                     Behavior of and Requirements for
                            Internet Firewalls

Status of this Memo

Copyright Notice

Abstract

   This memo defines behavioral characteristics of and interoperability
   requirements for Internet firewalls. While most of these things may
   seem obvious, current firewall behavior is often either unspecified
   or underspecified and this lack of specificity often causes problems
   in practice. This requirement is intended to be a necessary first
   step in making the behavior of firewalls more consistent across
   implementations and in line with accepted IP protocol practices.

1. Introduction

   The Internet is being used for an increasing number of mission
   critical applications. Because of this many sites find isolated
   secure intranets insufficient for their needs, even when those
   intranets are based on and use Internet protocols. Instead they find
   it necessary to provide direct communications paths between the
   sometimes hostile Internet and systems or networks which either deal
   with valuable data, provide vital services, or both.

   The security concerns that inevitably arise from such setups are
   often dealt with by inserting one or more "firewalls" on the path
   between the Internet and the internal network. A "firewall" is an
   agent which screens network traffic in some way, blocking traffic it
   believes to be inappropriate, dangerous, or both.

   Note that firewall functions are disjoint from network address
   translation (NAT) functions -- neither implies the other, although
   sometimes both are provided by the same device. This document only
   discusses firewall functions.

1.1. Requirements notation

This document occasionally uses terms that appear in capital letters.
When the terms "MUST", "SHOULD", "MUST NOT", "SHOULD NOT", and "MAY"
appear capitalized, they are being used to indicate particular
requirements of this specification. A discussion of the meanings of
these terms appears in RFC 2119 [2].

2. Characteristics

Firewalls either act as a protocol end point and relay (e.g., a SMTP
client/server or a Web proxy agent), as a packet filter, or some
combination of both.

When a firewall acts a protocol end point it may

   (1)   implement a "safe" subset of the protocol,

   (2)   perform extensive protocol validity checks,

   (3)   use an implementation methodology designed to minimize
          the likelihood of bugs,

   (4)   run in an insulated, "safe" environment, or

   (5)   use some combination of these techniques in tandem.

Firewalls acting as packet filters aren't visible as protocol end
points. The firewall examines each packet and then

   (1)   passes the packet through to the other side unchanged,

   (2)   drops the packet entirely, or

   (3)   handles the packet itself in some way.

Firewalls typically base some of their decisions on IP source and
destination addresses and port numbers. For example, firewalls may

   (1)   block packets from the Internet side that claim a source
          address of a system on the internal network,

   (2)   block TELNET or RLOGIN connections from the Internet to the
          internal network,

   (3)   block SMTP and FTP connections to the Internet from internal
          systems not authorized to send email or move files,

(4)   act as an intermediate server in handling SMTP and HTTP
       connections in either direction, or

(5)   require the use of an access negotiation and encapsulation
       protocol such as SOCKS [1] to gain access to the Internet, to
       the internal network, or both.

(This list of decision criteria is only intended to illustrate the
sorts of factors firewalls often consider; it is by no means
exhaustive, nor are all firewall products able to perform all the
operations on this list.)

3. Firewall Requirements

   Applications have to continue to work properly in the presence of
   firewalls. This translates into the following transparency rule:

      The introduction of a firewall and any associated tunneling or
      access negotiation facilities MUST NOT cause unintended failures
      of legitimate and standards-compliant usage that would work were
      the firewall not present.

   A necessary corollary to this requirement is that when such failures
   do occur it is incumbent on the firewall and associated software to
   address the problem: Changes to either implementations of existing
   standard protocols or the protocols themselves MUST NOT be necessary.

   Note that this requirement only applies to legitimate protocol usage
   and gratuitous failures -- a firewall is entitled to block any sort
   of access that a site deems illegitimate, regardless of whether or
   not the attempted access is standards-compliant. This is, after all,
   the primary reason to have a firewall in the first place.

   Also note that it is perfectly permissible for a firewall to provide
   additional facilities applications can use to authenticate or
   authorize various sorts of connections, and for the firewall to be
   configurable to require the use of such facilities. The SOCKS
   protocol [1] is one example of such a facility. However, the
   firewall MUST also allow configurations where such facilities are not
   required for traversal.

3.1. Examples

   The following sections provide some examples of how the transparency
   rule actually applies to some specific protocols.

3.1.1. Path MTU Discovery and ICMP

   ICMP messages are commonly blocked at firewalls because of a
   perception that they are a source of security vulnerabilities. This
   often creates "black holes" for Path MTU Discovery [3], causing
   legitimate application traffic to be delayed or completely blocked
   when talking to systems connected via links with small MTUs.

   By the transparency rule, a packet-filtering router acting as a
   firewall which permits outgoing IP packets with the Don't Fragment
   (DF) bit set MUST NOT block incoming ICMP Destination Unreachable /
   Fragmentation Needed errors sent in response to the outbound packets
   from reaching hosts inside the firewall, as this would break the
   standards-compliant usage of Path MTU discovery by hosts generating
   legitimate traffic.

   On the other hand, it's proper (albeit unfriendly) to block ICMP Echo
   and Echo Reply messages, since these form a different use of the
   network, or to block ICMP Redirect messages entirely, or to block
   ICMP DU/FN messages which were not sent in response to legitimate
   outbound traffic.

3.1.2. SMTP Extensions

   The original SMTP protocol [4] didn't provide a mechanism for
   negotiating protocol extensions. When this was added [5], some
   firewall implementations reacted by simply adding the EHLO command to
   the list of accepted commands. Unfortunately, this is not
   sufficient: What is necessary is for the firewall to scan the list of
   EHLO responses and only allow the ones the firewalls understands
   through. If this isn't done the client and server can end up
   agreeing to use an extension the firewalls doesn't understand, which
   can then lead to unnecessary protocol failures.

4. Application Requirements

   Firewalls are a fact of life that application protocols must face.
   As such, application protocols SHOULD be designed to facilitate
   operation across firewalls, as long as such design choices don't
   adversely impact the application in other ways. In addition,
   application protocol specifications MAY include material defining
   requirements firewalls must meet to properly handle a given
   application protocol.

Examples of proper and improper application protocol design include:

(1)   Wrapping a new protocol around HTTP and using port 80 because
       it is likely to be open isn't a good idea, since it will
       eventually result in added complexity in firewall handling of
       port 80.

(2)   Defining a secure subset of a protocol is a good idea since it
       simplifies the firewall design process.

(3)   Specificating an appropriate firewall traversal mechanism if
       one exists is a good idea.

(4)   Registering a separate port for new protocols is a good idea.

5.   Security Considerations

   Good security may occasionally result in interoperability failures
   between components. This is understood. However, this doesn't mean
   that gratuitous interoperability failures caused by security
   components are acceptable.

   The transparency rule impacts security to the extent that it
   precludes certain simpleminded firewall implementation techniques.
   Firewall implementors must therefore work a little harder to achieve
   a given level of security. However, the transparency rule in no way
   prevents an implementor from achieving whatever level of security is
   necessary. Moreover, a little more work up front results in better
   security in the long run. Techniques that do not interfere with
   existing services will almost certainly be more widely deployed than
   ones that do interfere and prevent people from performing useful
   work.

   Some firewall implementors may claim that the burden of total
   transparency is overly onerous and that adequate security cannot be
   achieved in the face of such a requirement. And there is no question
   that meeting the transparency requirement is more difficult than not
   doing so.

   Nevertheless, it is important to remember that the only perfectly
   secure network is one that doesn't allow any data through at all and
   that the only problem with such a network is that it is unusable.
   Anything less is necessarily a tradeoff between usability and
   security. At present firewalls are being circumvented in ad hoc ways
   because they don't meet this transparency requirement and this
   necessarily weakens security dramatically. In other words, the only

reason that some firewalls remain in use is because they have
essentially been disabled. As such, one reason to have a
transparency requirement is to IMPROVE security.

6. Acknowlegements

   Bill Sommerfeld provided the text for the Path MTU Discovery example.
   This document has benefited from discussions with a number of people,
   including but not limited to: Brian Carpenter, Leslie Daigle, John
   Klensin, Elliot Lear, and Keith Moore.

7. References

   [1] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D. and L.
        Jones, "SOCKS Protocol Version 5", RFC 1928, April, 1996.

   [2] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement
        Levels", BCP 14, RFC 2119, March 1997.

   [3] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191,
        November 1990.

   [4] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821,
        August 1982.

   [5] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker,
        "SMTP Service Extensions", STD 10, RFC 1869, November 1995.

8. Author's Address

   Ned Freed
   Sun Microsystems
   1050 Lakes Drive
   West Covina, CA 91790
   USA

   Phone: +1 626 919 3600
   Fax: +1 626 919 3614
   EMail: ned.freed@innosoft.com

9. Full Copyright Statement

Acknowledgement