Firewall implementation at Janet-connected organisations

Firewall implementation at Janet-connected organisations

Matthew Cook, Loughborough University

Authors and Contributors

This document was put together by Loughborough University to share knowledge, experience and current trends surrounding firewall implementation with the JANET community. This Technical Guide is complemented by the JANET Training programme which provides courses covering multiple facets of firewall implementation as part of their extensive portfolio.

The primary author is Matthew Cook who has been a member of the Computing Services department at Loughborough University since 1999. He has an extensive knowledge of a wide range of operating systems and networking technologies and has spoken at many conferences on various security matters. These presentations, papers and lectures are available at: http://escarpment.net/ [1].

Other contributions and editing came from Gill Chester, Andrew Cormack and Katharine Iles of JANET.

Readers are assumed to have a basic knowledge of networking concepts and preventive security awareness. A companion *Security Matters* [2] guide is available.

References to particular products do not imply any recommendation.

Scope and Audience

The Internet is a double-edged sword: it is an excellent technology for communication and collaboration, but at the same time threats are created by this connectivity. Tools to protect against malicious and otherwise unwanted network traffic began to be developed very early in the history of the Internet. Firewalls to filter traffic are now a key part of any modern day network infrastructure and the concept has extended from a system administration term into everyday vocabulary.

Firewalls can be employed at different locations on the network, and in different configurations. Understanding what each of these can (and cannot) achieve is essential to

procuring and managing the right firewall system(s) to address a particular set of requirements. This Technical Guide surveys the theory and technology behind the implementation of firewalls on campus LANs of JANET-connected organisations. This Guide is aimed at the staff responsible for the implementation of an IT security policy in these organisations, of which a firewall is a single, but critical, part. Understanding the detailed functions of any firewall system is essential to its successful operation and continued maintenance. A firewall is a long term investment, with most solutions installed for approximately three to five years. Moreover, in addition to the technical issues, consideration needs to be given to customer support, documentation and training.

The Threat of the Internet

The Internet is growing at an exponential rate: the number of people connected to the Internet was recorded as 1,086 million by the Internet World Statistics organisation on 18 September 2006 (http://www.internetworldstats.com [3]**)**.Of course, it is difficult to ascertain the accuracy of such a number, but it gives an idea of the sheer volume. Moreover, this estimate was based on the number of IP addresses in use, so may have been distorted by the use of NAT and PAT technologies which can conceal many computers and people behind one or a small pool of real-world IP addresses.

With the increasing number of people and computers connected to the Internet, the danger from Internet-based threats also rises, for a number of reasons. The Internet becomes a more attractive arena for malicious activity because there is a higher target population, with an increased dependency on the technology and greater variety of connected devices, but less specialist knowledge of how to deal with the risks. In addition, there are far more people actively working on malicious code for personal or financial gain.

It is easy to see why it is essential that threats to a local IT infrastructure are managed effectively. It is a serious incident if a computer is compromised and backdoors installed, but this also creates the possibility of theft of credit card details and access to Building Management Systems, remote IP surveillance cameras and a whole host of information stored on local information systems. Apart from the potential harm to staff and property, the obvious public relations issues and media reporting of the issue, there will be significant staff time involved executing disaster recovery plans.

A simple DoS attack which prevents users reading their e-mail may be manageable for a few hours, but prolonged outage will become mission-critical very quickly.

With the increased dependency on information systems in a culture of online learning and 24/7 access, it is critical that a well-implemented firewall is part of any organisation’s overall IT security policy.

What Does a Firewall Protect Against?

A firewall provides a level of protection from network-based attacks by allowing good traffic and denying bad traffic as defined by a security policy.

In normal operation the firewall allows a user to carry out usual IT activities (sending and receiving e-mail, web browsing, file transfer) but also prevents unauthorised access to these

systems, DoS attacks and sometimes Peer to Peer file sharing. What services users can and cannot access depends on the local IT policy.

Any machines connected to the Internet will be vulnerable to repeated attacks; recent research has shown that attacks can happen as frequently as every minute. Scanning attempts to establish what weaknesses exist in computer systems. Malicious network worms and malware popups are just some threats a firewall can protect against.

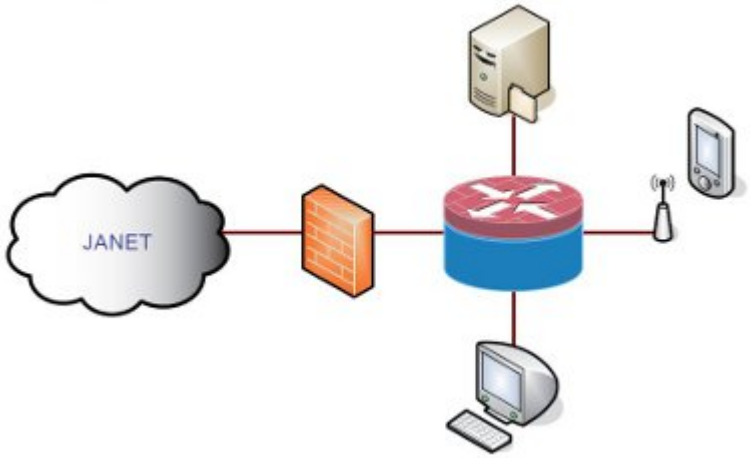
Firewall rules are commonly based upon the TCP/IP suite of protocols. The security policy to be implemented is written as a set of rules which apply restrictions to a block of network addresses (a netblock) and to a protocol and transport. The rules could be very generic (no FTP), or very specific (no FTP inbound from host 192.168.1.10). One small error in the firewall rules could leave systems without protection or, alternatively, block all access to an essential service.

Traditionally, firewalls have been installed at the gateway to the organisational network. This methodology has been the standard for a long time, but is starting to become a minimum configuration. Threat propagation within the LAN is increasing, to which the UK Higher and Further Education communities are particularly vulnerable.

[4]

Figure 1: A typical organisational firewall implementation

To provide a more comprehensive protection policy, firewalls can be installed between different network segments or VLANs by adding hardware firewall cards to network switches and routers. The best thing is a set of router ACLs.



[5]

Figure 2: Protecting internal networks with multiple firewalls or firewall modules

One of the key problems facing network administrators is the growing population of mobile IT users. Laptop or Portable users can bring a computer that is infected with a network worm into the organisation, connect it to the LAN and bypass all the firewall defences installed at the perimeter. With the drive for increased provision for ubiquitous computing, this creates

another IT security issue to consider.

Defence in Depth

Defence in depth was originally used to describe multiple levels of defensive lines, fortifications and barriers to attack which were employed in a military context. The term is now synonymous with IT security theories. Providing multiple levels of protection can enhance the overall security of an organisation instead of relying simply on one system, such as a firewall.

Following the defence in depth principle would suggest the implementation of firewall technology at the ingress points to an organisation's network, such as the traditional Internet connection, wireless networks, backup routes, extranet mobile devices or anywhere that external Internet connectivity is possible. This defence can be further enhanced with the introduction of a cleaner feed from a first tier device or multiple firewalls. Moreover, firewalls should be employed alongside other security systems, including IDS, IPS and content filters, which should also provide warning information and feedback.

In addition, multiple vendor technology can be used to provide enhanced protection. If there are security issues with a product from a particular vendor, then this is an obvious risk until a patch or work-around is released. With protection from two different vendors, the defence is increased as the same issue will not usually be a vulnerability in both.

Servers and desktop computers also should adhere to the same security policy, and antivirus/ malware should be installed on all computers. Servers should have their operating systems hardened and should ideally run different anti-virus software to the desktop computers. Services installed on the servers should be sandboxed and configured to mitigate machine compromise.

Firewall or Router ACLs?

As many active network devices such as routers and switches include ACLs, it can seem at times that firewalls are redundant. However, firewall rules and router ACLs are not equal in functionality.

Network router ACLs are not a replacement for a firewall implementation, but can complement the security of the network. They can implement simple packet-blocking rules at high speeds and, depending on the hardware, can be applied in hardware ASICs for increased performance.

However, ACLs on network devices create additional load on the router or switch which will degrade performance. Another consideration is the limited logging space and flexibility available. In addition, they cannot provide packet inspection at higher levels.

History of Firewalls

Packet filters seem to be first mentioned in connection with the Xerox Alto operating system in 1976, followed by implementations on the UNIX® platform in the early 1980s.

A paper by J Mogul et al, *The Packet Filter: An Efficient Mechanism for User-level Network Code*

, published in 1988, finally realised the usefulness of the firewall concept (http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-87-2.pdf [6]).

Later that year, the Morris Worm created by Robert Morris of Cornell University is reputed to be the first Internet worm. It made use of the computer systems of MIT and was designed to gauge the size of the Internet. The academic merit of such an activity was questionable in the first place, and the flawed architecture of the code created an even bigger problem. The code could install itself on a system multiple times and therefore render the system unusable — a primitive but effective DoS attack. The author had foreseen that checking if the code was already present on a system would provide an easy and effective mechanism for defence and therefore decided to design it to replicate regardless 14% of the time. With the worm exploiting known vulnerabilities in Sendmail, FingerD and RSH privileges, it spread at an alarming rate. This highlighted an issue with network security and led to the further development of firewalls as we know them today.

The early firewalls, although basic, still form the cornerstones of the technology as used today. Development by AT&T Laboratories and Gene Spafford of Purdue University created the second generation of circuit level and application layer firewalls.

The first reported commercial implementation of a firewall came out of the work of Marcus Ranum and was released by the Digital Equipment Corporation as the SEAL product. Later development focused on dynamic packet filtering which formed part of the first commercial offering from Check Poinr® in 1994.

Development of dedicated Kernel Proxy architecture firewalls resulted in the release of the Cisco Centri Firewall in 1997. The Kernel Proxy architecture uses the concept of network sessions to dynamically construct a stack for the session and its specific protocol. Firewall development is constantly evolving and incorporating more and more features. Firewalls including IPS and mitigation technology are becoming commonplace.

Modes of firewall operation

Modes of firewall operation

Different Types of Firewall

Firewalls operate in different ways, depending on the layer on which they are implemented.

Circuit level firewall: data link layer

Various names are used for this type of firewall. It is the type provided whenever NAT and PAT technology are used (see 3.7).

When a protected computer starts a conversation with a remote computer, the traffic is intercepted by the circuit level firewall, which forwards the request. When the return traffic

reaches the firewall, the internal tables are checked to establish if it needs forwarding to a protected computer or if it is a non-requested conversation.

The key advantage of this kind of firewall is that only return traffic from conversations that were initiated from behind the firewall will be allowed through. As there is no direct connectivity between the protected computer and the external network, any unrecognised conversations are dropped. However, this can also be a disadvantage as anything requested by the protected computer will be received even if it is malicious content. SOHO routers, commonly used for broadband connections at home, generally provide a circuit level firewall through NAT and/or PAT.

Packet filtering firewall: network layer

Firewalls acting at the network layer were the first to be developed and are probably the best understood by network administrators. They work by examining each packet against a set of defined rules.

These rules usually relate to:

- source and destination IP addresses
- source and destination ports
- protocol at transport or network layer (IP, TCP, UDP, ICMP etc)
- physical interface
- direction (ingress or egress)
- packet state.

On the whole, packet filters can only allow or drop and log traffic. The traffic contents are not altered. When a packet is received it is evaluated against a set of rules and once the packet matches a rule the defined action is taken. This means that the order of the rules is critical as the first match found determines what happens to that particular packet. Rules are easily defined using simple logic. For instance, to block all incoming SMTP traffic, a rule can be defined for all TCP source traffic to the local destination network matching SMTP port 25.

Packet filtering can also remove other network traffic. For example, it can match specifically TCP or UDP as well as ICMP at the network layer or IGMP.

Application firewall: application layer

Firewalls acting at the application layer inspect traffic at a much higher level than traditional firewalls. They can be network devices placed inline, proxy servers to handle specific traffic or applications running on a server to filter traffic to a particular program.

Firewalls on the application layer operate differently to those on the network layer because of how data is transmitted across networks. Each chunk of data consists of two parts, the ‘header’ and the ‘payload’. (Using a postal analogy, the header is the envelope and the payload the letter inside.) Conversations between computers comprise many of these chunks of data, known as packets. An application layer firewall can inspect the payload as well as the header and can look at a series of packets together.

One of the key features of an application layer firewall is its facility to block any packets that

do not comply with the RFC standard for the protocol being inspected. For example, an exploit against a web server that uses a subtly altered HTTP packet will be blocked. An application layer firewall can also act as a content filter: by examining the payload, packets containing Java™, ActiveX® or malware can be blocked. Content can even be reassembled and virus checked before being passed on to the end user.

In most modes, the operation of such a firewall is transparent to the user, except for the time lag caused by the decoding of the complete packet. With faster performance devices and the optimising of rule sets this is now rarely a concern, though earlier dedicated proxy servers required configuration of individual applications.

A relatively new term for the unwanted content that can be filtered by an application layer firewall is Anti–X. This covers a range of different areas, including anti-virus, ad/spy/ malware, worms, spam and phishing.

Traditional proxy server/web cache appliances can sometimes offer some of the features of an application firewall as well as the advantages of caching contents. However, most application layer firewalls are also excellent reporting tools and can generate numerous auditing logs in combination with facilities for authentication and real-time alerting. The term ‘application firewall’ can also be used for applications which intercept content for sanity checking before passing it to the ultimate destination. URL Scan is an example of such an application, which filters the URLs passed to a Microsoft® IIS system.

Firewall Settings

Default allow and default deny

There are two different types of firewall policy: default allow and default deny. The default allow firewall rule set allows all connections through the firewall unless otherwise stated. There is no implicit or explicit ‘deny all’ at the end of the rule set.

Early firewall policies just blocked a few known malicious signatures or activities, but as the threat level and frequency of attacks increased over time, it was recognised that it was more prudent to implement a default deny policy.

In addition, with some default allow firewalls there is a short time lag between when the firewall code is initialised and when additional rules are loaded. This presents a security risk for that short window of time.

A default deny firewall rule set will deny all connections through the firewall unless a connection matches a specific rule. An empty default deny rule set has no effective connectivity. Many current firewall systems are configured as default deny and do not even require an explicit ‘deny all’ statement at the end of the rule set. However, for the ease of logging and debugging, it is best to add the statement explicitly as a reminder.

For example, when monitoring an access list on a Cisco device, without explicit deny rules it is difficult to get a breakdown of the dropped packets using the command:

Janetrouter> sh access-list 101

Separate deny statements will provide further statistics on each protocol being dropped:

access-list 101 deny tcp any any log

access-list 101 deny udp any any log

access-list 101 deny ip any any log

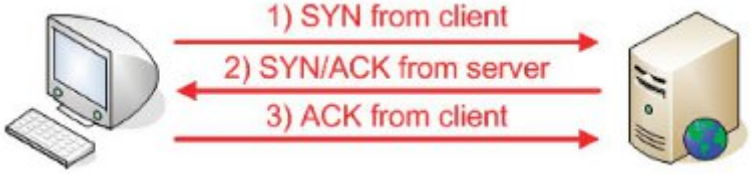
A firewall cannot possibly detect or prevent all possible threats. However, a default deny configuration will provide limited protection against zero day attacks using new port ranges, instead of none at all.

Stateless and stateful

Firewalls were initally stateless, examining each packet individually against the firewall rules. The firewall has no information as to whether the packet is the start of a new connection or part of an established one or any memory of packets that may or may not have gone before.

A stateful firewall can identify conversations and track activity to deny new connections from a hostile network, while permitting established connections to traverse the firewall. This is achieved through an internal table of attributes for each connection: IP addresses, ports, direction of flow and in some cases sequence numbers. Subsequent packets, if matched in the internal table, are then forwarded with less stringent examination because they are part of an established connection.

In the case of a three way handshake, a stateless firewall would examine each embryonic packet that forms part of the handshake. A stateful firewall would recognise the three way handshake, the continuing established connection and the resultant tear down at the end. The first packet sent from the client with the SYN bit set is recognised as part of a new connection by the firewall. The server will reply with a packet where the SYN and the ACK bit is set; this half-open state is allowed back through the firewall and recorded in the internal table. The final stage of the client replying with a packet with the ACK flag set brings up the TCP connection in the established state. UDP does not use the same three way handshake as it is a connectionless protocol, so a firewall will deem a UDP connection to be established at the receipt of the first packet.



[7]

Figure 3: Flags in TCP three way handshake

The key advantage to a state table is that the rule set is only interrogated when the first packet with the SYN bit set is received by the firewall or when the rule set changes. The subsequent packets making up the same connection will be passed or blocked without rechecking the whole rule set. When the firewall rule set changes the state table must be cleared and all initial connections are again matched against the new rule set. Sessions in the state table will time out when no traffic has been received for a period of time or when there is

a flood of packets with the SYN bit set from the same host which reach a pre-defined threshold.

Level of inspection

DPI Layer 7 inspection and other similar terms refer to the level at which a firewall, or indeed any network device examines packets.

The technology used by DPI firewalls is the same as in application firewalls and the two are often combined in modern firewalls. Both the ‘header’ and the ‘payload’ of packets will be inspected and with this extra data the firewall will be able to match packets against more complex rules.

A DPI firewall will use a set of signatures, more akin to an IDS or IPS. When a packet is matched against a signature, it can be dropped, tagged (including QoS marked), rate limited and logged. Packet signatures can be simple HTTP requests with **cmd.exe** in the URL or more complex indicators of NetBIOS/SMB worm activity.

DPI systems have a considerable overhead, which is why manufacturers are starting to implement their code in silicon ASICs to obtain faster wirespeed performance. This allows the device to maintain the state table for the stateful firewall as well as the current state of the application using the conversation.

Placing the detection technology on the firewall allows malicious packets to be detected and dropped earlier in their ingress to the network. Traditionally, an IDS system would be able to alert the network administrator to such activity, or an IPS system would use source quench, send resets or activate rules using tools like **shun**. Moving this technology closer to the malicious network can only improve the security of an organisation.

The requirement for IDS and IPS systems is not diminished, however, as it is impractical for a firewall to monitor the number of rules that a dedicated IDS/IPS system can at a backbone wirespeed. Moreover, additional functionality enhances the security in depth principle.

[8]

Figure 4: Deep Packet Inspection blocking malicious HTTP URLs

Hardware Firewalls

Hardware firewalls have long been considered a superior firewall platform. The main reasons for this are speed related: hardware manufacturers have speeded up packet throughput using hardware ASICs and optimised code. However, with advances in modern hardware platforms and the additional DPI features of newer firewalls, the speed differential is narrowing.

Cisco offers the PIX hardware solution, a packet filtering firewall with stateful inspection. There are several DPI features in newer versions which enhance the rules that can be created. Some PIX functionality has recently been introduced in the Cisco ASA.

There are many other hardware firewall vendors, including SecureIron® perimeter traffic manager from Foundry Networks® and the SSG and NetScreen products from Juniper®.

Firewall Software

Off the shelf software

There are many software packages available to provide firewall functionality on a range of platforms. There are a number of factors to be assessed in chosing a software solution but it is important to remember the firewall is only as secure as the underlying operating system and its configuration.

In addition to purchasing the software, hardware will need to be specified appropriately. Consideration should be given to CPU and memory throughput as well as the performance of the bus connecting the network adaptors. The yearly maintenance cost for the software needs to be factored into the recurring budget, along with hardware platform maintenance and support.

The Microsoft® ISA server provides web proxy functionality, VPN endpoint termination and a stateful DPI firewall. The latest version of Microsoft® ISA server was released in 2006.

Microsoft® ISA server integrates well with AD and many of its rules can be associated with AD users and computers to provide a better match for security policies. Network load balancing is included along with the ability to publish web pages securely to IIS servers within the organisational DMZ. It is also easier to facilitate firewall rules for services like Outlook® Web Access.

There are two versions of ISA Server, standard and enterprise. The enterprise version supports clustering and load balancing across multiple ISA Servers and requires Windows Server® 2003.

The Check Point Firewall-1® solution is a CD-based install onto Windows®, Sun Solaris™, Red Hat®, Nokia’s bespoke hardware platform or a bare metal customised Check Point SecurePlatform™. It provides a stateful firewall with DPI, a host of additional features and extensive management, status, reporting and auditing capabilities for managing one or many devices.

The newer NGX™ platform offers more granular multicast control as well as enhancing the number of network-based applications supported without having to write extensive rule sets in-house.

Open source

There are a range of open source UNIX/Linux-related firewall systems, all of which run on an installed operating system. However, this software suffers in the same way as other software

in the area of wirespeed performance.

NetFilter is the firewalling code in the Linux kernel and provides a packet filter firewall, NAT, Connection Tracking and other features. IPTables is the tool which creates the rules to be managed by NetFilter. IPTables code replaced IPChains from Linux 2.2 (which in turn replaced the ipfwadm code in Linux 2.0). IPTables is far superior to IPChains as it allows the firewall code to operate in a stateful manner monitoring the state of the connection using the tracking layer of NetFilter.

Rules are grouped into chains, which are sets usually following a common thread, for example, protocol or netblock. Instead of a packet being checked against each rule in a long list, the check can branch to different chains to obtain a closer match and reduce the time lag. If a check reaches the end of the chain without finding a match, then the global policy for that chain dictates the action, usually to drop the packet.

IPFirewall (or ipfw) is a FreeBSD® and Mac OS X-based IP packet filter which has the additional functionality of a traffic accounting feature. IPFilter (or ipf) is installed by default in Solaris 10, FreeBSD and NetBSD. Both ipfw and ipf are available as loadable kernel modules or can be included in a fresh kernel.

IPCop is a Linux-based distribution designed purely to provide a firewall platform. The firewall is based on the Linux NetFilter code and provides the same stateful firewall in an easy to manage solution.

There are a number of add-ons for IPCop which provide Anti-X style functionality along with traffic data reporting and the SNORT IDS.

M0n0wall is an embedded firewall distribution based on FreeBSD. It can be installed like IPCop, or can run from a LiveCD or on an embedded system. M0n0wall provides a stateful packet filter firewall, NAT, VPN endpoint termination and a captive portal.

Firewall location and configuration

Firewall location and configuration

Design

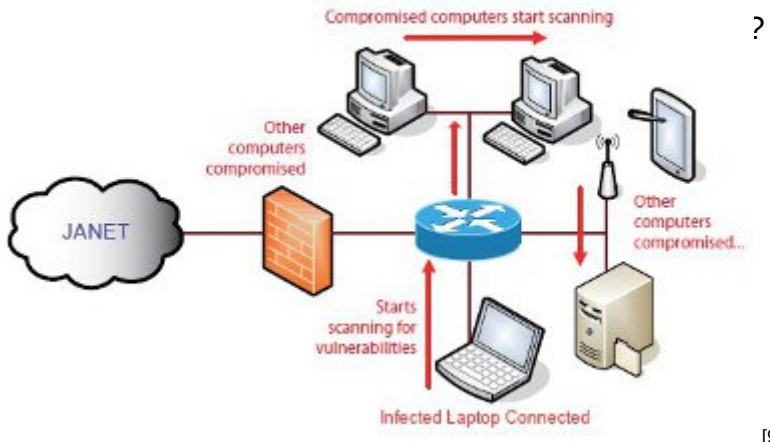
The perimeter of an organisation’s LAN is the obvious place to locate its security protection. However, the perimeter has moved: no longer is there just one ingress to a LAN, but many points of access. Wireless networks, modems, secondary Internet connections and the migration of laptops between networks mean the boundary is constantly moving.

Moreover, it is critical to ensure that new or updated firewalls will be future-proof. The increase in network bandwidth available has been significant in recent years and is stretching firewall resources. Firewalls need to be able to manage the maximum available upstream

bandwidth, otherwise a DoS attack could result in firewall failure.

Isolating networks

Logically, networks are already isolated by the netblocks that define them. To enable communication, routing protocols allow traffic between configured netblocks. However, do ?



[9]

Figure 5: The spread of infection on a campus LAN

Malicious code — worms in particular — can propagate through many different attack vectors. One of the more common is the Microsoft Windows® NetBIOS/SMB/CIFS.

Attacks taking advantage of insecure NetBIOS/SMB/CIFS fileshares and vulnerabilities in the underlying code are a common cause of swift spreading LAN-based infections. A good form of defence against this is to block these protocols from LANs where not required. The ports TCP/UDP 137, 138, 139 and TCP 445 can be blocked between LAN segments to isolate individual networks from attack. If a computer is compromised by a worm in one segment, only other vulnerable computers on that segment are compromised. However, when core file, print and AD authentication is required, then these protocols will have to be allowed to the network where the servers are located.

With the increased volume of different attack vectors against varying software, the implementation of a default deny policy between networks is becoming common, with exceptions set to allow desired services to operate.

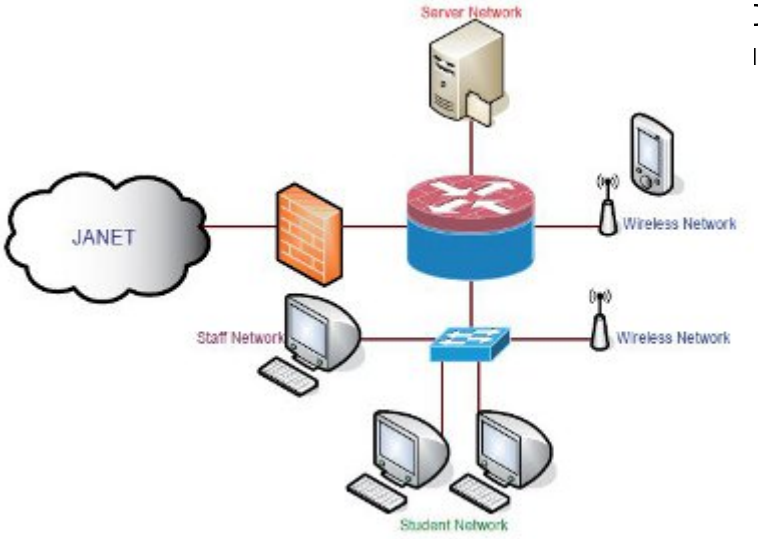
Isolating different classes of users

Different classes of users often require different levels of access to IT systems. Many organisations already separate student and staff networks. The practice is very sensible, as it provides another layer of defence. However, a problem occurs when student and staff traffic meets in open access areas or on wireless networks. It is good policy to allow access to specific information systems only from wired staff networks.

Isolation can be achieved using VLAN technology which is already common on most organisational networks. Many VLANs can exist on a network at the same time, although there are limits on some vendors' hardware.

VLANs allow networks to be separated so that different policies can be assigned to each. There are additional benefits from a networking perspective, including a reduction in broadcast domain sizes and easier administration.

802.1Q trunking provides a method for multiple VLANs to be fed to networking devices. VLANs are tagged with IDs for differentiation with ID 1 reserved for the default untagged traffic. Most vendors use the 802.1Q standard, except for 3Com®, who use Virtual LAN Trunk. Cisco historically used Inter-Switch Link, but this is now deprecated over 802.1Q. However, security measures which rely on VLANs are not entirely dependable because of the technique



D tag on encapsulated packets in a
ıction network.

[10]

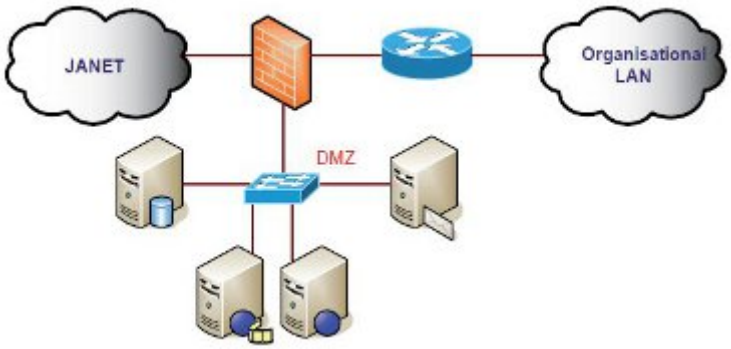
Figure 6: Use of VLANs to separate network traffic

Demilitarised Zone

A DMZ describes a network in which the host servers are located. Limited connections from the Internet are allowed into the DMZ to provide services like web (HTTP) access and e-mail (SMTP et al). Connections from the DMZ to the internal network are not usually allowed by default, which protects the computers inside from compromised hosts in the DMZ.

Hosts in the DMZ are frequently additionally protected using NAT or PAT to further obfuscate the networking configuration.

A DMZ is often implemented using a third physical interface on the firewall, but an alternative is to use two firewalls in series with the DMZ. This provides an additional level of protection for the internal network.



[11]

Figure 7: Organisational Demilitarised Zone implementation

Protecting Sensitive Information Systems

Some computer systems on any network are more critical than others. Computers which store sensitive information present a higher risk because of their attractiveness for attack and its subsequent impact. Sensitive information systems can be protected by a number of different methods and using a combination of these provides defence in depth and enhances overall security.

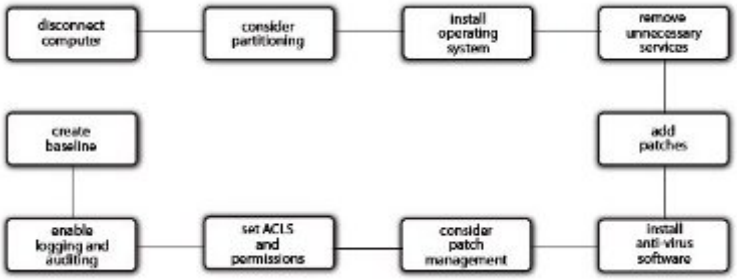
Host security

It is important to ensure computers are built and maintained in a secure manner to prevent intrusion through operating system and configuration vulnerabilities. There are a number of steps that can be taken to secure an operating system, from the most basic at installation stage to more granular changes post-configuration.

It is good practice to ensure the computer is either disconnected from the network entirely or connected to a heavily firewalled development network at build time. Operating systems are often far from secure during installation and being connected to the production network would leave the computer vulnerable. To reduce the risk of a DoS attack, different disk partitions for system volume, user storage, individual services and logs is ideal. It is also worth considering whether all the services enabled are actually required: does IIS or Apache need to be running on all computers?

Post-installation it is essential that all operating system and service patches are applied. This needs to be achieved securely, not via an unprotected network. Anti-virus software needs to be installed, along with the patch management software such as yum, smpatch, Windows® automatic updates, SUS or WSUS.

Once the machine has been configured, all ACLs and permissions set, and all logging and auditing enabled, it is wise to create a machine baseline snapshot. This will give a standard to compare the computer against should it begin to behave differently. It will make it easier to identify additional open ports or CPU-intensive processes.



[12]

Figure 8: Host security implementation plan

Isolation

Increasingly, information systems are isolated from other systems. This can take the form of dedicated VLANs, small netblocks per system groups, NAT, PAT, router ACLs or additional firewalls.

When a system is isolated, the traffic both into and out of the system is restricted which means the system is more difficult to compromise. If the system does get compromised then spread is significantly reduced.

Firewalls

Firewalls installed specifically to protect information systems can provide another layer of protection and dedicated rules. It is recommended that, if possible, two different hardware vendors are used to provide security against vulnerabilities in the firewall code.

The firewall could be host-based or network-based, although it is worth remembering that host-based firewalls are typically inferior. They are inflexible and often fail open, as opposed to network-based firewalls which fail closed.

Protecting a number of machines with a variety of requirements behind a firewall can be achieved with virtual firewalls or different contexts.

Network ACLs

An alternative to a fully functioning firewall is to protect information systems using network ACLs. Network ACLs can be implemented on routers or on some network switches. With Cisco Enhanced Images, network ACLs can be implemented on incoming traffic.

Access lists provide the flexibility to filter packets at both ingress and egress of network interfaces, according to IP address, protocol and application.

Secure communications

Even with modern packet-switched Ethernet, there is still a possibility that communications traffic can be sniffed. For example, tools like macof can be used to turn switches into hubs if they are not suitably protected. Secure communications can also be used to prevent antireplay and man-in-the-middle attacks.

All authentication and other sensitive data should be secured. If the protocol being used

does not support secure encryption, then an SSL tunnel can be employed.

Physical Interfaces

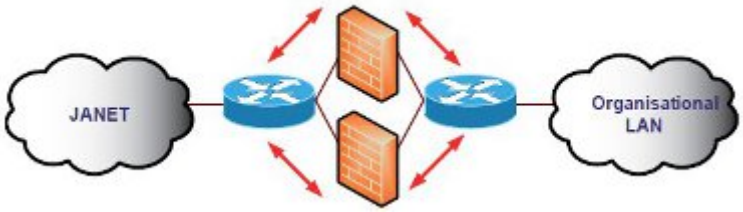
All firewalls will have a number of interfaces which can be physical or virtual (or sub). Physical interfaces are where actual cables are connected to attach the firewall to the network infrastructure. All firewalls must have a minimum of two physical interfaces for normal operation, but this is not a limit. Interfaces for DMZ, management and failover all present configuration options.

Virtual interfaces are used when there are fewer physical interfaces available than required, or to support VLANs and/or virtual firewalls/contexts. Virtual interfaces split a physical interface into separate interfaces depending on the 802.1Q trunk. It is recommended that at least the primary firewall for an organisation has physical inside, outside and DMZ (if appropriate) interfaces, as they are, by their nature, more secure than virtual ones.

Failover

The provision of failover is a key issue in firewall implementation as fault tolerance needs to be a priority within the network infrastructure.

Failover can be implemented in two ways traditionally: Active/Active and Active/Passive. With the Active/Active method, two firewalls run concurrently, sharing the traffic to provide failover should one fail.



[13]

Figure 9: Firewalls in an Active/Active failover operation

With Active/Passive failover, two firewalls run concurrently, but traffic is only handled by one. When failure occurs, the other firewall takes over.

[14]

Figure 10: Firewalls in an Active/Passive failover operation

Multiple contexts can be used to create numerous virtual firewalls with different configurations on the same piece of hardware. This enables two devices to balance the load and provide fault tolerance.

[15]

Figure 11: Firewalls configured for Active/Active failover operation and load balancing using multiple contexts

Router ACLs and CBAC

Router ACLs were the first protection technology implemented by organisations. However, they can increase resource usage and CPU overhead. Dedicated firewalls are more flexible and can provide better fault tolerance. Router ACLs should only be used to isolate netblocks and implement limited rules. Core Cisco chassis-based routers can offload firewall features using a FWSM which can provide 1,000 virtual firewalls per installation.

CBAC is a Cisco IOS option for existing routers which monitors packets and implements a Layer 3 stateful inspection. This is a good solution for small installations.

CBAC also provides DoS protection and enforces timeout and threshold controls. This includes restricting the total number of half-open sessions and rules based on time scales and hosts.

When a packet is received at an interface it is evaluated against the existing outbound access list, and may be permitted to pass. (A denied packet would simply be dropped at this point.) The packet is then inspected by CBAC to determine the state of the packet's connection. This information is recorded in a new state table entry created for the new connection.

Based on the state information, CBAC creates a temporary access list entry which is inserted at the beginning of the external interface's inbound extended access list. This entry is designed to permit inbound packets that are part of the same connection as the outbound packet just inspected. The outbound packet is then forwarded out of the interface.

Later, an inbound packet reaches the interface which is part of the same connection established with the outbound packet. The inbound packet is evaluated against the inbound access list, and is permitted because of the temporary access list entry previously created.

The inbound packet is then inspected by CBAC, and the connection's state table entry is updated as necessary. On the basis of the updated information, the inbound extended access list temporary entries might be modified in order to permit only packets that are valid for the current state of the connection.

Any additional inbound or outbound packets that belong to the same connection are inspected to update the state table entry and to modify the temporary inbound access list entries as required.

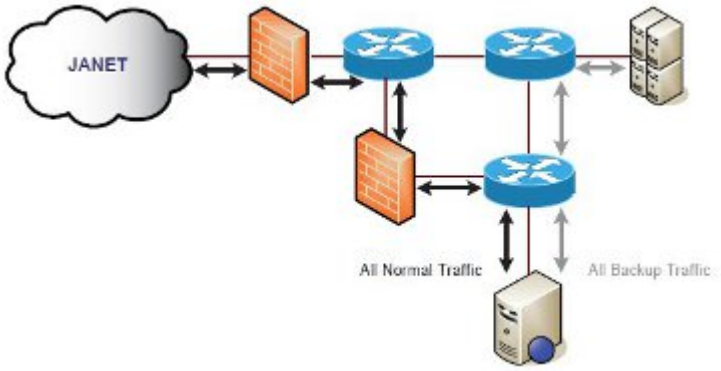
When the connection terminates or times out, the connection's state table entry is deleted and the temporary inbound access list entries are deleted.

Policy Based Routing

PBR is used to enable routers to make decisions on where to route traffic according to policies configured on the device. This can divert traffic around a firewall or ensure it always goes through it.

With backup traffic, it can be useful to ensure that it is diverted around a firewall rather than overwhelming it. A rule can be constructed to identify traffic between a source network and backup machines on particular ports. When a router sees a match for this traffic, it is directed to a particular netblock instead of using the routing table to identify an appropriate entry.

PBR is very flexible and can match packets on not only addresses but ports, protocols and packet size. PBR can also be used to provide cut-through routing between a private network a ally need to traverse the public Internet.



[16]

Figure 12: Policy Based Routing used to route backup traffic

Operational Modes

Firewalls can be configured to operate in a number of different modes and some can even operate in multiple virtual modes.

In routed mode, the firewall acts as a router deciding where traffic should go and whether it should traverse the firewall. If the addresses on the inside interfaces are not Internet-facing, then the firewall will have to use NAT or PAT to translate the traffic.

Network Address Translation

NAT allows a single device, such as a router, to act as an agent between the Internet (or ‘public network’) and a local (or ‘private’) network. This means that either a single or a pool of Internet IP addresses is required to represent an entire group of computers to anything outside their network.

When an internal computer requires a connection to the Internet, the NAT router accepts the request and translates the private IP address (e.g. 192.168.1.10) into a public address (e.g. 193.60.199.195). The mapping between them is entered into a table and the request forwarded to the Internet. The return packet is checked against the table to find the originating private IP address and then forwarded inside the network.

If more than one computer requests Internet content, additional IP addresses are used from the pool in a one-to-one relationship. An address is only used while a session is in progress and it is returned to the pool once the request has been completed. Once the pool of addresses has been exhausted, no internal machines can make further Internet connections until an address becomes free. However, one configuration often implemented is to provide one additional public Internet IP address by PAT to enable translation once the NAT pool has been exhausted.

NAT Zero is often used for DMZ computers: a one-to-one mapping is configured so a public-facing Internet IP address is assigned to each computer which NAT translates to a private address.

Port Address Translation

PAT is a similar technology to NAT, except instead of providing an Internet IP address for each internal computer from a pool, it uses a single Internet IP address and a different port for each request.

When a request is received by the PAT router from a computer inside the network, the request is forwarded to the public Internet IP with a source port specific to that request. The source port is entered into a table so the response can be translated back to the original internal private address.

PAT is more often used where only one Internet-facing IP address is available, for example, on home broadband routers (though the technology is often advertised as NAT). PAT could use a maximum of 65535 ports and therefore 65535 simultaneous requests from internal computers. However, there are some limitations: in most implementations PAT will not use the well known ports 0-1023. In addition, the processing power required to use all the remaining ports would be considerable, beyond the scope of most appliances and generally impractical. Cisco, for example, recommend a practical limit of 2000 connections, and therefore ports, using PAT.

Alternative modes

An alternative mode of operation is the Transparent, Bridged, Bump-in-the-Wire or Stealth mode firewall. This is a firewall which acts like a traditional network bridge, filtering traffic that

traverses it. The two physical interfaces are the two bridge interfaces and are not allocated IP addresses. Traffic between the inside and outside networks is simply bridged. This type of firewall is the easiest to install as it requires no alteration to network numbering, and acts at the data link layer (Layer 2) instead of the network layer (Layer 3). Two key benefits of this mode are the perceived improved security as the firewall device will not be easy to detect, will not appear on a traceroute and will not be accessible if the firewall interfaces are not assigned IP addresses. Secondly, there will be a performance improvement due to the simpler operation and the removal of the routing requirement. For management, an additional physical interface can be configured and placed on a protected management network or managed out-of-band.

Firewall rules and logging

Firewall rules and logging

Good Practice

All firewalls work on the premise of rules configured to implement the site security policy. Rules are so critical to the operation of the firewall that it is vital they are fully understood before a firewall implementation is deployed.

The most important thing to remember is that firewall rules are processed in sequence. When a packet is analysed it is checked against each rule in turn until it finds a match or reaches the end. If the firewall finds a match the action defined is executed: this could be allow or deny with the option to log. If no match is found then the default action of the firewall is executed. The default action is, again, allow or deny, depending on the firewall configuration. It is good practice to define an explicit allow or deny at the end of each rule set.

It is important to optimise the rules so that as few rules as possible are checked before a hit is made. However, exceptions need to be placed before explicit deny rules. For example, to block all outgoing HTTP traffic except that from the web caches, a rule would be written to drop all outgoing HTTP. This would, initally, block all traffic including that from the web caches. An additional rule would need to be added before the default deny to ensure traffic from the IP addresses assigned to the web caches is allowed.

IP addresses should be used instead of DNS hostnames, as it is easy to subvert DNS queries to undermine the firewall rules. It is important to ensure that when the firewall rules have been changed they are completely reloaded to ensure there is no conflict with previous rules.

It is critical that rules are as specific as possible to ensure that correct packets are matched and the number of false positives is kept to a minimum. The best advice is to block everything at first and then open holes to increase functionality one step at a time.

Regulations and Guidelines

The Data Protection Act

There are two elements of firewall implementation where the DPA 1998 [17] specifically applies:

- If an IP address can identify an end user then the storage of logs is subject to the DPA.
- It is possible that content of communications may be revealed with Layer 7 inspection.

This is regarded by the law as well as by users as more of an intrusion and will therefore require a stronger justification.

If the content of packets may be revealed or recorded, laws governing interception are also likely to apply, in particular the Regulation of Investigatory Powers Act 2000 [18].

The implementation of a firewall as part of a complete IT security policy will go some way to ensuring that the seventh principal of the DPA is addressed:

> ‘Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data’.

At present there is very little case law to determine ‘appropriate measures’. However, taking steps to provide an effective firewall is a good move forward.

ISO/IEC standards

The standards ISO/IEC 17799:2005 and ISO/IEC 27001:2005 provide a comprehensive collection of information security good practice advice.

Previously known as BS7799, the standards have been updated and revised. BS 7799 Part 1 was updated in June 2005 to become ISO/IEC 17799:2005. Seventeen controls were added while others were deleted and merged to give a total of 133 information security controls. This document will be re-released in April 2007 as ISO/IEC 27002:2007.

BS7799 Part 2 was revised in 2005 as ISO/IEC 27001:2005 and defines a framework for an Information Security Management System. This can be used to design, monitor and control security policies and rules within an organisation.

BS ISO/IEC 27001:2005 does not specifically refer to firewalls, but to a number of controls concern policies that would affect the operation of a firewall system. These include controls relating to the isolation of networks and systems which would be difficult to implement without the use of VLAN technology, router ACLs or firewalls. The purchase and installation of a firewall will not cover all the suggested controls in BS ISO/IEC 27001:2005. However, it can be used as a tool in the organisational information security policy to mitigate risk.

Blocking Common Threats

There are many Internet-based threats to network security, but there are some that every organisation should be blocking. Examples of possible rules are given in the NetFilter/

IPTables format (see 2.4.2).

With a default deny firewall, it is less critical to ensure all malicious attack vectors are protected against as they will be blocked inherently. However, if there is not a default deny rule in place, the rules necessary to implement the organisation's security policy will need to be considered carefully.

Stealth TCP Scans

Port scanning activity is common, with many different methods used in attempts to subvert firewalls, IDS and IPS. Blocking undesirable combinations of TCP flags can counteract this:

iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP

iptables -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

iptables -A FORWARD -p tcp --tcp-flags FIN,RST FIN,RST -j DROP

iptables -A FORWARD -p tcp --tcp-flags ACK,FIN FIN -j DROP

iptables -A FORWARD -p tcp --tcp-flags ACK,PSH PSH -j DROP

iptables -A FORWARD -p tcp --tcp-flags ACK,URG URG -j DROP

OS Fingerprinting

Crackers using port 0 fingerprinting can be prevented from identifying hosts on the inside network using the following code:

iptables -A FORWARD -p tcp --dport 0 -j DROP

iptables -A FORWARD -p udp --dport 0 -j DROP

iptables -A FORWARD -p tcp --sport 0 -j DROP

iptables -A FORWARD -p udp --sport 0 -j DROP

Protocols

There are a number of protocols that are unlikely to be used across the Internet and therefore can be blocked at the organisational firewall:

- Net BIOS/SMB/CIFS – TCP/UDP 135-139 and TCP 445
- Microsoft® RPC over HTTP – TCP/UDP 593
- BootP/DHCP – TCP/UDP 67 and 68
- SNMP – TCP/UDP 161and 162
- NFS – TCP/UDP 2049
- Microsoft® SQL – TCP 1433
- MySQL – TCP 3307

- LPD – TCP 515
- TFTP – UDP 69
- Simple UNIX Services (TIME, CHARGEN etc)

Blocking ICMP

ICMP is a straightforward method for transporting communication messages and errors. As with many of the Internet protocols, it has been abused over the years and the original purpose subverted. ICMP is used regularly for DoS attacks, network scanning and tunnelling other communications.

Whether ICMP should be blocked is a hot topic of debate for network administrators. Some consider ICMP an essential part of network diagnostics and should therefore be allowed, while others believe it opens an undesirable security hole.

One compromise is to rate limit ICMP and only allow certain packet types. ICMP fragmentation is seldom seen outside a malicious attack, so it is recommended to block these packets explicitly, and allow only ICMP echo-reply, echo-request, time-exceeded and packet-too-big (these are required, respectively, for the **ping** command, traceroute and MTU discovery).

iptables -A FORWARD -p icmp --fragment -j DROP

iptables -A FORWARD -p icmp --icmp-type echo-reply -j ACCEPT

iptables -A FORWARD -p icmp --icmp-type echo-request -j

ACCEPT

iptables -A FORWARD -p icmp --icmp-type time-exceeded -j

ACCEPT

iptables -A FORWARD -p icmp --icmp-type fragmentation-needed

-j ACCEPT

iptables -A FORWARD -p ICMP -j DROP

access-list outside deny icmp any any fragments

access-list outside permit icmp any any echo-reply

access-list outside permit icmp any any time-exceeded

access-list outside permit icmp any any packet-too-big

access-list outside deny icmp any any

access-list inside deny icmp any any fragments

access-list inside permit icmp any any echo-request

access-list inside permit icmp any any time-exceeded

access-list inside permit icmp any any packet-too-big

access-list inside deny icmp any any

access-group outside in interface outside

access-group inside in interface inside

Logging

Log files are critical to the successful management of network devices. They can be taken from network hubs, switches, routers, firewalls and almost any other network device.

Most firewalls have limited local logging so the logs need to be extracted for analysis and examination. Logs can be the only indication an attack has occurred and can provide the only information about it. It is often easier to trace problems if logs are aggregated together in the same place. This also protects against crackers removing the traces of log files when they compromise a system.

SNMP monitoring of firewalls can be used to generate logs by sending requests or receiving traps. This information can be stored in logs, sent out as alerts or graphed. Logging may not only show security issues. For example, an exhausted DHCP scope where computers are using automatic Microsoft-generated addresses is shown in Cisco PIX logging output.

Network Time Protocol

The timestamp is critical to log files, especially on systems with a high level of transactions. Accurate time synchronisation is essential to the smooth running of any server, especially if running directory services. E-mail headers can sometimes give a clue to the timezone of the originating computer system or if the clock is not synchronised by NTP then it may be possible to guess the time difference fairly accurately.

When quoting or comparing timestamps, it is necessary to know whether the system's clock is synchronised to an external reference, and what timezone the times refer to.

Note: The JANET NTP service provides this level of accuracy. Time synchronisation can be completed with one of four JANET time servers. However, it is recommended that at least two local systems are used to distribute time around an organisation instead of all computers synchronising across JANET. For further information see:

http://community.ja.net/library/janet-services-documentation/network-time-service [19]

UNIX/Linux-based firewalls

UNIX/Linux-based firewalls

Why use UNIX/Linux?

Firstly, unless a network administrator is familiar with the UNIX/Linux platform, then it is not recommended that a UNIX/Linux firewall is implemented. It will be easier to maintain a secure system if the administrator has skills in maintaining the underlying operating system. UNIX/Linux systems typically rely on a CLI more than a GUI, making navigating the system more difficult for the first time user.

Modern Linux kernels (from versions 2.4 and above) include NetFilter/IPTables which filters incoming, outgoing and forwarded traffic. Earlier Linux kernels provided IPChains (2.2) or the ported IPFirewall code (2.0).

UNIX/Linux systems can be configured easily with a low system footprint, making a small, clean install with the minimum configured services. Creating a system with a small footprint makes it easier to maintain and more secure, as operating system updates are only required for packages that are installed.

When building Linux systems there is the opportunity to build a custom kernel. The ability to create a custom monolithic kernel has advantages and disadvantages. The standard kernel included with distributions has a number of options enabled and others disabled and extensions can be included as loadable modules.

Creating a custom kernel provides the ability to incorporate just the elements of the kernel that are required for the operation of a firewall, which includes networking drivers and enhancements like SELinux, GrSecurity, SMP and Magic SysRq.

SELinux and GRSecurity are extensions to the Linux operating system which provide local security enhancements. These are particularly useful on multi-user interactive systems and can also provide protection against buffer overflow attacks and similar exploits.

SMP extensions to the kernel provide support for multiprocessor systems. Magic SysRq supplies a number of operations which can be accessed using the SysRq key in the event of a system crash. These operations can also be achieved using a serial line connection allowing out-of-band crash recovery, disc sync and reboot.

The main disadvantages in creating tailored kernels are the inability to use the vendorsupplied kernel updates and the limited support provided on most contracts. It is essential to weigh up the pros and cons of both solutions.

Requirements

As the firewall is being built on a PC-based hardware system, it is important to consider the hardware configuration required to run it. Linux distributions will run on most hardware vendor platforms and therefore compatibility with major hardware should not be an issue. The

hardware required will be dependent on the following factors:

- wirespeed
- complexity of rules
- number of interfaces
- scalability
- data retention
- policy and support.

For a small organisation with a 10Mbit/s internet connection and a default deny firewall policy, implementing a simple two interface routed NAT firewall storing data for 14 days can be built on a very small footprint, often an old desktop PC.

Larger organisations with 100Mbit/s internet connections, more complex firewall rules, more interfaces and longer data storage will require more powerful platforms.

The firewall is one of the most critical pieces of hardware in the organisation's network infrastructure, so ensuring a reliable platform which is supported by maintenance is paramount. Most hardware vendors have grades of support from within working hours to 24 hours a day, seven days a week. Support can usually be purchased for three years at the time of procurement and then extended for a further year at a time.

Hardware

When purchasing new hardware, it is important not to procure a machine that is underspecified and will cripple the firewall. It is always worth using a server grade motherboard for a production server, along with a rack mount chassis with dual power supplies.

Memory is not a critical requirement: a firewall servicing 100Mbit/s wirespeed typically uses no more than 512Mb of RAM, although as memory is relatively inexpensive, it is worth purchasing 1Gb as a minimum.

CPU speed is also not a critical requirement: the same firewall will run with a small load average using a twin Pentium® Xeon 2.8Ghz processor. Dual processors are not an absolute requirement, although it helps reduce any lag on a heavily-loaded firewall.

It is recommended that good quality branded network interfaces are used, and are installed with a view to future expansion and dedicated out-of-band management. A common installation is Intel® PRO (S) interfaces, with the PCI-X variants in server boards.

Disc space is required for booting the firewall itself and log file storage. This can be achieved with a RAID mirrored pair of system discs and the remaining discs used for log file storage.

Distribution

As with the operating system on which a firewall is based, the choice of distribution is just as critical and should be dependent on the operator's familiarity with it.

Most modern Linux distributions such as Fedora, SUSE® and Ubuntu will use an

implementation of NetFilter/IPTables. NetFilter is the firewalling code in the Linux kernel that provides a packet filter firewall, NAT, Connection Tracking and other features through kernel modules. Sample scripts for building the firewall are included with the distributions and further clues are often found in the **rc directory**. Fedora Linux includes the lokkit tool for easy configuration from the command line or from the GNOME desktop.

OpenBSD distributions use the pf packet filter included in the operating system kernel to provide firewall functionality, while FreeBSD® and Mac OS X use the ipfw packet filter. The latter system is very similar in operation to NetFilter/IPTables.

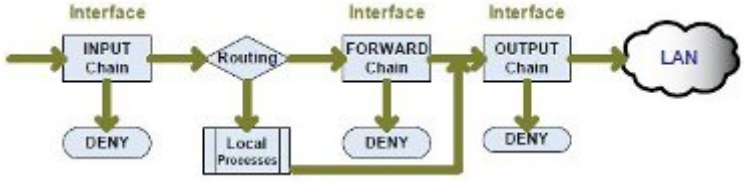
Solaris 10, NetBSD and earlier versions of FreeBSD use the ipf packet filter, which is installed by the operating system. This can provide similar firewall services as a loadable kernel module or be included when compiling a fresh kernel.

Firewall software

The choice of firewall code will probably be determined by the preferred operating system and distribution. All the different firewall systems look very similar on the surface, but they are subtly different underneath. IPChains is not supported by most modern distributions so is only mentioned here in passing. The rest of this section will concentrate on NetFilter/ IPTables and demonstrate how it differs from ipfw. IPTables is the name of the tool which creates the rules to be managed by the overall NetFilter firewall code.

The ipfw code supports three filtering chains: INPUT, FORWARD and OUTPUT. All packets are processed by the INPUT chain, then if accepted, routed to the local computer or forwarded.

Packets forwarded are subjected to the FORWARD chain and finally all packets, including those locally generated, are processed by the OUTPUT chain. Therefore, an ipfw firewall will process forwarded packets against three chains.

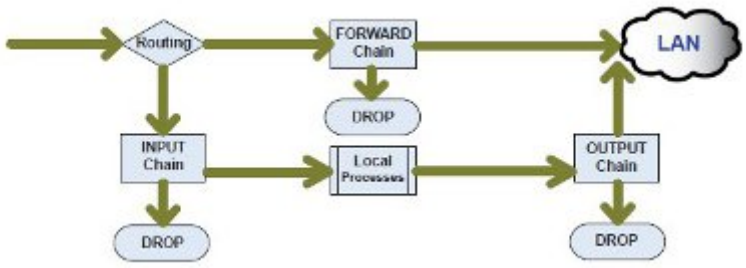


[20]

Figure 13: How ipfw processes a packet through the firewall

NetFilter/IPTables initially uses three chains: INPUT, FORWARD and OUTPUT. All packets are firstly interpreted by the routing element.

If a packet is to be forwarded, it is only processed by the FORWARD chain, before being directed straight out of the appropriate interface. If a packet is destined for the local computer or locally generated then it is processed by the INPUT and OUTPUT chains respectively.



[21]

Figure 14: How NetFilter processes a packet through the firewall

There are many small differences between IPTables and previous firewall code. IPTables is built of many modules. For example, the INPUT and OUTPUT interfaces are identified separately. Translation functions are separated from packet filtering. The logging option is now a rule target and NAT is separate from the packet filtering element.

NetFilter firewalls are constructed using the very powerful IPTables tool which controls the creation and management of all the elements of the firewall.

There are three tables of multiple chains within NetFilter:

- FILTER
- NAT
- MANGLE

The FILTER table is the most commonly used and by default holds the chains FORWARD, INPUT and OUTPUT. The NAT table provides Network Address Translation functions and the MANGLE table is used to alter packets as they are inspected by the firewall code.

The syntax and order of IPTable commands are very strict and follow a standard layout:

iptables <option> <chain> <matching criteria> <target>

The most commonly used **option** is –A to add a rule after the last currently active rule in a chain.

The **chain** entry will be the default INPUT, OUTPUT or FORWARD in the filter table, a user-defined chain, or one contained in the NAT or MANGLE tables.

Matching criteria is the statement which identifies the packets to be acted upon. This could be by source, destination, port number or other supported methods.

Finally the **target** is where the packet is destined, whether it is dropped, logged, allowed or manipulated further.

Implementing a NetFilter/IPTables firewall

Initialising the firewall

The easiest way to create a firewall is from a shell script that is executed when the computer running the firewall starts. On a Fedora or SUSE system, the convention would be to place

the script in **/etc/rc.d/** named appropriately as **rc.firewall** or **rc.netfilter**.

The firewall script should be owned and executable by root only. The firewall script can then be started from **rc.local** each time the computer boots.

It is possible to use variables in the shell script for the local network and for specific networks for services. However, the use of variables is down to personal choice. An example of a variable would be:

LOCAL=”192.168.1.0/24”

WEBSERVER=”192.168.1.80”

As with any shell script, it starts by calling the shell in which to execute the script:

#!/bin/sh

As a security measure to avoid starting the computer with no firewall rules, the network configuration for the firewall can be included in a script, which in turn loads the firewall rules.

For a Linux bridging firewall with two interfaces, this script would configure the network before loading the rules.

#!/bin/sh

echo Bringing interfaces up...

ifconfig eth0 up

ifconfig eth1 up

echo Sleeping...

sleep 10

echo Bringing bridging up...

brctl addbr br0

brctl addif br0 eth0

brctl addif br0 eth1

brctl stp br0 off

echo Sleeping...

sleep 10

echo Adding bridge IP address and default route

ifconfig br0 up

ifconfig br0 inet 192.168.1.2 netmask 255.255.255.0 broadcast

192.168.1.255

echo Adding firewall rules

/etc/rc.d/rc.firewall_rules

Customised rulesets

Once a basic firewall script has been created, there are a number of specific services which may still need to be blocked. It is impossible to provide an exhaustive list, but as a starting point some are listed below:

- Net BIOS/SMB/CIFS – TCP/UDP 135-139 and TCP 445
- Microsoft® RPC over HTTP – TCP/UDP 593
- BootP/DHCP – TCP/UDP 67 and 68
- SNMP – TCP/UDP 161and 162
- NFS – TCP/UDP 2049
- Microsoft® SQL – TCP 1433
- MySQL – TCP 3307
- Microsoft® RDP – TCP 3389
- LPD – TCP 515
- TFTP – UDP 69
- Simple UNIX Services (TIME, CHARGEN etc)

Many web sites offer advice on what should be blocked at the firewall, but it is important to tailor the rules to the software and services running at your organisation. When using a particular piece of network-based backup software, it would be prudent to ensure that it is blocked from off-campus access.

When defining the rules, it is essential to specifically allow services which are running on your network which require Internet access. The most important thing is to make the rule as specific as possible. For example, when creating a rule to allow SMTP access to Message Transfer Agents on your internal network, do not allow port 25 access to the entire network, but ensure it is only allowed to specific hosts.

For example, do not use:

iptables –A FORWARD –d 192.168.1.0/24 --dport 25 –j ACCEPT

Instead, use these rules for the primary and secondary mail routers:

iptables –A FORWARD –d 192.168.1.25 --dport 25 –j ACCEPT

iptables –A FORWARD –d 192.168.1.26 --dport 25 –j ACCEPT

iptables –A FORWARD –d 192.168.1.0/25 --dport 25 –j DENY

The third line, which explicitly denies port 25 traffic to the rest of the netblock, is very useful

when first creating firewall rules and also when debugging. Explicit deny rules like this will be shown on screen by the iptables **–L FORWARD –v** command along with hit counters to show how many times each rule has been matched.

When optimising rules for a production environment, if there is a explicit DENY for all traffic at the end of the rule set, some DENY rules may be removed. However, rules can only be removed if there will not be an accidental match further down the rule set.

Multicast

Multicast traffic is often overlooked when setting firewall policies. Multicast is enabled by most RNOs and is an integral tool for many collaboration projects. Internally, multicast is used when deploying computer images using packages like Norton Ghost™, and for locating resources with Service Location Protocol and Rendezvous/Bonjour on Mac OS X. It is advisable to ensure that appropriate rules are in place to protect multicast traffic, as an example of creating a user-defined rule. Multicast traffic can be identified and then examined in further detail in the MCAST chain.

iptables -F MCAST

iptables -X MCAST

iptables -N MCAST

iptables -A FORWARD -i br0 -d 224.0.0.0/4 -j MCAST

 # NTP

iptables -A MCAST -d 224.0.1.1 -j DROP

 # SGI IRIX objectserver/directoryserver

iptables -A MCAST -d 224.0.1.2 -j DROP

 # rwhod

iptables -A MCAST -d 224.0.1.3 -j DROP

 # Service Location Protocol

iptables -A MCAST -d 224.0.1.22 -j DROP

 # Microsoft Active Directory server

iptables -A MCAST -d 224.0.1.24 -j DROP

 # Service Location Protocol Directory Agent

iptables -A MCAST -d 224.0.1.35 -j DROP

 # Cisco’s PIM Auto Rendezvous Point

iptables -A MCAST -d 224.0.1.39 -j DROP

iptables -A MCAST -d 224.0.1.40 -j DROP

 # HP Device Discovery

iptables -A MCAST -d 224.0.1.60 -j DROP

 # Sun RPC

iptables -A MCAST -d 224.0.2.2 -j DROP

 # Altiris Ghosting / Multicast Usenet

iptables -A MCAST -d 225.1.2.3 -j DROP

 # Norton Ghosting

iptables -A MCAST -d 229.55.150.208 -j DROP

 # Source Specific Multicast (SSM) groups

iptables -A MCAST -s 233.80.58.0/24 -j ACCEPT

iptables -A MCAST -d 233.80.58.0/24 -j ACCEPT

iptables -A MCAST -d 232.0.0.0/8 -j DROP

 # ImageCast ghosting

iptables -A MCAST -d 234.42.42.42 -j DROP

 # ImageCast ghosting

iptables -A MCAST -d 234.142.142.142 -j DROP

 # Locally (admin) scoped groups

iptables -A MCAST -d 239.0.0.0/8 -j DROP

Logging

By default, logging from the NetFilter/IPTables firewall code will be logged as **priority 4 kern.warn** messages and written to **/var/log/messages**. The priority can be changed with the **--log-level option**. The creation of the messages is controlled by the syslog process.

Syslog works on a series of facilities and the priority of alerts sent from them. The facilities are categories of event which allow easier diagnostics and log file analysis. The priority of alerts is set by the programmer for the software in question. The alerts are used to determine if and where the messages should be logged. This is all configured in **/etc/syslog.conf**. Firewall logging can be redirected to a new file by adding a line to **/etc/syslog.conf**, but it is important to make sure an empty file exists and other log management software like log rotate is

updated.

kern.warn/var/log/firewall

Note: In some older UNIX/Linux distributions the white space in the **/etc/syslog.conf** file requires tab characters instead of simple spaces.

Open source projects

Open source projects

SmoothWall and IPCop

For some time, projects have been running to establish dedicated appliances based upon Linux, such as firewalls and web caches. These projects provide a pre-configured tool specific to one task which is easy to implement and configure.

The SmoothWall project was created by Richard Morrell and Lawrence Manning in 2000 and rapidly became the open source firewall implementation of choice for those wanting a cheap firewall on standard PC hardware. SmoothWall was based upon the Linux operating system and came pre-configured with all user-changeable options available through a web interface, negating the requirement for Linux experience. However, commercial interests led to a split in development efforts, including the creation of IPCop in 2003.

IPCop is a Linux-based distribution with the sole purpose of providing a firewall platform. The firewall is based upon the Linux NetFilter code and provides the same stateful firewall in an easy to manage solution. It is open source under the GNU Public Licence and is available on CD-ROM in more than 17 languages. IPCop is used by many organisations in the UK HE and FE communities.

Installation requires a dedicated computer, although a VMware image can be used for familiarisation with the project and interface. The minimum hardware requirements are a staggering 386 processor, 32Mb of RAM and a 300Mb hard disc, but lower specifications of hardware will not provide performance for faster wirespeeds and more advanced features.

Features of IPCop include:

- secure configuration through any web browser
- support for multiple interfaces
- DHCP Server daemon for the internal network
- VPN server with certificate support
- caching DNS server
- web cache
- Intrusion Detection System
- traffic shaping and QoS facilities.

Configuring IPCop after installation is easy, using the machine console or once the interfaces have been configured, through any web browser. Later versions also support a HTTPS (SSL) connection. The interfaces are easily identified by colour: green for inside, red for outside, orange for the DMZ and blue for wireless.

Once configured, like any hardware used as a server, IPCop can operate ‘headless’ without keyboard and monitor.

M0n0wall

M0n0wall is an embedded firewall distribution based on FreeBSD®. It can be installed like IPCop, or can run from a LiveCD or on an embedded system. M0n0wall provides a stateful packet filter firewall, NAT, VPN endpoint termination and a captive portal.

The distribution is very similar to that of IPCop with the primary addition of the captive portal. A number of UK HE/FE sites have been using the captive portal element of M0n0wall as part of the early LIN trial before the full JANET Roaming service was created.

Cisco security appliances

Cisco security appliances

Overview

Cisco offers the PIX firewall solution, acquired from Network Translation in 1995, along with the original Centri firewall which ran on the Windows NT® operating system. The Cisco PIX, however, runs its own proprietary system called PIX OS, currently at version 7. The PIX is a packet filtering firewall with stateful inspection, and there are several DPI features in the newer versions which enhance the rules that can be created.

The Cisco PIX is available in a small SOHO unit and a rack-mount form factor. More recently, Cisco introduced some of the PIX functionality in the Cisco ASA. Cisco also offers the FWSM in a blade form factor for the larger chassis router. This has been updated to the same code level as the Cisco PIX.

The Cisco PIX and ASA have numerous features including:

- purpose-built security appliances
- stateful packet inspection
- routed, NAT and bump-in-the-wire operation
- user authentication of connections
- protocol and application inspection engines for Layer 4 through 7
- VPN connectivity for site-to-site and as endpoints
- virtual firewalls
- stateful failover.

PIX versus ASA

Changes to the product range mean Cisco PIX Firewalls have been renamed Cisco PIX Security Appliances. This has resulted in the creation of a security appliance family which includes both the PIX and ASA devices, as well as the Cisco FWSM and the basic Cisco IOS firewall.

The Cisco ASA is not a direct replacement for the Cisco PIX, but an enhancement in technology. With the convergence of the code at version 7, the Binary and ASDM image files support both platforms.

The Cisco ASA supports the following additional features over a Cisco PIX:

- web-based VPN
- VPN load balancing
- upgradeable module slot
- CF card support
- aux port support.

Placing the SSM into the Cisco ASA provides additional services to the host device. The IPS 5.0 software can be supplied in either inline mode or as a more traditional IDS monitoring operation. The SSM device also provides a dedicated Gigabit port for out-of-band management or can be managed as a session across the backplane in a similar fashion to the way 6500 series routers managed routing and switching with IOS and CatOS.

FireWall Service Module

The Cisco FWSM provides an integrated firewall module to be installed in the 6500 and 7600 router chassis. The FWSM has been recently upgraded to run the same version of code as the PIX and ASA.

There are a number of advantages and disadvantages to using the FWSM. The device can support far more connections at faster wirespeed than either the PIX or ASA, but lacks the additional functionality that the ASA now provides, such as IDS, IPS and VPN termination. The FWSM can provide 5Gbit/s throughput and a million concurrent connections. One of the key benefits is support for 1000 virtual interfaces (256 per virtual firewall) and a maximum of 4000 VLANs allowing the unit to scale for any campus organisation. Failover can be configured in Active/Passive or Active/Active for either two cards in one chassis or a card in two chassis for full inter-chassis failover.

Versions of PIX OS

There have been many recent developments to the code that runs on the Cisco PIX. A number resulted from the release of the Cisco ASA and both code trains merged in version 7.

Some of the new features include:

- time-based ACLs
- no NAT requirement
- security contexts
- bump-in-the-wire firewall
- inspection support for FTP, ESMTP, NIS+, RPC, ICMP, H323, GTP, MGCP, RTSP
- modular policy framework
- application firewall enhancements
- Active/Active failover configuration
- SSHv2 support
- SNMP v2c support.

In the past, the Cisco PIX always required NAT configuration, which precluded the implementation of the device in many situations. The newer version of the code introduces the command **nat-control** which can be negated in the usual manner to turn off the NAT requirement.

Modular Policy Framework enhancements have been imported from QoS implementation to bring the class-map, policy-map and service-policy functions to the Cisco PIX and ASA. Changes have also been made to the syslog functionality in version 7 of the code. There are 36 configurable parameters. One of these includes the Event List function which provides a method for the aggregation of certain types of messages internally before forwarding them to syslog for processing.

Application firewall enhancements improve the filtering of content transmitted using port 80. A number of rules analyse the content to block tunnelling, peer to peer and other malformed HTTP requests.

To determine the version of firewall code running on the device, the command **show version** can be used. This command also displays information about interfaces, licenses and codes.

Firewall# sh ver

Cisco PIX Security Appliance Software Version 7.0(4)

Device Manager Version 5.0(4)

Compiled on Thu 13-Oct-05 21:43 by builders

System image file is “flash:/image”

Config file at boot was “startup-config”

Firewall up 46 mins 56 secs

Hardware: PIX-515E, 64 MB RAM, CPU Pentium II 433 MHz

Flash E28F128J3 @ 0xfff00000, 16MB

BIOS Flash AM29F400B @ 0xfffd8000, 32KB

0: Ext: Ethernet0 : address is 0017.5976.5508, irq 10

1: Ext: Ethernet1 : address is 0017.5976.5509, irq 11

2: Ext: Ethernet2 : address is 000e.0ca9.40c9, irq 11

Licensed features for this platform:

Maximum Physical Interfaces : 3

Maximum VLANs : 10

Inside Hosts : Unlimited

Failover : Disabled

VPN-DES : Enabled

VPN-3DES-AES : Disabled

Cut-through Proxy : Enabled

Guards : Enabled

URL Filtering : Enabled

Security Contexts : 0

GTP/GPRS : Disabled

VPN Peers : Unlimited

This platform has a Restricted (R) license.

Serial Number: 810000000

Running Activation Key: 0x00000000 0x00000000 0x00000000

0x00000000 0x00000000

Configuration last modified by enable_15 at 13:20:29.106 UTC

Thu Jan 4 2007

Configuration using the GUI

The Cisco PIX and ASA security appliances are provided with a graphical management tool, the ASDM. The ASDM replaces the previous graphical tool, the PIX Device Manager. The ASDM can be used to configure, manage and monitor one or more of Cisco's security appliances. It allows for five simultaneous connections, or five per context, 32 in total when using virtual firewall contexts. It requires a Java VM which allows platform-agnostic execution and the traffic is encrypted by DES or 3DES depending on the license installed on the host device.

In the home window, the ASDM provides an overview of the security appliance with general information on the device:

- hostname
- device version
- ASDM version
- firewall mode
- total flash memory
- device uptime
- device type
- context mode
- total memory.

Other panes in the window provide:

- interface status
- VPN status
- system resource allocation
- traffic status
- syslog messages.

Configuration using the CLI

The Cisco PIX and ASA security appliances are based on the command set found in Cisco IOS. They have the same 16 privilege levels and have four administrative access modes. The first access mode is unprivileged, which is shown by the right angled bracket character. This is the first access mode entered when connecting to a device via either the serial console, telnet or SSH.

Firewall>

The second access mode is privileged and is shown by the hash character. Privileged access

is enabled by entering the **enable** command and password if set.

Firewall>

Firewall> enable

Firewall#

Configuration mode allows configuration changes to be made to the system and this is shown by the **(config)** flag. Configuration access is granted by the command **configure terminal** (or **conf t**.).

Firewall>

Firewall> enable

Firewall(config)#

The fourth mode is the monitor mode, which allows password recovery and software image update in the event of a problem. Monitor mode is entered as part of the boot sequence and is recognised from the **Monitor>** prompt. **Monitor>**

There is a help system available in IOS to assist entering commands. It is context-sensitive, which allows help to be given on what is currently being entered. Typing the question mark character (**?**) gives a list of the next available options.

Connections to the security appliance can be made through various methods, each requiring slightly different operations.

Configuring a device for the first time requires the use of a serial connection to the console using terminal emulation software. All security appliances will be provided with a suitable cable, called a rollover cable, which needs to be connected to the serial port of the computer. Most modern laptops will require a USB-serial adaptor as they do not have a traditional serial port.

The terminal emulation software should be configured with the settings of 9600 8-N-1 to ensure correct communication. Windows® comes supplied with the HyperTerminal package, although users may prefer TeraTerm. UNIX and Linux systems have many different packages, including cu.

Firewall implementation

Firewall implementation

There are a number of issues that need to be considered before a new firewall is deployed or an existing one replaced on an organisation’s network.

Requirements Analysis

Defining a requirements specification will allow a successful evaluation of the various solutions available. There are many different elements involved in a firewall solution and the balance which needs to be achieved between these will differ significantly between organisations.

Interface

At a simple level, a firewall needs at least two network interfaces. Any firewall used to protect anything other than a small organisation will require more interfaces or the potential to expand the system. It should be established whether interfaces are generic and can be used for any firewall role or are specific to management of the device. The physical presentation of the interfaces will need to match those of the networking infrastructure into which the device is going to be placed. Most firewalls will have 10/100(/1000) copper Ethernet interfaces or fibre SC connectors.

DMZ

If the firewall is going to provide protection for computers within a DMZ it will need an additional interface for each DMZ network. Some firewalls will be able to support multiple DMZ interfaces and this is used increasingly to support wireless network configuration as well as traditional server farms.

Failover and load balancing

It is important to establish whether failover requires a dedicated interface, is managed by dedicated cable, or is managed inbound. As firewalls support load balancing and fault tolerance, they are one of the most critical devices in the organisation’s network infrastructure, and it is important to ensure appropriate facilities are available. The provision of basic requirements such as dual power supplies, redundant supervisors and removable fan trays can be established easily, but it is important to check if the firewall can operate in an Active/Passive or Active/Active mode of failover operation (see 3.4).

Operational Mode

Many devices will support different modes of operation (routed, NAT/PAT, bump-in-thewire), which can be configured as part of the device setup. It is important to ensure there are no limitations in the method of operation to be used.

Speed

The wirespeed throughput will determine the traffic that can traverse the firewall before it becomes a bottleneck. It is important to base calculations on the speed of the device when it is performing DPI/IDS/IPS if you are going to use these functions. Vendors will often quote performance figures for a number of scenarios, so make sure you identify the correct figures.

Authentication

If users are going to be required to authenticate before packets are allowed to traverse the firewall, how does the firewall implement authentication? It could be web based or via a special client. It is also worth considering whether the device requires a local database of credentials, or if it can query another authentication service, such as AD, Radius or LDAP.

Content filtering

It is important to establish what functions you require the firewall to provide. Simple rule-based firewalling will almost certainly be included and, in the vast majority of cases, stateful inspection. Other additional features include DPI, Intrusion Detection, Intrusion Prevention, Anti-X and VPN termination.

Virtualisation

Virtual firewall facilities can be useful to partition firewalling into different areas for isolation or to better organise Active/Active failover. What is the maximum number of virtual firewalls that can be created?

Platform

Not all firewalls are appliance based: some run as an installation on top of an installed operating system. A decision will have to be made as to whether the necessary skills are available to support the operating system as well as the firewall. If this can be achieved, which operating systems and versions can be supported by the relevant staff?

Reporting and management

The provision for reporting and management is often investigated as secondary to the technical aspects, but these will be the facilities used to interact with the device on a day-today basis after installation.

Reports need to be available in a number of formats, preferably tailored to the requirements of the individual. The device may provide an overview, alerts when something occurs or just a daily report. The interface to the logs is also important: if one has invested resources in a syslog installation, it would be illogical to procure a firewall that cannot output logs to a syslog server. Conversely, can the operating systems used to manage the network infrastructure support any applications required? Are they written in Java?

The firewall requires an interface for configuration of interfaces, rules and policies. Often the choice between GUI and CLI is a personal one, so the option of both will satisfy more staff. There may be additional requirements for the user interface. Almost certainly it needs to be secure: HTTPS instead of HTTP or SSH instead of telnet.

Documentation and support

A recent trend in the IT industry is to make product manuals available before purchase, either through pre-sales staff or the vendor website. This is an excellent way to find out about the product in more detail than is available in a datasheet. It is worth investigating training options for staff – are there vendor-run courses at different levels?

Support is essential for a mission-critical application like a firewall, so different options for maintenance will need to be discussed. Obtaining details about replacement times for parts and response times for human contact is critical, as when a fault occurs, this is the time you need to speak to someone, quickly.

If an additional operating system is involved, establish where the support burden lies, including operating system patching and maintenance.

Added value

Many firewalls offer ‘added value’ functionality: features that are included above and beyond the basic firewalling code. These may include VPN termination functions, which can be used for creating a tunnel between two firewalls or from an end user to the firewall. However, there is debate as to the security of terminating end user connections on the firewall or if they should be terminated on a dedicated VPN server within the firewall DMZ. It is important to check if these features are included in the purchase price or whether they require optional hardware upgrades or licensing.

Consultation

In any organisation there will be a number of individuals who will have concerns about any IT-related changes. A change in a firewall should include comprehensive consultation with all stakeholders.

The consultation will be different if the firewall policy is changing as opposed to a hardware or software change. If only the hardware and software are changing then the consultation will be internal, concentrating more on the technical issues like access to change rules, change control, reporting and logging and the networking implications.

If the firewall policy is to be changed then the consultation will need to be much wider. Stakeholders will come from all areas of the organisation and it is important to try to contact as many as possible during the consultation period. If the driver for change is clear and the benefit is obvious, the majority will be in support. This is all the more reason to get communication right. The one major problematic issue that is seen repeatedly is time frame: there is never a right time to change an IT system, but it has to be done some time.

Creation of Policy

The creation of a firewall policy should be carried out in conjunction with senior management within the organisation. The support of senior management is essential to ensure that IT staff can implement the policy and make decisions on firewall requests.

In the policy there should be answers to at least these key questions:

- Who is managing the policy?

- Which senior member of staff is sponsoring the policy?
- How often is the policy to be reviewed, and by whom?
- How are changes requested?
- How are disputes resolved?
- What is the Service Level Agreement?
- How is logging data managed and what is the data retention period?

As well as core decisions as to which rules will be approved and which will not, the policy may include technical elements. These may be an application version prerequisite, the location of a server on a specific netblock or a need for an individual to attend a training or briefing session.

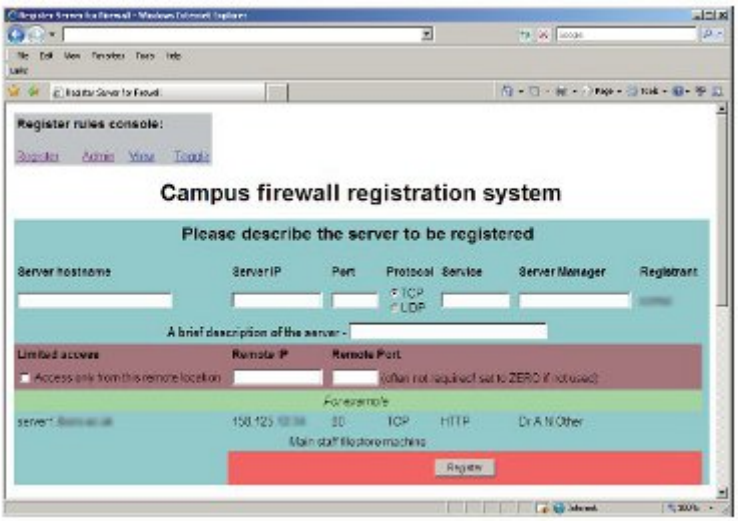
The policy should also contain details of in what circumstances rules will be removed, for example, to protect the organisation against compromised machines or in the event of an exploit being used against vulnerable services.

System Registration

For any system with more than a handful of rules, there needs to be some method to register systems for particular firewall rules. Most firewall implementations will be default deny, so the rules required will be holes in the firewall for particular services on specific hosts.

It is important to ensure that the firewall policy and guidance is available before any registration system is introduced. If no policy is in place then the registration procedure may get clogged with requests for NetBIOS ports to be opened, for example. A Frequently Asked Questions page is a useful method of communicating firewall policy, giving concrete examples of what will and will not be allowed.

The registration system could take a number of different guises, as some organisations have an open policy where IT administrators create their own rules. This might not be the most effective way of managing a firewall, as it may lead to unintentional security compromises. or online web forms to request firewall



[22]

Figure 15: Online firewall registration system

Auditing, Penetration Testing and Review

Once a new firewall system has been released into service, the protection offered will morph every time a firewall rule is changed. It would be impractical to audit the system every time a change is made, but it is prudent to periodically check the effectiveness of the firewall. Similar checks to those performed in the testing and evaluation stage (see 9.1) will ensure, as far as is possible, that the firewall is acting as an effective barrier against Internet-based attacks and threats.

Penetration testing is one way of auditing the technical aspects of the effectiveness of a firewall; however, it is worth remembering that it only tests one element of the system. Penetration tests can be performed in-house or through a commercial company. Results can vary greatly according to the technology and methodology used. A penetration test is not the only answer to the auditing issue and passing any test does not mean your organisation is secure.

Be careful to ensure when performing a penetration test that you have the permission of all necessary stakeholders for the infrastructure that you are using. It is usually best to conduct the testing on-site from a computer connected to the outside interface of the site firewall.

Conducting a penetration test across JANET or the public Internet will more than likely alert other organisations including JANET-CERT and result in wasted resources as they investigate the matter as a real attack against your organisation. However, there may be a policy that allows this within a Regional Network, so it is worth speaking with the managing agents first.

Any form of penetration testing needs to be more than just a port scan; therefore, it is recommended to seek the advice of a professional who has the knowledge not only to perform the tests but more importantly to interpret the output.

After auditing the technological aspects of the firewall system, it is worth investigating how well other procedures are working. This includes not only the monitoring of traffic problems but also whether requests for firewall changes are being actioned, recorded and checked correctly.

While there is not a definitive set of auditing requirements for firewalls, a number of controls exist in BS ISO/IEC 27001:2005 which apply to the technology used for firewall implementation.

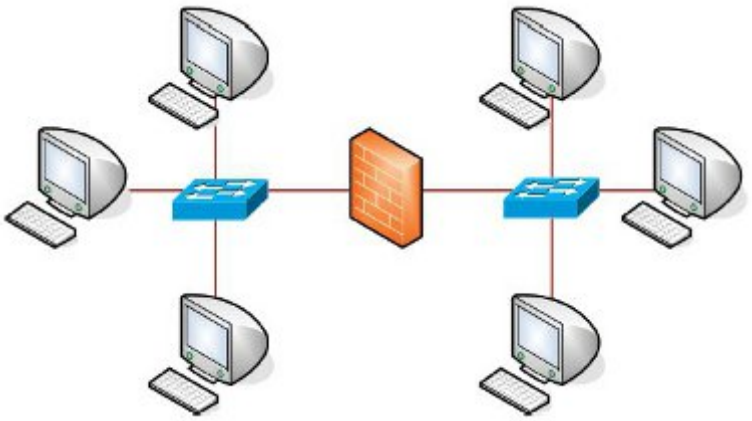
Technical consideration

Technical consideration

Testing

Before any system is introduced into the production environment, it must undergo a period of testing and evaluation. This is usually done in a test environment away from the production infrastructure.

Once the firewall has been configured according to the security policy, testing under load should be the next stage. Multiple test computers can be connected on the inside and outside interfaces of the firewall, and then load testing tools can be used to transfer large quantities of data. Netcat, Deluge, Siege and Diesel are examples of open source tools, and there are also ame task.



[23]

Figure 16: Firewall testing using multiple PCs

Once wirespeed throughput testing has been completed, it is important to test the firewall functionality using penetration testing. Both a simple port scan and a more complex penetration test of the services protected behind the firewall will show how the configuration of the firewall reacts to malicious threats. For example, it is useful to monitor the difference between dropping and resetting unwanted connections.

As a further test, PCs on the inside interface can be configured to reflect the operating systems and configurations of standard computers inside the organisation, so that streaming and other services can be tested, and the best rules for protecting the internal infrastructure designed.

Finally, load balancing and failover should be tested, to observe the results before it happens for real. Issues with tailoring heartbeat timeouts and other failover aspects can then be addressed before implementation.

Problematic Protocols

With any firewall implementation there will be problems with functionality of some protocols. There are technical solutions provided in the software by some vendors, but it is important to consider the monitoring of these protocols during installation.

These issues include:

- H.322 and related videoconferencing software: videoconferencing is often the most difficult application to implement across a firewall
- UDP: a lot of streaming services rely on a TCP connection to initiate the stream, which

is then formed of UDP packets with the destination of the host. The filtering of incoming UDP is becoming easier as the intelligence of firewalls becomes more sophisticated.

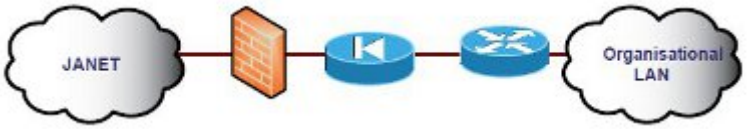
Pre-screening or Clean Feed

The increasing number of Internet threats and the demands of network traffic are proving difficult for some existing firewalls to manage successfully. An alternative to procuring a new firewall is to investigate a clean feed style technology.

Clean feed technology provides a pre-screened subset of the Internet traffic destined for the organisation. All the traffic initially passes through a first tier firewall which screens the traffic for threats and drops malicious traffic before providing a cleaner feed to the existing firewall.

Depending on the technology deployed, the first tier firewall may screen out most DoS attacks and port scans, restrict ICMP and use DPI to remove threats from the protocols that it can analyse.

The existing firewall therefore only has to manage the cleaner feed, which will have considerably lower bandwidth requirements. The firewall rules for granular control will still be applied at the existing firewall.



[24]

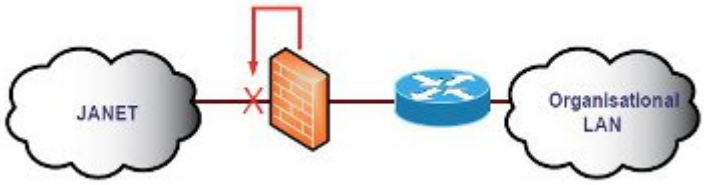
Figure 17: A first tier firewall providing a cleaner feed for an existing Cisco PIX

Integration with IDS and IPS

IDS and IPS are well established technologies that monitor network traffic for malicious content. The key difference between the two is how they react to a detected threat. An IDS will provide alerts when malicious content is identified, but human intervention is required to manage the issue. An IPS automates the mitigation to actively prevent malicious content compromising the network infrastructure.

Firewalls increasingly include IDS or IPS to complement the firewall code. They can be provided as additional software, additional hardware or fully integrated and configured by default.

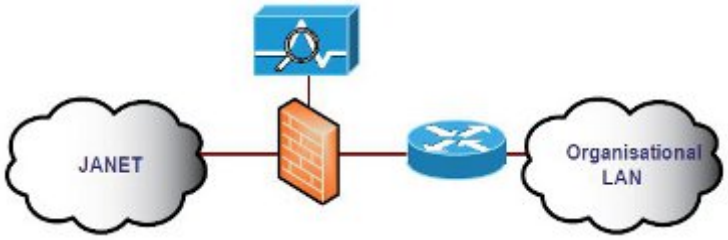
There are several different modes of operation. The system can be deployed inline so all traffic passing through the firewall is seen by the IDS/IPS and mitigated as appropriate before it is forwarded across the firewall.



[25]

Figure 18: Mitigation of malicious traffic using an IPS in inline mode on the firewall

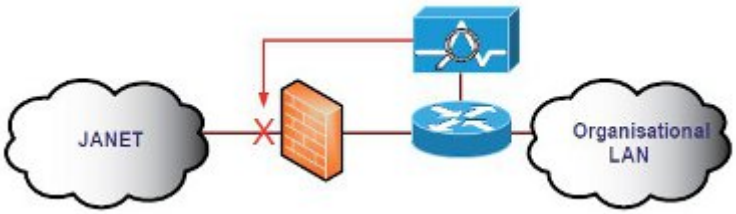
Some firewalls send all traffic received at the outside interface to the IDS/IPS as well as forwarding it according to the firewall rules. Alternatively, the IDS/IPS data can be sent to an external device for reporting.



[26]

Figure 19: Monitoring traffic using an IDS in promiscuous mode on the firewall

Often IPS-enabled firewalls can mitigate malicious traffic detected by systems elsewhere on the network. This is achieved by a secure connection that allows the firewall to issue a command or dynamically add a firewall rule. For example, on Cisco devices the **shun** command is used to mitigate traffic for a predetermined length of time.



[27]

Figure 20: An external IDS/IPS mitigating malicious traffic on the firewall

---

Links
[1] http://escarpment.net/
[2] http://community.ja.net/library/janet-services-documentation/security-matters-technical-guide
[3] http://www.internetworldstats.com/
[4] http://community.ja.net/system/files/images/firewalls-tg-01.jpg
[5] http://community.ja.net/system/files/images/firewalls-tg-02.jpg
[6] http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-87-2.pdf
[7] http://community.ja.net/system/files/images/firewalls-tg-03.jpg
[8] http://community.ja.net/system/files/images/firewalls-tg-04.jpg
[9] http://community.ja.net/system/files/images/firewalls-tg-05.jpg
[10] http://community.ja.net/system/files/images/firewalls-tg-06.jpg
[11] http://community.ja.net/system/files/images/firewalls-tg-07.jpg
[12] http://community.ja.net/system/files/images/firewalls-tg-08.jpg
[13] http://community.ja.net/system/files/images/firewalls-tg-09.jpg
[14] http://community.ja.net/system/files/images/firewalls-tg-10.jpg
[15] http://community.ja.net/system/files/images/firewalls-tg-11.jpg
[16] http://community.ja.net/system/files/images/firewalls-tg-12.jpg
[17] http://www.hmso.gov.uk/acts/acts1998/19980029.htm
[18] http://www.hmso.gov.uk/acts/acts2000/20000023.htm
[19] http://community.ja.net/library/janet-services-documentation/network-time-service

[20] http://community.ja.net/system/files/images/firewalls-tg-13.jpg
[21] http://community.ja.net/system/files/images/firewalls-tg-14.jpg
[22] http://community.ja.net/system/files/images/firewalls-tg-15.jpg
[23] http://community.ja.net/system/files/images/firewalls-tg-16.jpg
[24] http://community.ja.net/system/files/images/firewalls-tg-17.jpg
[25] http://community.ja.net/system/files/images/firewalls-tg-18.jpg
[26] http://community.ja.net/system/files/images/firewalls-tg-19.jpg
[27] http://community.ja.net/system/files/images/firewalls-tg-20.jpg