be prepared to justify the need for all the IP addresses you will use in terms of the number of computers you have, or will have - it is not possible to obtain 10 times as many IP addresses as you need simply for administrative convenience.
- There is a bureaucratic overhead that both you and your ISP need to be prepared to undertake.
- Unless you are a very large organisation with thousands of computers who can justify a direct allocation of addresses, you will need to do this all over again when you change providers.

Many organisations and ISPs choose to sidestep these issues by only allocating a single global IP address to the customer,who then installs a NAT device at the end of the connection and uses self allocated private addresses on their internal network.

The way that NAT works is very similar to stateful inspection firewalling, but with the added twist that the Firewall modifies the address part of all packets on the way through.

The NAT gateway sees an outgoing packet from an internal private address, to an external global Internet address. It makes a note of the (internal, private) source address of the packet, and the destination server address and port number. It then overwrites the source IP address with it's own single global Internet address and sends it on towards the Internet.



The remote server receives the packet with the NAT gateway's address as the originator,and directs it's replies at this address.

When the reply packet arrives back at the NAT gateway, it looks up the address and port number in it's table, works out what the (internal) address of the real originator was, substitutes this into the destination address and forwards on through the Internal network.

Limitations of NAT

Although NAT is an extremely convenient way to avoid IP address allocation issues, the technique itself does have some limitations.

Firstly most simple NAT gateways can only deal with substituting addresses which occur at the start of the packet in an area called the header.

The designers of Internet application protocols never really envisaged the use of NAT, and some applications themselves use the address of the computer they are talking to and bury it in the application data part of the packet. Unless the NAT gateway knows about how to interpret the application data as well as the Internet headers for these protocols, then they will not operate properly in a NAT environment.

Examples of protocols that have this problem include FTP, (file transfer protocol), and a protocol called H.323 which is used extensively my Microsoft Netmeeting and similar audio/video applications. Problems with NAT and FTP are easily dealt with by using a protocol mode called passive FTP which doesn't have the same issues with NAT. Unfortunately the H.323 protocol issues are more fundamental, and you may well find that this protocol will not work with most NAT gateways.

Security Implications of NAT

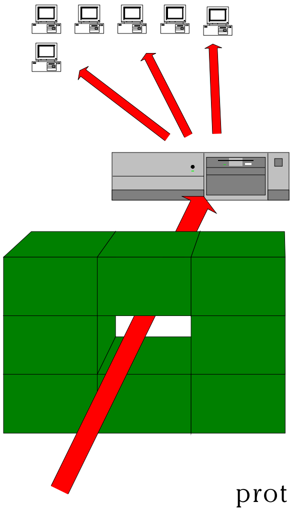
It is a widely held belief that the presence of NAT, and use of private internal addresses renders a network immediately secure. This is a most dangerous notion!

The basis of this is that with *outgoing only* NAT, an attacker cannot connect directly to a machine on the internal network, even if the Firewall rules are accidentally configured to allow this.The reasoning then goes that seeing as the Firewall is now fail-safe, the network is invulnerable.

The problem with this assertion is that it's assumption that *outgoing only* NAT will be the only thing enabled is often false, and ignores the possibility that an attacker will compromise the network not by making a direct connection at a packet level with an internal host, but will instead find another mechanism to make it call him.

Outgoing only Solution

Many simple Firewall solutions are sold by ISPs and system resellers on a "fit and forget" basis on the assumption that a simple, cheap packet filter or stateful inspection device is perfectly secure so long as it incorporates NAT, and is configured to allow only outbound connections.



The problem with this approach is that in order to do anything useful, the first thing most users need to do is open holes, or reverse NAT connections to internal servers. Once this is done, Firewall's protection can be entirely sidestepped by an attacker and information on the internal network is no longer particularly secure.

Holes and Incoming Traffic

An example of the kind of hole which is typically opened up in a Firewall is that necessary for mail delivery.

On the Internet, a protocol called SMTP is used to deliver between mail servers. This works in effect by the mail sender's machine connecting to the mail recipient's server and pushing the e-mail.

In order to accept mail from the Internet onto a local mail server it is usual to open up a hole which allows any server to connect to the local mail server.

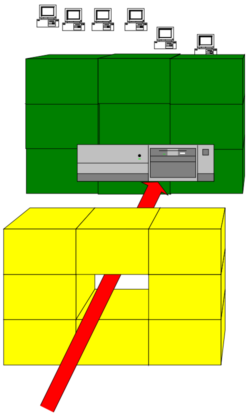
This will often be justified using logic which says that this is only a small hole to one specific service on one specific host, and the rest of the internal network is still fully protected by the Firewall "*outbound only*" rule.

Unfortunately what this does is open up the internal mail server to any attack that is possible against the software installed on it, and if this is at all complex, there will be lots of potential attacks.

As an example, a recent search on Bugtraq (an industry source of application vulnerability data) against a popular mail server, Microsoft Exchange showed that there had been 4 major vulnerabilities discovered, just between March and July 2002.

Many of these vulnerabilities would have allowed a remote hacker not only to gain unauthorised access to the server itself, but also to then use it as a launch point to attack any other system on the network, just as if the Firewall wasn't there.

No Holes: the Demilitarised Zone



The classic solution to the problem of opening up holes in the network perimeter to allow access to services is the Demilitarised Zone or DMZ. Named after the buffer zone between opposing forces in a military peacekeeping scenario, the DMZ is a special separate network of servers to which external untrusted hosts have access, but which have no access to the Internal network.

Large enterprise Internet access and Firewall systems always incorporate at least one level of DMZ as this is seen as essential to preventing the vulnerabilities described above which are inherent in opening up holes in the Firewall onto the internal network.

The issue with this solution for the medium sized or smaller enterprise is one of cost. A typical DMZ solution requires at least three

Glossary

***UDP**: User Datagram Protocol - a simple Internet transport protocol which conveys trivial repeatable requests and responses between client and server in a very efficient but less reliable way.*

***Vulnerability**: In a computer security context, a specific defect or "hole" in an application which is known to hackers and allows them to subvert security or take control of a computer.*

devices, the external Firewall, the internal Firewall, and the DMZ server machine. This means of course three times the cost which may not be feasible or proportionate for a small organisation wishing to secure it's ADSL Internet connection.

Application Proxies

Another mechanism for controlling risks when Internal servers must allow connections from the Internet is to use a technique called Application Proxies on a single external firewall.

These work by terminating the external connection at a special service within the firewall. As the name suggests, this service acts as a proxy for the real server, implementing the application protocol in the same way as the real server running on the internal network. It forms a connection to the internal server, only passing on application protocol elements that pass it's strict checks of correctness.

This way, most mechanisms for subverting the internal application server are blocked.

Using an application proxy is not without difficulty as their complexity tends to mean that they need to be implemented on firewalls which are significantly more powerful than the relatively simple systems used for basic packet filters. This, and the fact that such firewalls are typically sold to "Enterprise" customers mean that their cost is often uneconomic for small businesses.

Application proxy firewalls also tend to require frequent software updating to ensure that they are running latest versions of the proxy code. This occurs both when new exploits are identified which need to be blocked, but also when problems occur in interactions between the proxy and widely deployed applications (in other words when the proxy is actually breaking an otherwise working connection due to over strict or even erroneous checking).

A More Manageable Solution

Given that full blown DMZ design, or high deployment and management costs of an application proxy firewall, are usually not economic for businesses of 1-1000 employees with broadband Internet connectivity, a better solution is required.

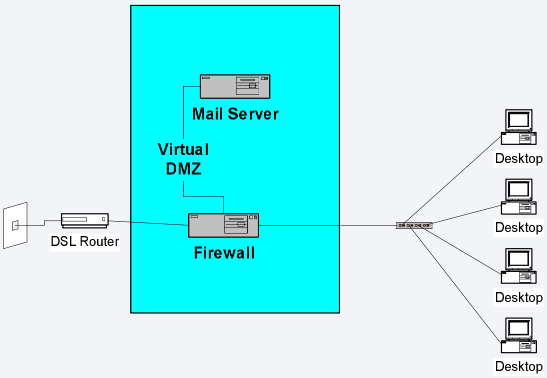
A managed solution bridges the gap between ineffective "hope for the best" fit and forget firewalls, and adequate, if expensive "Enterprise" DMZ and proxy systems.

How These Work

Taking a range of fixed, but immediately secure and useable firewall configurations, it is possible to provide fully managed firewall and network systems for a simple, low, fixed initial setup and monthly managed solution cost.

Typical Solution

Most organisations really don't want or need to invest in their own Internet server systems at all. An example, a typical configuration with extremely low cost of ownership is a managed mail server solution where we provide and manage both the firewall, and the customer's own mail server as a managed service. Based on the secure BSD operating system, all of the systems are initially set up, and continually monitored as part of the service.



The managed solution provides both the firewall, and e-mail server as a maintenance and risk free solution, leaving you to provide just the network connectivity and desktop network to which the secure systems are connected.

Other Solutions

The fully managed approach lends itself to the accessible provision of a full range of services which are normally associated with large enterprise networks

Virtual Private Networks

Using cost effective managed services, the provision of VPN systems allowing safe and cost effective sharing of information between multiple office and mobile workers becomes something accessible to even the smallest business.

Combined managed Firewall and VPN nodes provide the latest secure IPSec encryption technology, which gives you assurance that your confidential data cannot be accessed in transit. The managed dimension means you know that the network is being continually monitored for problems by experts, and software kept up to date to meet evolving threats.

Finding out more...

For information on secure managed solutions, please visit the RPANetwork site at ***www.rpanetwork.co.uk***

Alternatively feel free to call **Rob Pickering** directly on **0845 644 2805** at any time, or e-mail him at **rob@rpanetwork.co.uk**