Operations Area Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: March 18, 2016

F. Gont
SI6 Networks / UTN-FRH
F. Baker
Cisco Systems
P. Hoffman
VPN Consortium

September 15, 2015

On Firewalls in Internet Security
draft-gont-opsawg-firewalls-analysis-00

Abstract

This document analyzes the role of firewalls in Internet security, and suggests a line of reasoning about their usage. It analyzes common kinds of firewalls and the claims made for them.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The IETF has a long and fractured discussion on security. Many early RFCs simply didn't address the topic - and said as much. When the IESG started complaining about that, it was told that there was no market interest in the topic that was measurable in money spent. Those who *were* interested in the topic set forth frameworks, rules, and procedures without necessarily explaining how they would be useful in deployment, and dismissed questions as "from those who don't understand." In many cases, as a result, deployments have been underwhelming in both quantity and quality, and the Internet is noted for its problems with security. What is clear is that people need to think clearly about security, their own and that of others. What is not clear is how to do so in a coherent and scalable manner.

Prophylactic perimeter security in the form of firewalls, and the proper use of them, have been a fractious sub-topic in this area. This document suggests a line of reasoning about the use of firewalls, and attempts to end the bickering on the topic, which is, for the most part, of little value in illuminating the discussion. It also analyzes common kinds of firewalls and the claims made for them.

2. Terminology

In this document, a firewall is defined as a device or software that imposes a policy whose effect is "a stated type of packets may or may not pass from A to B". All modern firewalls allow an administrator to change the policies in the firewall, although the ease of administration for making those changes, and the granularity of the policies, vary widely between firewalls and vendors.

Given this definition, it is easy to see that there is a perimeter (the position between A and B) in which the specific security policy applies. In typical deployed networks, there are usually some easy-to-define perimeters. If two or more networks that are connected by a single device, the perimeter is inside the device. If that device is a firewall, it can impose a security policy at the shared perimeters of those networks.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Reasoning about Firewalls

3.1. The Role of Firewalls in Internet Security

One could compare the role of firewalls in prophylactic perimeter security to that of the human skin: the service that the skin performs for the rest of the body is to keep common crud out, and as a result prevent much damage and infection that could otherwise occur. The body supplies prophylactic perimeter security for itself and then presumes that the security perimeter has been breached; real defenses against attacks on the body include powerful systems that detect changes (anomalies) counterproductive to human health, and recognizable attack syndromes such as common or recently-seen diseases. One might well ask, in view of those superior defenses, whether there is any value in the skin at all; the value is easily stated, however. It is not in preventing the need for the stronger solutions, but in making their expensive invocation less needful and more focused.

3.2. The End-to-End Principle

One common complaint about firewalls in general is that they violate the End-to-End Principle [Saltzer]. The End-to-End Principle is often incorrectly stated as requiring that "application specific functions ought to reside in the end nodes of a network rather than in intermediary nodes, provided they can be implemented 'completely and correctly' in the end nodes" or that "there should be no state in the network." What it actually says is heavily nuanced, and is a line of reasoning applicable when considering any two communication layers.

> [Saltzer] "presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level."

In other words, the End-to-End Argument is not a prohibition against e.g. lower layer retries of transmissions, which can be important in certain LAN technologies, nor of the maintenance of state, nor of consistent policies imposed for security reasons. It is, however, a plea for simplicity. Any behavior of a lower communication layer, whether found in the same system as the higher layer (and especially application) functionality or in a different one, that from the perspective of a higher layer introduces inconsistency, complexity, or coupling extracts a cost. That cost may be in user satisfaction, difficulty of management or fault diagnosis, difficulty of future innovation, reduced performance, or other forms. Such costs need to be clearly and honestly weighed against the benefits expected, and used only if the benefit outweighs the cost.

From that perspective, introduction of a policy that prevents communication under an understood set of circumstances, whether it is to prevent access to pornographic sites or prevents traffic that can be characterized as an attack, does not fail the End-to-End Argument; there are any number of possible sites on the network that are inaccessible at any given time, and the presence of such a policy is easily explained and understood.

What does fail the End-to-End Argument is behavior that is intermittent, difficult to explain, or unpredictable. If I can sometimes reach a site and not at other times, or reach it using this host or application but not another, I wonder why that is true, and may not even know where to look for the issue.

3.3. Building a communication

Any communication requires at least three components:

- a sender, someone or some thing that sends a message,
- a receiver, someone or some thing that receives the message, and
- a channel, which is a medium by which the message is communicated.

In the Internet, the IP network is the channel; it may traverse something as simple as a directly connected cable or as complex as a sequence of ISPs, but it is the means of communication. In normal communications, a sender sends a message via the channel to the receiver, who is willing to receive and operate on it. In contrast, attacks are a form of harassment. A receiver exists, but is unwilling to receive the message, has no application to operate on it, or is by policy unwilling to. Attacks on infrastructure occur when message volume overwhelms infrastructure or uses infrastructure but has no obvious receiver.

By that line of reasoning, a firewall primarily protects infrastructure, by preventing traffic that would attack it from it. The best prophylactic might use a procedure for the dissemination of Flow Specification Rules [RFC5575] to drop traffic sent by an unauthorized or inappropriate sender or which has no host or application willing to receive it as close as possible to the sender.

In other words, as discussed in Section 3.1, a firewall compares to the human skin, and has as its primary purpose the prophylactic defense of a network. By extension, the firewall also protects a set of hosts and applications, and the bandwidth that serves them, as part of a strategy of defense in depth. A firewall is not itself a security strategy; the analogy to the skin would say that a body protected only by the skin has an immune system deficiency and cannot be expected to long survive. That said, every security solution has a set of vulnerabilities; the vulnerabilities of a layered defense is the intersection of the vulnerabilities of the various layers (e.g., a successful attack has to thread each layer of defense).

3.4. The middle way

There is therefore no one way to prevent attacks; there are different kinds of firewalls, and they address different views of the network (please see Section 4 for further discussion). A zone-based firewall (Section 4.1) views the network as containing zones of trust, and deems applications inside its zone of protection to be trustworthy. A role-based firewall (Section 4.2) identifies parties on the basis

of membership in groups, and prevents unauthorized communication between groups. A reputation, anomaly, or signature-based intrusion management system (Section 4.3) depends on active administration, and permits known applications to communicate while excluding unknown or known-evil applications. In each case, the host or application is its own final bastion of defense, but preventing a host from accepting incoming traffic (so-called "host firewalls") does not defend infrastructure. Each type of prophylactic has a purpose, and none of them is a complete prophylactic defense.

Each type of defense, however, can be assisted by enabling an application running in a host to inform the network of what it is willing to receive. As noted in Section 4.1, a zone-based firewall, generally denies all incoming sessions and permits responses to sessions initiated outbound from the zone, but can in some cases be configured to also permit specific classes of incoming session requests, such as WWW or SMTP to an appropriate server. A simple way to enable a zone-based firewall to prevent attacks on infrastructure (traffic to an un-instantiated address or to an application that is off) while not impeding traffic that has a willing host and application would be for the application to inform the firewall of that willingness to receive. The Port Control Protocol [RFC6887], or PCP, is an example of a protocol designed for that purpose.

4. Common kinds of firewalls

There are at least three common kinds of firewalls:

- o Context or Zone-based firewalls, that protect systems within a perimeter from systems outside it,
- o Pervasive routing-based measures, which protect intermingled systems from each other by enforcing role-based policies, and
- o Systems that analyze application behavior and trigger on events that are unusual, match a signature, or involve an untrusted peer.

4.1. Perimeter security: Protection from aliens and intruders

As discussed in [RFC6092], the most common kind of firewall is used at the perimeter of a network. Perimeter security assumes two things: that applications and equipment inside the perimeter are under the control of the local administration and are therefore probably doing reasonable things, and that applications and equipment outside the perimeter are unknown. It may enforce simple permission rules, such as that external web clients are permitted to access a specific web server or that external SMTP MTAs are permitted to access internal SMTP MTAs. Apart from those rules, a session may be

initiated from inside the perimeter, and responses from outside will be allowed through the firewall, but sessions may never be initiated from outside.

In addition, perimeter firewalls often perform some level of testing, either as application proxies or through deep packet inspection, to verify that the protocol claimed to be being passed is in fact the protocol being passed.

In many scenarios the existence and definition of zone-based perimeter defenses is arguably a side-effect of the deployment of Network Address Translation [RFC2993]. Since e.g. a single address is shared among multiple systems, the NAT device needs to translate both the IP addresses and the transport protocol ports in order to multiplex multiple communication instances from different nodes in the same external address. Thus, the NAT device must keep a state table to know how to translate the IP addresses and transport protocol ports of incoming packets. Packets originating from the internal network will either match an existing entry in the state table, or create a new one. On the other hand, packets originating in the external network will either match an existing entry in the state table, or be dropped. Thus, as a side effect, NATs implicitly require that communication be initiated from the internal network, and only allow return traffic from the external network.

Some applications make the mistake of coupling application identities to network layer addresses, and hence employ such addresses in the application protocol. Thus, Network Address Translation forces the translator to interpret packet payloads and change addresses where used by applications.

As a result, if the transport or application headers are not understood by the translator, this has the effect of damaging or preventing communication. Detection of such issues can be sold as a security feature, although it is really a side-effect of a failure. While this can have useful side-effects, such as preventing the passage of attack traffic that masquerades as some well-known protocol, it also has the nasty side-effect of making innovation difficult. This has slowed the deployment of SCTP [RFC4960], since a firewall will often not permit a protocol it doesn't know even if a user behind it opens the session. When a new protocol or feature is defined, the firewall needs to stop applying that rule, and that can be difficult to make happen.

4.2. Pervasive access control

Another access control model, often called "Role-based", tries to control traffic in flight regardless of the perimeter. Given a rule that equipment located in a given routing domain or with a specific characteristic (such as "student dorms") should not be able to access equipment in another domain or with a specific characteristic (such as "academic records"), it might prevent routing from announcing the second route in the domain of the first, or it might tag individual packets ("I'm from the student dorm") and filter on those tags at enforcement points throughout network. Such rules can be applied to individuals as well as equipment; in that case, the host needs to tag the traffic, or there must be a reliable correlation between equipment and its user.

One common use of this model is in data centers, in which physical or virtual machines from one tenant (which is not necessarily an "owner" as much as it is a context in which the system is used) might be co-resident with physical or virtual machines from another. Inter-tenant attacks, espionage, and fraud are prevented by enforcing a rule that traffic from systems used by any given tenant is only delivered to other systems used by the same tenant. This might, of course have nuances; under stated circumstances, identified systems or identified users might be able to cross such a boundary.

The major impediment in deployment is complexity. The administration has the option to assign policies for individuals on the basis of their current location (e.g. as the cross-product of people, equipment, and topology), meaning that policies can multiply wildly. The administrator that applies a complex role-based access policy is probably most justly condemned to live in the world he or she has created.

4.3. Intrusion Management: Contract and Reputation filters

The model proposed in Advanced Security for IPv6 CPE [I-D.vyncke-advanced-ipv6-security] could be compared to purchasing an anti-virus software package for one's computer. The proposal is to install a set of filters, perhaps automatically updated, that identify "bad stuff" and make it inaccessible, while not impeding anything else.

It depends on four basic features:

o A frequently-updated signature-based Intrusion Prevention System which inspects a pre-defined set of protocols at all layers (from layer-3 to layer-7) and uses a vast set of heuristics to detect

attacks within one or several flows. Upon detection, the flow is terminated and an event is logged for further optional auditing.

- A centralized reputation database that scores prefixes for degree of trust. This is unlikely to be on addresses per se, since e.g. temporary addresses [RFC4941] change regularly and frequently.

- Local correlation of attack-related information, and

- Global correlation of attacks seen, in a reputation database

The proposal doesn't mention anomaly-based intrusion detection, which could be used to detect zero-day attacks and new applications or attacks. This would be an obvious extension.

The comparison to anti-virus software is real; anti-virus software uses similar algorithms, but on API calls or on data exchanged rather than on network traffic, and for identified threats is often effective.

The proposal also has weaknesses:

- People don't generally maintain anti-virus packages very well, letting contracts expire,

- Reputation databases have a bad reputation for distributing information which is incorrect or out of date,

- Anomaly-based analysis identifies changes but is often ineffective in determining whether new application or application behaviors are pernicious (false positives). Someone therefore has to actively decide - a workload the average homeowner might have little patience for, and

- Signature-based analysis applies to attacks that have been previously identified, and must be updated as new attacks develop. As a result, in a world in which new attacks literally arise daily, the administrative workload can be intense, and reflexive responses like accepting https certificates that are out of date or the download and installation of unsigned software on the assumption that the site admin is behind are themselves vectors for attack.

Security has to be maintained to be useful, because attacks are maintained.

5. Firewalling Strategies

There is a great deal of tension in firewall policies between two primary goals of networking: the security goal of "block traffic unless it is explicitly allowed" and the networking goal of "trust hosts with new protocols". The two inherently cannot coexist easily in a set of policies for a firewall.

5.1. Blocking Traffic Unless It Is Explicitly Allowed (default deny)

Many networks enforce the so-called "default deny" policy, in which traffic is blocked unless it is explicitly allowed. The rationale for such policy is that it is easier to open "holes" in a firewall for allowing specific protocols, than trying to block all protocols that might be employed as an attack vectors; and that a network should only support the protocols it has been explicitly designed to support.

The drawback of this approach is that the security goal of "block traffic unless it is explicitly allowed" prevents useful new applications. This problem has been seen repeatedly over the past decade: a new and useful application protocol is specified, but it cannot get wide adoption because it is blocked by firewalls. The result has been a tendency to try to run new protocols over established applications, particularly over HTTP [RFC3205]. The result is protocols that do not work as well they might if they were designed from scratch.

Worse, the same goal prevents the deployment of useful transports other than TCP, UDP, and ICMP. A conservative firewall that only knows those three transports will block new transports such as SCTP [RFC4960]; this in turn causes the Internet to not be able to grow in a healthy fashion. Many firewalls will also block TCP and UDP options they don't understand, and this has the same unfortunate result.

6. Assumptions on IP addresses and Transport Protocol Port Numbers

In a number of scenarios, firewalls rules have traditionally been specified in terms of the associated IP addresses and transport protocol port numbers. In general, this assumes that the associated IP addresses are stable, and that there is a "well known" transport protocol port number associated with each application.

In the IPv4 world, IP addresses may be considered rather stable. However, IPv6 introduces the concept of "temporary addresses" [RFC4941] which, by definition, change over time. This may prevent the enforcement of filtering policies based on specific IPv6

addresses, or may lead to filtering based on a specific address prefixes (as opposed to specific IPv6 addresses). In some scenarios, from the point of view of enforcing filtering policies, it might be desirable to disable temporary addresses altogether.

For example, an administrator might prefer that the caching DNS server or SMTP MTA always employ the same source IPv6 address, as opposed to the temporary addresses that change over time.

The server-side transport protocol port is generally the so-called "well-known port" corresponding to the associated application. While widespread, this practice should probably be considered a kludge/short-cut rather than a "design principle" that can be relied upon for the general case. For example, use of DNS SRV records [RFC2782], or applications such as "portmapper" [Portmap] [RFC1833] might mean that the associated transport protocol port number cannot be assumed to be well-known, but rather needs to be dynamically learned.

7. State Associated with Packet Filtering

There are two main paradigms for packet filtering:

- o Stateless filtering

- o Stateful filtering

Stateless filtering implies that the decision on whether to allow or block a specific packet is solely based on the contents of such packet. One common example of such paradigm is the enforcement of network ingress filtering [RFC2827], in which packets may be blocked based on their IP addresses. Stateless filtering scales well, since there are no state requirements on the filtering device other than that associated with maintaining the filtering rules to be applied to incoming packets.

On the other hand, stateful filtering implies that the decision on whether to allow or block a packet is not only based on the contents of the packet, but also on the existence (or lack of) previous traffic/state associated with such packet. A common example of such paradigm is a firewall that "allows outbound connection requests and only allows return traffic from the external network" (such as the policy implicitly enforced my most NAT devices). For obvious reasons, the firewall needs to maintain state in order to be able to enforce such policies. For example, a firewall may need to keep track of all on-going communication instances, possibly applying timeouts and garbage collection on the associated state table.

Stateful filtering tends to allow more powerful packet filtering, at the expense of increased state. Thus, stateful filtering may be desirable when trying to perform deep packet inspection, but may be undesirable when the firewall is meant to block some Denial of Service attacks, since the firewall itself may become "the weakest link in the chain".

8. Enforcing Protocol Syntax at the Firewall

Some firewalls try to enforce the protocol syntax by checking that only packets complying with existing protocol definitions are allowed. While this can have useful side-effects, such as preventing the passage of attack traffic that masquerades as some well-known protocol, it also has the nasty side-effect of making innovation difficult. For example, one of the issues in the deployment of Explicit Congestion Notification [RFC3168] has been that common firewalls often test reserved/unused bits and require them to be set to zero to close covert channels. When a new protocol or feature is defined, the firewall needs to stop applying that rule, and that can be difficult to make happen.

A somewhat related concept is that of traffic normalization (or "scrubbing"), in which the filtering device can "normalize" traffic by e.g. clearing bits that are expected to be cleared, changing some protocol fields such that they are within "normal" ranges, etc. (see e.g. the discussion of "traffic normalization" in [OpenBSD-PF]). While this can have the useful effect of blocking DoS attacks to sloppy implementations that do not enforce sanity checks on the received packets, it also has the nasty side-effect of making innovation difficult, or even breaking deployed protocols.

> For example, some firewalls are known enforce a default packet normalization policy that clears the TCP URG bit, as a result of the TCP urgent mechanism being associated with some popular DoS attacks. Widespread deployment of such firewalls has essentially rendered the TCP urgent mechanism unusable, leading to its eventual formal deprecation in [RFC6093].

9. Performing Deep Packet Inspection

While filtering packets based on the layer-3 protocol header fields is rather simple and straight-forward, performing packet filtering based on the upper layer protocols can be a challenging task.

For example, IP fragmentation may make this task quite challenging, since even the very layer-4 protocol header could be present in a non-first fragment. In a similar vein, IPv6 extension headers may represent a challenge for a filtering device, since they can result

in long IPv6 extension header chains [RFC7112] [I-D.gont-v6ops-ipv6-ehs-packet-drops].

This problem is exacerbated as one tries to filter packets based on upper layer protocol contents, since many of such protocols implement some form of fragmentation/segmentation and reassembly. In many cases, the reassembly process could possibly lead to different results, and this may be exploited by attackers for circumventing security controls [Ptacek1998] [RFC6274].

In general, the upper in the protocol stack that packet filtering is performed, the more state that is required at the filtering device. And when stateful packet filtering is warranted, its associated security implications should be considered.

10. Recommendations

Zone-based firewalls, when used, SHOULD exclude all session initiation from outside the zone regardless of attributes such as the use of IPsec. They SHOULD also facilitate the use of a protocol such as PCP by hosts to identify traffic (IPsec AH, IPsec ESP, transports in general, or transports using specified destination port ranges) that they are willing to receive, and interpret that into rules permitting specified traffic to those specific systems. Being fully automated and easily understood, such firewalls are appropriate for networks with passive administration.

Role-based firewalls can be implemented using routing technology. For example, if Alice should not be able to send a message to Bob, Alice's routing system might not have a route to Bob, or Bob's routing system might not have a route to Alice. Role-based firewalls can also be implemented using filtering technology; Alice, Alice's router, Bob's router, or Bob may have a filter that prevents communication between them. While there can be issues in specific cases, a routing implementation is generally more scalable and more easily managed.

Reputation, anomaly, or signature-based intrusion management is generally proprietary; a service maintains the list of exclusions, which must be updated as new kinds of attacks are developed. Implementations SHOULD be designed for frequent and scalable updating.

As further discussed in Section 4.1, firewalls of any type SHOULD NOT attempt to perform the kind of deep packet inspection and surgery that is common with Network Address Translators [RFC2993]. There is marginal value in detecting the spoofing of applications by attack

traffic, but the side-effects of preventing protocol improvement and application innovation are destructive and unnecessary.

11. IANA Considerations

This memo asks the IANA for no new parameters. It can before publication as an RFC by the RFC Editor.

12. Security Considerations

This note reasons about security considerations. It introduces no new ones.

13. Acknowledgements

This document is based on [I-D.ietf-opsawg-firewalls-00] authored by Fred Baker, and [I-D.ietf-opsawg-firewalls-01] authored by Paul Hoffman.

14. References

14.1. Normative References

[RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, DOI 10.17487/RFC1833, August 1995, <http://www.rfc-editor.org/info/rfc1833>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3205] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, DOI 10.17487/RFC3205, February 2002, <http://www.rfc-editor.org/info/rfc3205>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <http://www.rfc-editor.org/info/rfc4941>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <http://www.rfc-editor.org/info/rfc6887>.

[RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <http://www.rfc-editor.org/info/rfc7112>.

14.2. Informative References

[I-D.gont-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., LIU, S., and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-00 (work in progress), July 2015.

[I-D.ietf-opsawg-firewalls-00]
Baker, F., "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-00 (work in progress), June 2012.

[I-D.ietf-opsawg-firewalls-01]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.

[I-D.vyncke-advanced-ipv6-security]
Vyncke, E., Yourtchenko, A., and M. Townsley, "Advanced Security for IPv6 CPE", draft-vyncke-advanced-ipv6-security-03 (work in progress), October 2011.

[OpenBSD-PF]
OpenBSD, , "pf(4) manual page: pf -- packet filter", 2015, <http://www.openbsd.org/cgi-bin/man.cgi/OpenBSD-current/man4/pf.4&query=pf>.

[Portmap] Wikipedia, , "Portmap", 2014, <https://en.wikipedia.org/wiki/Portmap>.

[Ptacek1998]
Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", 1998, <http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <http://www.rfc-editor.org/info/rfc2782>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <http://www.rfc-editor.org/info/rfc2827>.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <http://www.rfc-editor.org/info/rfc2993>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <http://www.rfc-editor.org/info/rfc3168>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://www.rfc-editor.org/info/rfc4960>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <http://www.rfc-editor.org/info/rfc5575>.

[RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <http://www.rfc-editor.org/info/rfc6092>.

[RFC6093] Gont, F. and A. Yourtchenko, "On the Implementation of the TCP Urgent Mechanism", RFC 6093, DOI 10.17487/RFC6093, January 2011, <http://www.rfc-editor.org/info/rfc6093>.

[RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011, <http://www.rfc-editor.org/info/rfc6274>.

[Saltzer] Saltzer, J., Reed, D., and D. Clark, "End-to-end arguments in system design", ACM Transactions on Computer Systems (TOCS) v.2 n.4, p277-288, Nov 1984.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org