Evolution of Network Security

Introduction

The Internet was born in a military and academic environment. In this environment, the users were invariably trustworthy, and were working collaboratively to make the technology operate for their mutual benefit. As a result of this, Internet Protocol (IP) and the standard applications that operated over IP were not originally designed with security in mind.

Today, the inherently insecure IP is still at the heart of Internet operation, along with a number of the long-standing services that run over IP, such as:

- Name lookups – Domain Name Service (DNS)
- File transfers – File Transfer Protocol (FTP)
- Email – Simple Mail Transfer Protocol (SMTP)
- Web browsing – Hyper-Text Transfer Protocol (HTTP)

The core technologies that operate the Internet are no more secure now than they were back in the trusting days when the Internet was first developed. However, now the Internet has grown to massive proportions and has millions of people connected to it, many of whom are highly untrustworthy. Online crime, mischief, espionage, extortion, and much more are ever increasing.

Therefore, Internet users need to take care to manage their data security needs. All manner of undesirables are roaming the unguarded streets of the Internet, so there must be strong defenses in place between them, and precious data and services.

Over the years, as the value of data and online services has grown, and the threats they are under have grown, the networking industry has developed a range of security devices and software to combat the threats.

This white paper provides a brief overview of how firewalls and related network security systems have evolved over the years, in order to:

- explain why security solutions have evolved the way that they have
- put some of the jargon into context
- understand the current state of the art in network security
- look some way into the future to consider what will come next

The three eras of network security

Broadly speaking, the evolution of network security can be divided into 3 areas:

1. Packet filtering
2. Session Inspection
3. Application Control

Although the evolution of anything is a continuous process, and does not jump forward cleanly from one distinct period to the next, with hindsight it is possible to recognize certain characteristics that were dominant at certain times.

The following sections describe each of these eras, and explain the reasons that drove the technology forward.

1. Packet filtering

Network data is transported in packets. Both legitimate data flows and malicious hacking attacks consist of the same thing–series of data packets. In essence, a firewall is a device that sits at the point where a network meets the Internet, and sorts the good packets from the bad ones.

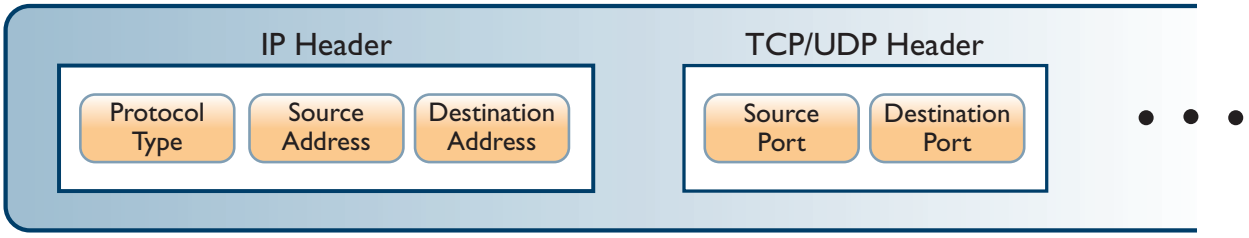
From a distance, and even close up, bad packets are not instantly distinguishable from good packets. As such, when the Internet started opening up to the public in the late 1980s and early 1990s, the networking industry needed to find reliable ways to block attacks while still allowing legitimate traffic to flow. Methods for sorting good packets from bad began to be devised.

The method that the first firewalls applied was packet filtering:

1. Decide on a certain limited set of packet types that you want to allow through.
2. Block any other type of packet.
3. Create a device that will look at every packet that it is asked to forward, and decide whether or not that packet matches the "allow through" criteria.

The criteria that packet filters look at in packets to make their decision is the so-called "5-Tuple" of fields in the packets:

- Source IP Address
- Destination IP Address
- IP Protocol Type
- Source Port Number
- Destination Port Number



IP Packet

The Source IP Address is the IP address of the sender of the packet, and the Destination IP Address is the IP address of the device that the packet is being sent to.

The IP Protocol Type is a number that gives some indication of the type of communication the packet is involved in. Examples of IP protocol type are:

- **Transmission Control Protocol - TCP (type 6)**
  This is the most well-known type of IP protocol, and is used for controlled two-way conversations. Common Internet applications such as web-browsing, Reliable File Transfers, and email all use TCP. A TCP session is a one-to-one conversation, where both ends of the conversation must agree to have the conversation before any real data is exchanged. TCP packets also contain information to enable the two end-points to decide if they have missed any packets, and request that missed packets be resent.
- **User Datagram Protocol - UDP (type 17)**
  UDP is used for communication that is essentially one-way, like video streaming, or that is a real-time two-way conversation, like VoIP. UDP does not have the overhead of session agreement or checking for lost packets, and as such is used for applications where the order in which packets arrive is important, so that resending lost packets is pointless–if a piece of phone conversation is lost, there is no point in resending it a second or so later, as the conversation has moved on by then.
- **Internet Control Message Protocol - ICMP (type 1)**
  Small messages that enable network devices to inform each other of errors, detect each other's presence, or ask each other to send more slowly.
- **Signalling protocols**
  Protocols used for advertising routes from router to router, or for choosing among multiple gateway routers, or for indicating where multicast streams should be sent–for example OSPF (type 89), PIM (type 103), VRRP (type 112) and IGMP (type 2).
- **VPN protocols**
  Packets that are being exchanged in secure VPN tunnels are typically encapsulated in special VPN protocol types like IPSsec (type 50), Authentication Header (type 51) and GRE (type 47).

The Source and Destination Port are numeric identifiers that determine individual conversations. The port numbers do not represent physical ports in any way; they are just numbers that appear at certain locations near the start of UDP and TCP packets. UDP and TCP are the only IP protocols that carry these particular identifiers, as they are the protocols that are used for conversations between end-user devices.

The port numbers are required because two machines might be running multiple conversations between each other at the same time. For example, your PC could be downloading a web page from a server at the same time it is sending an email to the same server. Both of these conversations involve TCP packets with the same sets of IP addresses, so some other identifiers are required to distinguish between which packets belong to the web browsing conversation, and which belong to the email conversation.

All the packets involved in a specific conversation will contain a specific pair of port numbers. For example:

- The packets from your PC to the server in the web browsing conversation could have source port 2792, and dest port 80 (and replies from the server back to your PC will have source port 80 and dest port 2792)
- The packets from your PC to the server in the email conversation could have source port 5492, and dest port 25 (and replies from the server back to your PC will have source port 25 and dest port 5492)

So, packet filters use these fields to determine which packets to allow and which to drop. The guiding principle is "allow through only what you MUST allow, and drop everything else". Therefore, with regard to IP addresses and IP protocol types, packet filters do things like:

- Look at incoming packets' destination IP addresses, and only let in those packets destined to addresses that it knows should be on the inside of the network.
- Look at incoming packets' source IP addresses, and only let in those packets sourced from addresses that it knows should be on the inside of the network.
- Look at outgoing packets' source addresses, and only allow those packets sourced from addresses that it knows should be on the inside of the network.
- Look at packets' IP protocol type, and only pass through packets belonging to protocols that need to be passed. For example, if the local network is not using OSPF to exchange routes with the Internet, then there is no reason to forward OSPF packets.

Port numbers are more complex. A number of conventions have evolved with regards to TCP and UDP port numbers. In particular, certain port numbers have become associated with certain types of service. For example:

- 20 and 21 are used for FTP
- 25 is used for SMTP
- 53 is used for DNS
- 80 is used for HTTP (unencrypted web browsing)
- 443 is used for HTTPS (encrypted web browsing)

Hence, email servers, for example, will accept TCP packets with a destination port value of 25. This is referred to as "Listening on port 25". Similarly, web servers listen on ports 80 and 443, and so on.

Packet filter entries involving port numbers could be defined in the following way:

- We have a web server at IP address **a.b.c.d**, so we will allow in TCP packets destined to a.b.c.d, with destination port equal to 80 or 443, and we will allow out packets from **a.b.c.d** with source port equal to 80 or 443.
- We have an email server at IP address **e.f.g.h**, so we will allow in TCP packets destined to **e.f.g.h**, with destination port equal to 25, and we will allow out packets from **e.f.g.h** with source port equal to 25.
- We allow our users to perform web browsing, so we will allow out TCP packets with destination port equal to 80, and allow in TCP packets with source port equal to 80, but not packets that constitute the start of a TCP session.

This approach to firewalling had some value:

- It weeded out clearly malicious traffic, such as incoming sessions to unusual TCP port numbers.
- It was simple to understand, and simple to implement.
- As the logic was simple, it could be easily realized in a hardware device, to provide accelerated performance.

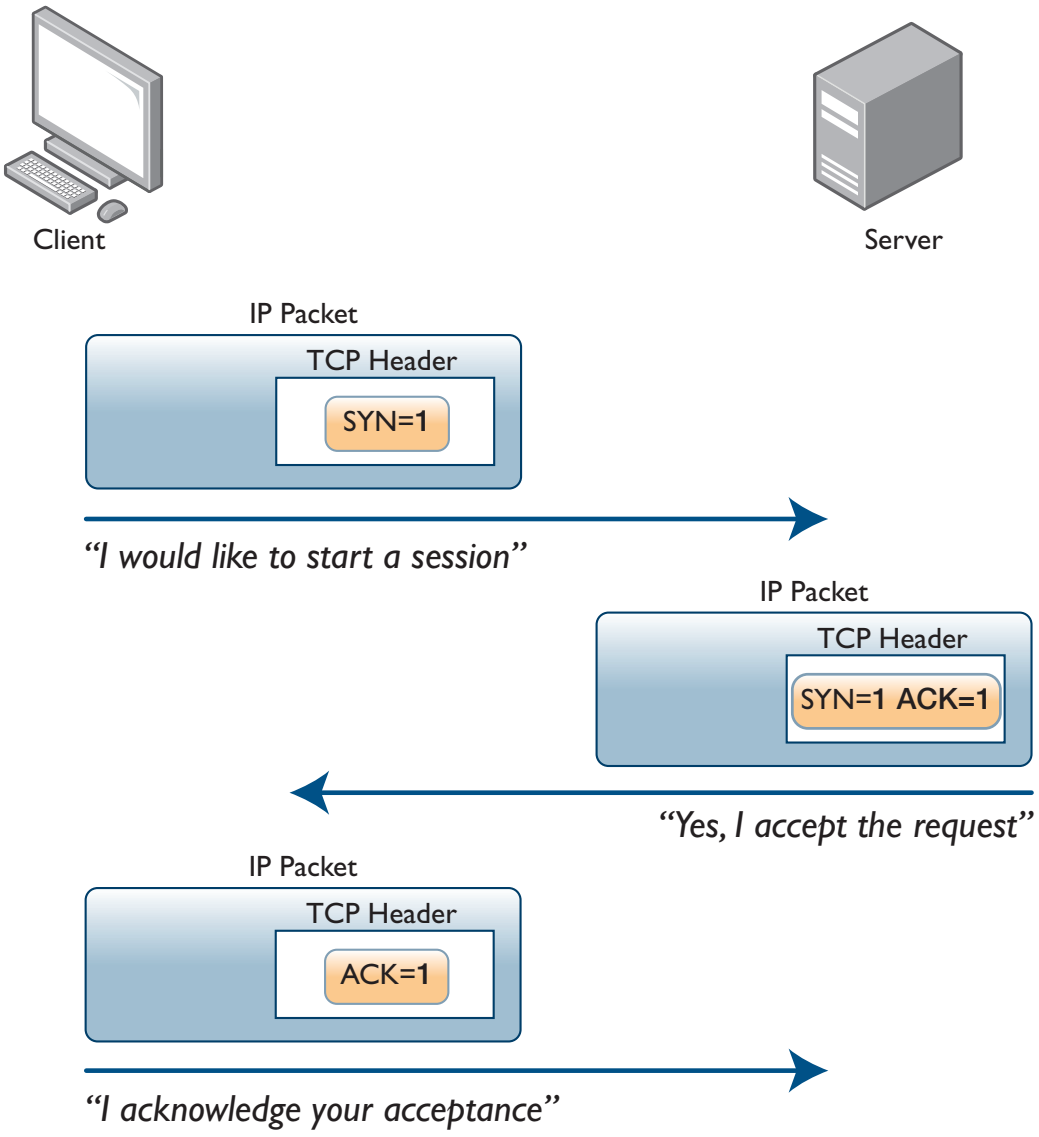
However, it also has some significant shortcomings:

- Because each packet is treated in isolation, packet filtering is not effective in blocking Denial of Service (DoS) attacks, like sending a large number of session initiation packets to a server (called a SYN attack), or sending specific sequences of fragmented packets.
- It does not have the ability to effectively deal with the subtleties of FTP, whereby the user connects to the FTP server via a TCP session to one port number, but the server connects back from a different TCP port number.
- Because it has no session awareness, it has no ability to check if a user's legitimate session has been hijacked by a malicious user.
- It has no ability to detect attacks that involve embedded malicious content deeper into packets, as it only examines packets as far as the TCP/UDP port numbers, and no further.

2. Session Inspection

The bulk of data communication consists of conversations, either dialogues or monologues. Packet Filtering can be thought of as a process of looking at each individual word of a conversation in isolation, and trying to work out whether or not they belong to a legitimate exchange. The next stage in firewall development was a move to tracking the progress of whole conversations, or "sessions". A term commonly used for these types of firewalls is "Stateful Inspection", as they maintain awareness of the state of the sessions passing through them.

TCP is particularly well suited to session tracking. A TCP session must begin with the "3-way handshake", whereby certain flags in the TCP header portion of the packet are used to indicate a request to start a session, and an agreement to that request. The sequence in this handshake is:



Packet 1: I would like to start a session with you (SYN flag in the packet is set)

Packet 2: Yes, OK, I accept your request (SYN and ACK flags in the packet are set)

Packet 3: I acknowledge your acceptance of my request, thanks. (ACK flag in the packet is set)

Similarly, a TCP session should finish with a slightly different "goodbye" handshake, where the "FIN" flag is used to indicate that it is closing time.

Further to this, during the course of a TCP session, the packets contain "Sequence" and "Acknowledgement" counters. These counters indicate how much data each of the participants believes has been exchanged so far. Sequence and Acknowledgement counters are used to detect lost data, so that the participants can request the resending of lost packets.

All these factors provide good reasons for a firewall to actively track the state of the TCP sessions that are passing through it. The firewall can:

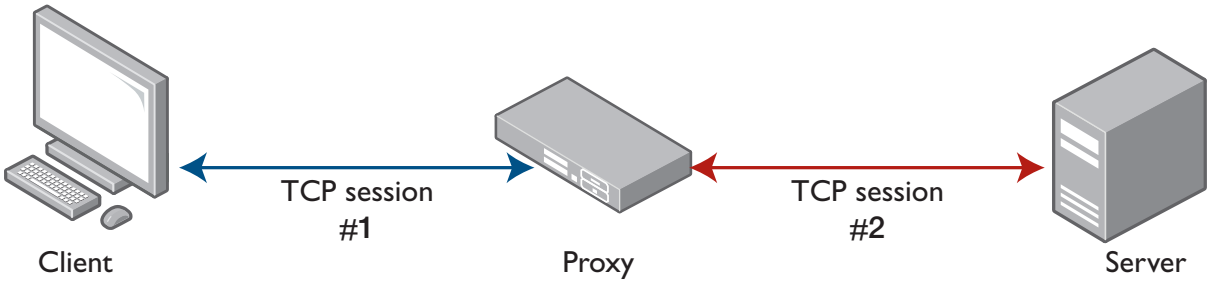
- Decide to allow in packets destined to internal workstations (as opposed to servers) only if those packets belong to existing sessions that were initiated by the workstation. This obviates the risky "and allow in TCP packets with source port equal to 80" type of rule that a simple packet filter needs to implement.
- Keep track of the sequence and acknowledgement numbers, and detect if a session is being hijacked.
- Recognize DoS attacks. If the firewall has seen that a host initiated many TCP sessions in a brief period, and they are all still in a state whereby the 3-way handshake has not yet been completed, then it can deem this to be suspicious behavior, and take action.
- Look for other, more subtle attacks that require the tracking of a series of packets, like the Local Area Network Denial (LAND) attack, which involves the sending of an invalid sequence of packet fragments.
- Apply Network Address Translation (NAT) to services like FTP or SIP, which embed IP addresses and/or port numbers inside packet data. By tracking such sessions, and knowing the point where the address/port info is exchanged between the participants, the firewall can update that content to match the address/port changes that NAT is applying to the packet headers.
- Detect attempts to scan the network for vulnerabilities. If an external host attempts to open a session on an internal host on several different port numbers in a brief period, or attempts to open sessions on a range of internal IP addresses, these are signs of scanning attempts. The firewall can deem this to be suspicious behaviour, and take action.

Although TCP is the most session-oriented of the IP protocols, it is still meaningful to perform session tracking on conversations that use other protocols, such as UDP, ICMP and IPSEC. As with TCP sessions, the firewall can then recognize whether incoming packets belong to existing sessions, and can spot scanning attempts, LAND attacks, and others.

Session-oriented firewalling is a significant improvement on simple packet filtering. However, there are still plenty of security breaches that can pass straight through a standard Stateful Inspection firewall. An example is the HTTP GET Flood attack, whereby a client establishes a valid TCP connection to a Web Server, but within that TCP session it sends a rapid series of requests for web pages, which puts a heavy processing load onto the server. Detecting this type of attack requires knowledge of the HTTP protocol syntax, which is a higher layer than TCP, and is beyond the scope of the typical Stateful Inspection firewall.

Proxy firewalls

A competing session-aware technology was the Proxy Firewall. A Proxy does not simply pass packets through, it operates as a dual end-point. It answers the TCP connection from the initiator, and re-initiates another, separate, session to the other participant. So, it is having separate conversations with the two end-points and acts as a go-between–it relays messages between the two communicators, but does not let them communicate directly to each other.



The advantage of this approach is that the Proxy is involved in the higher network layers of the conversation, and therefore will be more able to detect invalid/malicious behavior at those upper layers. Also, it can log more information about the upper layers of the conversations, providing a more detailed audit trail for possible later investigations.

The disadvantages of the Proxy are:

- It adds latency to the communications passing through it. Having to digest the information arriving from one end, and then retransmit this to the other end takes processing time.
- It is emulating a range of types of server and a range of types of client. As new services come into popular use on the Internet, the developers of the Proxy software need to develop emulations of those services. This becomes a very big job, and proxies invariably lag behind the popular usage on the Internet, and so they become a frustration to users who want to use new Internet-based services.
- The Proxy itself can become a target for hacking. Security flaws in its emulation of the services it is protecting can themselves be exploited. Taking down a network's security gateway is very tempting for hackers.

In addition, there are plenty of security breaches that pass straight through both Stateful Inspection and Proxy firewalls. Catching viruses attached to emails, malicious JavaScript embedded in web pages, data stealing, and pieces of content that target very specific vulnerabilities in specific applications, all require a deeper, more intense inspection of the data passing through the firewall.

Deeper inspection and application-specific security products

As security attacks became more subtle and complex, and their consequences became more costly, a range of solutions were developed to combat the range of threats:

- Firewall software was enhanced to look deeper into packets, to seek out sequences of bytes that constituted the "signatures" of known threats. This process is referred to as Deep Packet Inspection (DPI).
- Email-scanning software entered the market. The content of emails, which frequently runs across multiple packets, is scanned to check for spam, malicious attachments, inappropriate content, and other items that a business may want to filter out.
- Web content filters perform similar scanning of the web-browsing traffic going to/from a network. Malware embedded in this traffic can be detected and blocked, and certain web services may be blocked or selectively allowed to certain users.
- Intrusion Detection Systems (IDS) look for suspicious patterns of network activity. The SYN attack discussed earlier is a simple example of the sort of traffic pattern that an IDS would detect. Over time, databases of traffic signatures of known attacks have been built up, which provide patterns that IDSs look for.
- Data Leakage Prevention software examines data flowing out from a network to detect attempts to leak sensitive or confidential information.

It has become commonplace for businesses to have a whole set of security hardware and software products in operation, each managing its own aspect of network security. On top of that, it is not unusual to have a security management solution in place, which gathers and correlates data from the various security monitoring/enforcement products.

This state of affairs is less than ideal. Each different product:

- needs maintenance–software and signature database updates
- presents is own different paradigm for management and configuration
- has its own quirks or limitations that need to be worked around
- requires periodic subscription payments
- has potential interoperability issues with other devices in the network
- represents a possible point of failure

As result, there has been a strong trend towards unifying these various security services into a single solution that provides Unified Threat Management (UTM).

3. Application Control

The current state-of-the-art in firewall technology are appliances that provide a comprehensive suite of security capabilities, including:

- multilayer filtering–filtering based on attributes at multiple layers of the 7-layer OSI model
- virus and spyware scanning
- spam filtering
- web content filtering
- intrusion detection and protection
- SSL encryption/decryption
- IPSec and SSL VPN access concentration
- Data Leakage Protection
- IP reputation checking
- Application Control–a new characteristic, discussed below

All of these activities are performed at multi-Gigabit traffic rates, and provided with integrated graphical management and monitoring tools.

Some people refer to these as Unified Threat Management (UTM) devices, while others refer to them as Next-Generation Firewalls.

The proponents of each name have a set of reasons for their choice, and/or arguments for differentiating the products into two distinct categories, as well as explanations as to why one category is superior to the other. In a few years and with the benefit of hindsight, most of these distinctions will cease to exist, but the key characteristics of these devices will remain.

What is Application Control?

Apart from the unifying of multiple security activities into a single package, the other key innovation in the current phase of firewall evolution is Application Control.

With the emergence of the "Web 2.0" services, our interaction with the World Wide Web has become vastly more than simply viewing web pages.

Web-based services are now used for all manner of business activities, such as:

- document creation
- data storage
- customer Issue management
- video conferencing
- project scheduling
- banking
- and so much more…

Not to mention all the less business-oriented activities, such as:

- social networking
- online gaming
- buying/selling household items
- scheduling program recording on set-top boxes
- and many, many more…

The Internet can now be regarded as a massive Application Server, hosting an array of interactive applications. Rather than thinking of web browsing, FTP file transfer, DNS lookups and so on as applications in their own right, it now makes more sense to regard them as component tools that underlie the real applications that reside on the Internet.

This represents a paradigm shift from regarding the Internet as a repository of content to a repository of applications–only one of which is content delivery.

Recognizing that any organization needs to control the applications that its people use, and how they use them, the network security industry has embraced this new application-centric view of the Internet, and created Next-Generation Firewalls that have this paradigm at their core.

It no longer makes sense to categorize Internet interactions by looking at the TCP/UDP port numbers in the packets. Such a vast range of activities, both legitimate and illegitimate, now use TCP port 80 or port 443, that it makes no sense whatsoever to use those port numbers as criteria for deciding whether or not to allow sessions through. Conversely, the wide proliferation of web-based services means that all manner of other port numbers are in common usage for various activities within various applications.

Today, the fundamental question a Next-Generation Firewall needs to ask when it sees a new dataflow arrive is not “what port numbers is this flow using?” but “what application is generating this flow?” Whether it allows, blocks, or decides to perform ongoing monitoring of the dataflow is almost entirely governed by its recognition of the application to which the flow pertains.

Because applications are multi-faceted entities, a firewall can do more than just allow or deny communication with a given application. Most applications have different features within them–some of which an organization might allow, and some of which it might disallow. For example, voice communication on Skype might be allowed, but file transfers via Skype would not be allowed. Using an online application to write documents might be allowed, but using it to publish the documents would not be.

Additionally, Application Control needs to operate in a user-aware fashion. The Next-Generation Firewall must be aware that not all users are created equal–some have more rights than others. So, the firewall is aware of the identity of the user who is accessing given features within a given application, and will give different users access to different features.

Therefore, for a firewall to be truly effective in enforcing the business rules regarding online applications, it needs to understand the operation of the applications, and control how they are used, and who is using them.

Another notable aspect of the operation of Next-Generation Firewalls is the "single-pass" nature of their security scanning. Given the array of different security checks that these devices perform, they would introduce significant latency if they performed these checks sequentially, I.e. passed a packet through multiple different checking processes. Therefore, the devices employ sophisticated hardware-accelerated content filtering, which can perform all the different checks in a single pass and at high data rates.

Conclusion

The Internet is an essential tool for so much of what we do today. Individuals and businesses need to trust that the information they store online is safe, and the applications they use are not acting as conduits for bringing malware into their computers.

With the money that can be made from cyber-crime, and the temptation that offers to some very clever people, network security systems have to evolve quickly and continually in order to stay ahead.

As time has gone on, firewalls have looked progressively deeper into the data flowing through them, and have become progressively more able to interpret the meaning of that data.

This has resulted in a current generation of firewalls that take a multi-pronged approach to security scanning, and are based upon an application-centric view of users' interactions with the Internet. Unquestionably, this will not be the last word in firewall technology. As the use of the Internet continues to change, and new types of threat emerge, firewall technology will continue to develop apace, to maintain the level of security we must have in order for Internet usage to remain viable.

About Allied Telesis

For nearly 30 years, Allied Telesis has been delivering reliable, intelligent connectivity for everything from enterprise organizations to complex, critical infrastructure projects around the globe.

In a world moving toward Smart Cities and the Internet of Things, networks must evolve rapidly to meet new challenges. Allied Telesis smart technologies, such as Allied Telesis Autonomous Management Framework™ (AMF) and Enterprise SDN, ensure that network evolution can keep pace, and deliver efficient and secure solutions for people, organizations, and "things"—both now and into the future.

Allied Telesis is recognized for innovating the way in which services and applications are delivered and managed, resulting in increased value and lower operating costs.

Visit us online at **alliedtelesis**.com

C613-08018-00 RevB