

## NISCC Technical Note 10/04: Revised 23 February 2005

### Understanding Firewalls

#### Key Points

- Is an introduction to Internet firewalls
- Describes the main types of firewall techniques
- Includes a checklist of questions that can be used to select a firewall
- A default deny rule set is recommended
- State aware firewalls with application proxies provide the highest level of security

National Infrastructure  
Security Co-Ordination Centre  
PO Box 832  
London  
SW1P 1BG

Tel: 0870 487 0748  
Fax: 0870 487 0749  
Email: [enquiries@nisc.gov.uk](mailto:enquiries@nisc.gov.uk)  
Web: [www.nisc.gov.uk](http://www.nisc.gov.uk)

## Introduction

1. This technical note is intended as an introduction to firewalls, but also explores possible security features of firewalls and their current state of implementation. It is vendor-independent and does not discuss features available on particular commercial firewalls. For reports on particular firewalls that have been through formal evaluation under the UK government's IT Security Evaluation and Certification Scheme (ITSEC) see <http://www.cesg.gov.uk/>. The intention of this note is to increase technical awareness about firewalls, and to enable potential purchasers to determine the security features most appropriate to their networks. This note is aimed at managers and security officers who wish to inform their decisions on the choice of firewalls available, and the impact of those choices in security terms.
2. This note is not intended to be exhaustive in its coverage. The following URLs link to papers that provide alternative or more detailed coverage. It should be noted, however, that the terminology for firewall types has not been standardised, and some of these papers do not agree in their usage. [1]
  - <http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf>
  - <http://www.sans.org/rr/whitepapers/firewalls/>
  - <http://www.cert.org/security-improvement/practices/p053.html>
  - [http://www.radium.ncsc.mil/tpep/library/protection\\_profiles/traffwpp.pdf](http://www.radium.ncsc.mil/tpep/library/protection_profiles/traffwpp.pdf)
  - [http://www.radium.ncsc.mil/tpep/library/protection\\_profiles/traffwmpp.pdf](http://www.radium.ncsc.mil/tpep/library/protection_profiles/traffwmpp.pdf)
  - [http://www.radium.ncsc.mil/tpep/library/protection\\_profiles/alfwpp.pdf](http://www.radium.ncsc.mil/tpep/library/protection_profiles/alfwpp.pdf)
3. To begin with: what is a firewall? A brief answer is that a firewall is a device that handles or mediates traffic flow between one network and others, performing security checks on that traffic in accordance with a predetermined security policy. If traffic fails to match the security policy, then it is not allowed through the firewall. The security policy is usually enforced by a firewall rule set, against which traffic is checked.
4. The nature of these checks will vary with the type of firewall, but let us consider what these checks might be.
  - Checks for specified content in some layer of the protocol hierarchy
  - Checks for malformed or abnormal traffic at some layer of the protocol hierarchy
  - Checks that help determine that the traffic comes from the claimed source
5. The next section can be omitted if you have basic familiarity with TCP/IP.

## Background Information on TCP/IP

6. In this note, we are concentrating on TCP/IP firewalls because most networks use TCP/IP. Alternative networking protocols still in use are Microsoft's NetBEUI, Hewlett Packard's DECnet protocol family and Novell's IPX/SPX. However, the Internet makes use of TCP/IP, and most firewalls only support TCP/IP. To understand what a TCP/IP firewall can do, it is necessary to understand how TCP/IP works.
7. IP (Internet Protocol) [2] is the networking layer of the TCP/IP protocol suite. Above IP are transport layer protocols such as UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). Above TCP and UDP are application protocols such as HTTP (HyperText Transfer Protocol) used to transmit web pages across the World Wide Web.
8. Before they are sent, the layers are nested inside one another by the computer's TCP/IP stack, with the lowest outermost. The process of nesting one layer inside another is called encapsulation, so that lower layers of the protocol stack may be said to encapsulate the higher layers. For example, IP packets may contain TCP packets, which in turn may contain HTTP packets. The IP packet is split into units of physical transmission in the format of the data link protocol (Ethernet for example) and transmitted through the physical cabling. When the packet is received the packets are unwrapped, outmost layer first as they are passed up the TCP/IP stack on the receiving computer.
9. IP is designed for end-to-end routing across networks. It includes addressing information (the source and destination "IP address"), details of the transport layer protocol encapsulated in it and information on fragmenting IP packets into smaller packets and subsequent reassembly. IP itself does not provide any means of implementing a persistent connection or of checking whether packets have not arrived. It should be stressed that it is the responsibility of higher (transport) level protocols to ensure correct, error free communication.
10. IP has a control protocol, the Internet Control Message Protocol (ICMP), that is used to send troubleshooting, device control and management information on an IP network. A standard IP header is used to encapsulate ICMP packets, but because ICMP is used to report errors in IP packets, the ICMP packet will sometimes include IP headers belonging to those IP packets. ICMP is not designed to be absolutely reliable.
11. By contrast to IP and ICMP, TCP, which is a transport layer protocol, provides a reliable, connection-based service. TCP achieves this by:
  - Assigning numbers, called "ports", on the sending and receiving computer for the particular network service
  - Providing an initial connection establishment three-way handshake
  - Providing sequence numbers for all TCP packets so that they can be acknowledged and reassembled in the correct order on receipt
  - Providing acknowledgements of each packet received after that

12. IP and TCP packets contain flags that they use to identify the type of packet. Flags in IP are used to specify whether the IP packet needs to be broken up into smaller packets in order to be transmitted and whether the fragment is the last one. In the case of TCP, the initial TCP packet of a session will have the SYN (synchronise) flag set to request a connection as the first step in the handshake that establishes the connection, and the client will send a TCP packet with the FIN (final) flag set to request the server to close the connection. The other TCP flags are ACK (acknowledgement), RST (reset), URG (urgent) and PSH (push). ACK is used to acknowledge requests, RST is used to reset a TCP connection, URG is used to request that a packet is processed out of sequence, and PSH is used to push a packet directly to the application layer.
13. The state of a TCP connection is determined by the flags set in the packets and by the client and server sequence numbers. For example, to initiate a TCP connection the client sends a TCP packet with the SYN flag set (a so-called "SYN packet") and the client's sequence number,  $x$  say. In response, if the destination port is open, the server will send a packet with the SYN and ACK flags set (a so-called SYN/ACK packet) with its own sequence number,  $y$  say, and an acknowledgement number (the next expected sequence number from the client,  $x+1$ ). The client will then send back a packet with the ACK flag set with sequence number  $x+1$  and acknowledgement number  $y+1$ . This exchange, the three-way TCP handshake, is used to synchronise the sequence numbers in the session. To be successful the handshake requires particular flags to be set in a definite sequence. If the sequence began instead with a SYN packet with, for example, an ACK response rather than a SYN/ACK response, then the client would not send an ACK packet but would either timeout or close the connection. In the case of a successful connection subsequent TCP data packets should always have the ACK flag set until the connection is terminated. To terminate a connection normally, the client will send a FIN packet (with the ACK flag set to acknowledge the previous packet), which will receive an ACK response; and the server will then send a FIN packet (again with the ACK flag set), which will also receive an ACK response.
14. The other common transport layer protocol, UDP, although it uses "ports", does not establish connections by setting up and tracking packets: it is an unreliable, best effort, connectionless protocol.
15. At the highest level of the protocol stack are the application protocols. For example, HTTP is a connection-oriented application protocol, and as such runs over TCP. HTTP is a standard for downloading and uploading content to a web server. Its commands, such as GET and POST, reflect this purpose. Other common application protocols include SMTP (an email transport protocol) and DNS (Domain Name Service, which is used to map IP addresses to domain names). SMTP, like HTTP is connection-oriented, and runs over TCP, while individual DNS lookups do not require a connection and so use UDP.

## Firewall Technologies

16. Let us examine how, and to what extent, the three types of security check made by firewalls identified in the introduction are currently implemented.
17. It is assumed in the following that all firewalls under consideration have multiple network interface cards. Each network interface card will support at least one network, although some network interface cards support multiple independent networks. For simplicity it is further assumed that each type of network (“internal”, “external”, “DeMilitarised Zone (DMZ)”) will have physically distinct network cards.
18. It should be noted that the firewall types below are “pure” types of firewall. In practice most firewalls mix different technology types in order to maximise security and minimise loss of throughput.

### Checks for specified content in some layer of the protocol hierarchy

#### *Packet filter firewalls*

19. The most basic of firewalls will be able to accept or deny IP packets based on a set of rules relating to current IP and TCP headers. Typical rule sets relate to:
  - Source and destination ports for inbound and outgoing traffic
  - Source and destination IP addresses
  - Transport layer protocol used
  - IP options (loose/strict source routing)
20. Firewalls of this kind act as a type of router: they check the IP packets against the rule set and forward the packet to another network interface of the firewall if the packet is accepted by the rule set. These firewalls are known as *packet filter* firewalls. Packet filters check traffic primarily at the network (IP) layer. This can be shown schematically as follows (with relevant layers greyed, with darker grey indicating more checking):

Application layer
Transport layer - TCP/UDP headers (port numbers only)
Network layer - IP header (IP addresses, protocol, options)
Data link layer

#### *Checking destination ports*

21. Your organisation should have a policy for the network services that it wishes to offer across a network boundary. Network services are assigned standard ports (a web server, for example, usually runs on TCP port 80) [3], although the assignment of ports is a matter of convention that is not followed by

everyone. By blocking access to all destination ports on your network servers, except those that are necessary for your business, you can help enforce a sound organisational security policy. Similarly, the organisational policy may restrict staff access to services on other networks, which could be achieved by only allowing access to specified destination ports for outgoing traffic. Restricting outbound connections to certain ports will also help prevent some Trojan Horse programs from communicating to a remote computer (if the Trojan uses a port that is blocked). Of course, some more sophisticated Trojans will be able to detect which ports are allowed through the firewall and can even embed messages in a protocol that a particular allowed service uses.

22. NISCC recommends a default deny policy: any network service that is not a business requirement should be blocked. This applies to all of the IP and TCP header fields that are subject to filtering, but to IP addresses and port numbers in particular. Logging all denied traffic is also recommended.

#### *Checking source ports*

23. Checking source ports is less useful than checking destination ports because source ports tend to be assigned dynamically to non-standard ports for client services (for example, a web browser). However, it is a good idea to configure the firewall to deny and log traffic from source ports used by common servers (typically port numbers less than 1024) that your organisation does not need for incoming traffic, as common source ports can be often be used to exploit problems in firewall rule sets. Moreover, as incoming traffic from allowed server ports can be suspicious if it is not part of an established session, this type of traffic should also be blocked and logged.

#### *Checking IP addresses*

24. It may be part of the organisational policy to block access from or to certain IP addresses (if they have been “blackholed” for some reason). Likewise you may wish to restrict access to some network services on certain of your network’s servers to specified external computers. This type of access control could be enforced by allowing access between specific IP addresses (for specific network services). It should be borne in mind, however, that this type of security check will not be sufficient by itself without further authentication because it is possible to forge IP addresses and to hijack TCP sessions [4]. Internal IP addresses should not arrive on an external interface of the firewall, and so should be filtered, as should private or reserved internet addresses such as 10.x.x.x or 192.168.x.x (see RFC 1918 and RFC 3330 for a complete list of private IP addresses). Similarly external IP addresses should not arrive on an internal interface of the firewall, and should be blocked. If your organisation is using private IP addresses, your firewall will need to perform network address translation to make the traffic routable on public networks.

### *Checking the protocol*

25. You would also expect even the most basic firewall to be able to filter on protocol used (for example UDP), with the capability to check for protocols such as ICMP and the OSPF (Open Shortest Path First) routing protocol. If a firewall is used as an external gateway to other networks (with a dedicated external router), it should not allow through unnecessary data, including routing information.
26. It is also important to control the types of ICMP (Internet Control Message Protocol) packets allowed into and out of an internal corporate network. Type 0 (“echo reply”), type 3 (“destination unreachable”) and type 11 (“time exceeded”) should be allowed inbound in response to outgoing IP packets and echo requests (ICMP type 8). Other ICMP types should be blocked unless there is a business case for permitting them. More generally, there are some ICMP types which should not be allowed across an external boundary without a high degree of trust existing between the networks. These should include ICMP packets of types 5 (router redirects), and 12 (parameter problems).
27. Generally, unnecessary protocol types should be blocked because they could be used to change the configuration of your internal network, for example by changing a route, to identify or map your network, or to communicate with malicious software that has previously been installed on your network.

### *Checking the IP options field*

28. Basic firewalls should also be able to check that the options field of an IP packet does not specify strict or loose source routing. A strict source route would specify the path taken by the IP packet, while loose source routing specified a number of router addresses that a response packet must pass through. Such options should be blocked by your organisation’s external border router or firewall because, if the attacker had access to a particular router, he or she could forge a source IP address and intercept packets on the route back to the forged IP address.

### *Static packet filters*

29. Packet filters are either static or dynamic. “Static” refers to the fact that the firewall uses a fixed rule set (sometimes known as “an access control list”, especially in connection with routers) to accept or deny IP packets. In addition to basic packet filtering some packet filter firewalls allow the firewall administrator to configure rules to filter on flags in TCP headers or to allow access only from certain data link layer Media Access Control (MAC) addresses [5] (also known as the “physical” addresses of a network interface card). As an example of the former, you might configure the firewall to block packets with only the SYN flag set from a computer outside your network in order to ensure that all sessions are initiated from the internal network. As for the use of MAC address filtering, you may wish to restrict access to particular computers which use dynamically allocated IP addresses (although it is

possible to forge the MAC address address of a network interface). With maximum functionality, the TCP/IP stack of a static packet filter could be shown schematically as follows:

Application layer
Transport layer - TCP/UDP headers (port numbers, TCP flags)
Network layer - IP header (IP addresses, protocol, options)
Data link layer (MAC address)

30. Static packet filters are sometimes deployed on dedicated routers. Routers with packet filtering functionality are sometimes called *screening routers*.
31. Static packet filters have the disadvantage that they cannot cope easily with programs that dynamically allocate ports as the firewall rule set cannot be adjusted to open and close these ports unless the ports are left open permanently. Static packet filters typically provide only minimal checking of the transport layer, so they can pass on, or be subject to attack by, maliciously crafted TCP or UDP packets.

#### *Dynamic packet filters*

32. A dynamic packet filter is a firewall that has a firewall that uses a variable or dynamic rule set. Dynamic packet filters typically vary their rule set based on the state of a TCP connection. The need for a dynamic rule set can be illustrated by the requirements of FTP (File Transfer Protocol). In active mode FTP (File Transfer Protocol) opens an unprivileged port (higher than 1024) on the client and then two privileged ports on the server, a control port (TCP port 21) and a data port (TCP port 20); but the server then opens a data connection to a second, unprivileged port on the client. A dynamic packet filter should allow such connections from server to client in the case of FTP (which it can recognise by destination port 21), but block such a connection to a privileged port (1024 or lower) or block any server to client connection establishment in the case of most other protocols (such as HTTP).
33. The key point about dynamic packet filters is that they have the ability to analyse the flags and sequence numbers in a TCP packet and will deny any packets that do not correspond to a legitimate established connection. As far as UDP is concerned, as UDP is stateless, dynamic filtering is of limited applicability to protocols that use UDP. In the case of UDP, only the exchange of packets between client and server can be tracked. The TCP/IP stack of a dynamic packet filtering firewall can be represented schematically as follows:

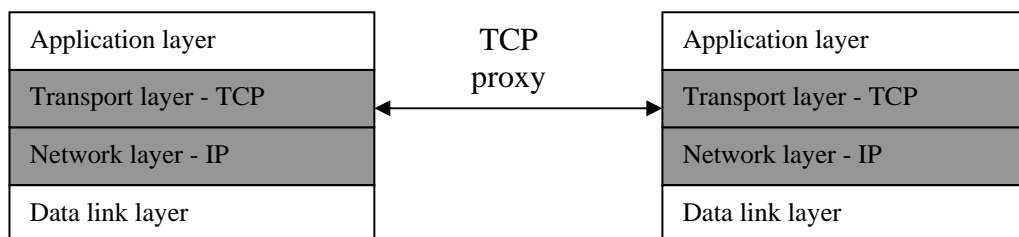
Application layer
Transport layer - TCP/UDP headers (port numbers, flags and sequence numbers)
Network layer - IP header (IP addresses, protocol, options)
Data link layer

### State Aware Firewalls

34. Dynamic packet filters are a type of state aware firewall, but it is possible that the dynamic rule set is not only determined by the state of any TCP connection but also by checking for certain patterns in the whole IP packet (in the application layer in particular), thus providing a check on the state of application layer protocol. This type of checking can enhance the performance of the firewall but it can also result in traffic being blocked incorrectly (“false positives”) and traffic not being blocked when it should be (“false negatives”) as well as potentially introducing vulnerabilities in the firewall software itself. In this technical note this generalised packet checking will be called state awareness, although it should be recognised that many authors use stateful inspection and dynamic packet filters as synonyms of state aware firewall. It should be noted that the term “deep packet inspection” has also recently been introduced specifically to describe the checking for attack patterns or anomalies in all layers of the IP packet [6].

### Circuit level proxies

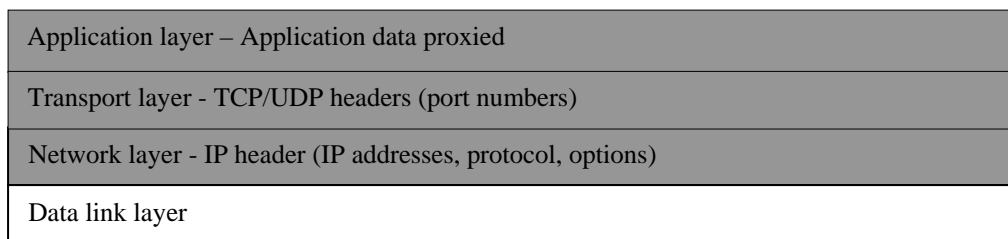
35. A state aware firewall that provides an alternative to a dynamic packet filter is a transport layer, or circuit-level, proxy. A proxy is a service that runs between firewall network interfaces and passes information between those interfaces. When two users in different networks wish to communicate using a TCP based protocol, a firewall that establishes separate connections between the firewall and the users and then forwards data between the two connections is called a *circuit level proxy* firewall. The circuit level proxy will check that the TCP sessions are correctly established and maintained, and protects the internal network from malicious TCP packets. Circuit level proxies are usually used on firewalls that also use application proxies (see below). Although circuit proxies will reject unexpected TCP packets, they slow down the connection and break the client/server model (“end-to-end connection”) of TCP. They sometimes also require specially modified client software.



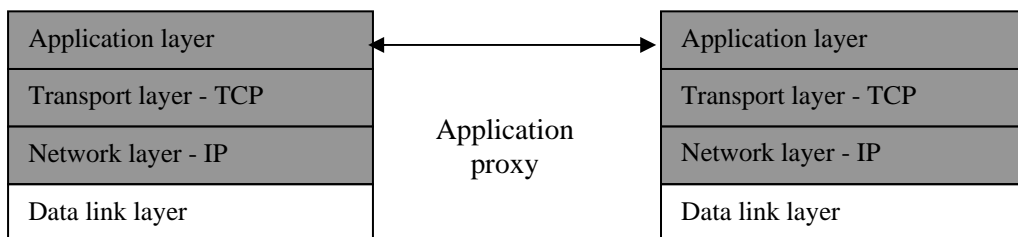
### Application proxy firewalls

36. Apart from state aware firewalls, all of the firewall types considered so far do not check the application data. Malicious application data is the most significant source of remote exploits, exploiting input validation errors and buffer overflows.

37. The most common way to check application data is to provide a proxy for the application. Proxying is primarily used to restrict or make more secure commands that can be executed on computer in the internal network. In the case of HTTP for example, it may be possible to restrict access by web address (the URL) or to restrict access to certain commands, for example the PUT command that writes data to the web server. Proxy firewalls intercept the application data as it comes in through a network interface and copy it to another network interface. This makes it possible to recreate only the commands and data that the proxy determines to be legitimate. The TCP/IP stack of a typical application proxy firewall (that also provides packet filtering) can be represented schematically as follows:



38. The proxy nature of an application proxy firewall is indicated schematically as follows:



39. As can be seen from the diagrams above, when proxy firewalls are combined with state aware firewalls, a high level of security can be achieved because the firewall will check all layers of the TCP/IP stack above the data link layer. However, proxy firewalls tend to restrict throughput because of the checking of, and possibly rewriting of, the application layer packets. They also require a securely written proxy for each network service allowed through the firewall.

### Checks for malformed or abnormal traffic at any layer of the protocol hierarchy

40. Certain types of firewall can check for certain types of unexpected packets. State aware firewalls can block traffic that is not expected as part of the currently available sessions. Proxy firewalls will not pass on data that does not comply with the standard for the application protocol being proxied, but they will not in general identify a buffer overflow unless the overflow violates the specification in the standard. It is also possible that a state aware firewall will identify malformed or abnormal packets (including some buffer overflows), but this depends on the sophistication of the packet inspection algorithm used.

41. Firewalls tend not to provide filtering on the data content (for example unsuitable domain names in HTTP packets), although there is an increasing trend for plug in products to be available for content filtering.
42. The extent to which firewalls block malformed packets is less clear. Some firewalls are liable to crash when they receive such packets. An example is the use of fragmented IP packets with the fragment offset in the IP packets set to overlap. Many implementations of TCP/IP stacks have been liable to crash when trying to reassemble overlapping fragmented packets. Ideally the firewall's TCP/IP stack should be able to check values for each field in traffic packets and reject those that contain values that could result in a packet that does not comply with the protocol standard. This checking would include testing overlapping packets. However, the more checking that is performed on traffic the more the throughput is potentially restricted.
43. It is possible to use artificial intelligence techniques to identify what count as normal packets in a data stream. A neural network could be trained to recognise normal traffic and to reject any abnormal traffic. The limitations of this approach are that the approach is not rule-based, and so cannot provide reasons why a packet was rejected, and that neural networks are liable to reject some legitimate packets.

#### **Checks that the traffic is coming from the claimed source**

44. There are several techniques that can be used to check that a packet is coming from the claimed source. They are as follows:
  - Checking that the IP address is received on the appropriate network interface
  - For traffic that includes domain names (such as email), performing a reverse name lookup
  - Providing authentication of the IP address; and
  - Using authentication services at the application level for connection based services (such as FTP and telnet) to establish trust in the user's identity
45. To prevent external computers claiming to have an internal IP address, many firewalls check that an internal IP address is not the source IP address of a packet received on the external interface. This security measure can be reinforced by configuring the firewall to hide or obscure the IP addresses of computers on the internal network. This technique is known as "Network Address Translation" (NAT), where it is common for a lookup table to be used by the firewall which maps internal addresses to external addresses and (if the number of internal addresses is greater than the number of external addresses available) port numbers to port numbers assigned by the firewall (which encode a reference to the internal IP address). Proxy servers can also be used to hide addresses on the internal network, which prevents attackers from mapping the computers on the internal network. The main disadvantage of NAT is that it slows down the connection as all the IP headers have to be rewritten. The internal system can be further hidden by using port translation (which may be used by NAT in any case), so that the identity of the actual port

used on the server is not disclosed to external system users. Port translation is often provided by using circuit level proxies.

46. For many Internet services, a domain name can be used instead of an IP address. It is easy to forge a domain name. Some application proxy firewalls check that the IP address matches the domain name by looking at the domain name corresponding to the IP address. This technique can be used in email for example to prevent “spam” mail sent with a forged domain name. Although a useful check, matching the IP address with the domain name is not an absolute guarantee of the authenticity of the domain name. It is sometimes possible for an attacker to change domain name records in a domain name server’s cache to reference incorrect IP addresses.
47. It is also possible to incorporate authentication into IP packets. The best known standard is IP Security, also known as IPSEC, which provides an additional authentication header that can be incorporated into all current versions of IP (ie versions 4 and 6). The authentication header contains a unique sequence number and authentication data, which by default is a unique number (called a “hash” or “digest”) corresponding to the invariant fields of the IP header and a shared secret key [7]. The authentication header provides authentication services that identifies the sender of the packet, and checks the integrity of the IP packet. IPSEC also provides network layer confidentiality by means of the Encapsulating Security Payload (ESP) [8]. IPSEC is implemented on some firewalls to provide a virtual private network across public or shared networks. The main disadvantages of IPSEC are that it is a complex protocol that can be difficult to configure, and that IPSEC by itself cannot provide fine-grained user or subject based security where more than one user uses a computer.
48. Proxy based firewalls also usually offer the capability of adding additional layers of authentication for applications. This authentication is commonly based on a password, a response to a challenge or a digitally signed public key certificate.

## **Firewall Security**

49. An important consideration with firewalls is the security of the firewall itself. Firewalls should only function as firewalls [9]; additional services are likely to make the firewall susceptible to attack. The firewall operating system should be designed to be secure. The TCP/IP stack should have been hardened, rewritten if necessary; and the use of mandatory access control functionality to separate the operating system from the proxies is a good way to provide security in depth. This is essential in case one of the proxies is discovered to have a vulnerability.
50. It is good practice to deploy a screening router to protect your firewall and to route traffic to the firewall. A screening router can be used to provide basic packet filtering and to decrease the load on the firewall, but it may have poor logging and be difficult to configure correctly. It is good practice to place shared or public servers on a separate network (called a DeMilitarised Zone,

DMZ for short) connected through a third network interface on the firewall. However, to protect your firewall and your network, a defence in depth architecture can provide increased assurance. The idea of defence in depth is to deploy different firewalls with distinct implementations connected in series to make it more difficult for attackers to compromise the firewalls and reach the internal network.

## Conclusions

51. From the discussion of types of firewalls in this paper, it is clear that not all types of firewall offer the same level of protection to your network.
52. Basic packet filter firewalls provide some protection against attackers gaining access to network services that you wish to restrict to the internal network or to specific external IP addresses. If your organisation is using a packet filter firewall, it should have the capability to block external IP addresses that claim to be internal addresses. You should also not rely on external IP addresses for authentication. The use of authentication for all services offered to particular external IP addresses is recommended.
53. State aware firewalls provide some protection of your internal network against unexpected packets (such as connections from source ports providing allowed network services) and against certain types of crafted packets (such as TCP SYN/FIN packets).
54. Proxy firewalls provide checking of application level packets. They should make the application proxied more secure and may provide additional authentication. Proxy firewalls may also provide checking on application level packets.
55. The use of NAT is worth considering as it enables the internal network addresses to be translated or hidden in communications with external networks.
56. It is worth noting that firewalls are only as secure as the least secure network service that is allowed through. Your organisation's firewall may help make that service secure, but there is no guarantee of that. Use of an externally available network service with known vulnerabilities on a server that has not been configured with security in mind is likely to lead to that server being compromised.

## Checklist

57. A checklist of the issues raised in this note is listed below. The issues concern firewall functionality and do not relate to any assurance measures that the functionality has been implemented correctly.

QUESTION
Does the firewall support IP v6?

Does the firewall provide basic packet filtering?
Is the firewall state aware?
If the firewall is state aware, does the firewall use signature based or anomaly based packet inspection?
Does the firewall provide dynamic packet filtering?
Does the firewall provide a circuit level proxy?
Does the firewall have high availability?
Does the firewall provide application proxies?
Do the application proxies cover all network services that you need to allow through the firewall?
Do the application proxies counter vulnerabilities in the application data?
Do the application proxies provide additional authentication?
Does the firewall support IPSEC?
Does the firewall provide checks on IP address forging?
Does the firewall provide NAT?
Has the firewall operating system been designed to be secure?
Has the firewall operating system been hardened?
Has the firewall TCP/IP stack been hardened/rewritten?
Does the firewall operating system use mandatory access control?
Does the firewall integrate with other security products (eg intrusion detection systems or anti-virus)?
Does the firewall support logging in a standard format (eg syslog)?
Does the firewall support DMZs?

## Notes

[1] UK government readers should refer also to: CESG Infosec Memorandum No.13; CESG Infosec Manuals M and P on protecting government connections to the Internet; Infosec Standard No.3 Connecting Business Domains; and Infosec Standard No.1 Residual Risk Assessment Method.

[2] All references to IP in this technical note refer to IP version 4. The general principles also apply to IP version 6, but the standards for IP packets and for the control protocol, ICMP, are different, defined in RFCs 2460 and 2463.

[3] See <http://www.iana.org/assignments/port-numbers>.

[4] By default IP addresses are not authenticated. It is possible to claim to be any IP address. Some addresses will be discarded by routers because there are non-routable reserved IP addresses or have special significance (like "0.0.0.0"). It is also possible, although more difficult, to intercept traffic and pass on your own data instead. This is a "man-in-the-middle" attack. Equally it is possible to send forged UDP packets or to hijack one end of the TCP handshake if the IP address you are claiming to be can be temporarily kept busy. These vulnerabilities in TCP/IP are discussed in Bellovin's classic paper "Security Problems in the TCP/IP Protocol Suite", which is available at [http://www.ja.net/CERT/Bellovin/TCP-IP\\_Security\\_Problems.html](http://www.ja.net/CERT/Bellovin/TCP-IP_Security_Problems.html).

[5] Filtering MAC addresses is possible only if the firewall is on the same local area network as the connecting machine, as routers will strip out the data link layer.

[6] See <http://www.securityfocus.com/infocus/1817>.

[7] See <http://www.ietf.org/rfc/rfc2402.txt>.

[8] See <http://www.ietf.org/rfc/rfc1827.txt>.

[9] Personal firewalls are also a useful protective measure for end-user systems.