Barbarians in the Gate: An Experimental Validation of NIC-based Distributed Firewall Performance and Flood Tolerance

Michael Ihde and William H. Sanders
Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign
Urbana, IL 61801, U.S.A.
Telephone: (217)333-0345 / Fax: (217)244-3359
Email: {ihde,whs}@crhc.uiuc.edu

Abstract

This paper presents our experience validating the flood tolerance of two network interface card (NIC)-based embedded firewall solutions, the Embedded Firewall (EFW) and the Autonomic Distributed Firewall (ADF). Experiments were performed for both embedded firewall devices to determine their flood tolerance and performance characteristics. The results show that both are vulnerable to packet flood attacks on a 100 Mbps network. In certain configurations, we found that both embedded firewall devices can have a significant, negative impact on bandwidth and application performance. These results imply first that, firewall rule-sets should be optimized for performance-sensitive applications, and second, that proper consideration must be given to attack risks and mitigations before either the EFW or ADF is deployed. Finally, we believe that future embedded firewall implementations should be vetted in a manner similar to that presented in this paper. Our experience shows that when their limitations are properly considered, both the EFW and ADF can be safely deployed to enhance network security without undue risk.

1. Introduction

With the increasing popularity of the Internet, the threat of cyber-attacks has become a significant problem. Many of the old maxims of network security are no longer effective against modern threats. Recent experience with worms, such as MyDoom and Sobig, have shown that standard firewalls provide inadequate protection for threats that can bypass the perimeter protection either through allowed communications, like e-mail, or through mobile hosts that temporarily leave the safety of a firewall only to bring a worm into the network behind the firewall.

The effects of these worms, which bypass the external firewall and then spread unchecked behind the firewall, have made it clear that traditional firewalls at the network perimeter no longer provide sufficient protection. Effective network security requires defense-in-depth, with security mechanisms at both the network perimeter and the network end-points.

Distributed firewalls are one mechanism that can be used to enhance the depth of the defense. With distributed firewalls, each host is protected by its own firewall, preventing any single vulnerability from affecting all hosts.

Although distributed firewalls provide enhanced network protection, we believe that it is dangerous to simply trust the implementation of any security mechanism. Security devices (software or hardware), especially those that have been recently developed, can harbor hidden vulnerabilities that an attacker may exploit, thus negating the usefulness of any additional security gained and providing a false sense of security. The prudent decision is to ensure that all security devices undergo a suitable level of validation to ensure that the device is free of vulnerabilities.

This paper presents our experience validating the flood tolerance of two NIC-based embedded firewall solutions, the EFW and the ADF. The EFW is a commercially available device from 3COM, while the ADF is a derivative of the EFW being developed by Adventium Labs. This validation was carried out during the validation of a survivable publish, subscribe, and query (PSQ) system developed in response to a DARPA challenge. In the candidate system, called *Designing Protection and Adaptation into a Survivability Architecture* (DPASA), the ADF was used as part of the defense-in-depth strategy. The ADF was employed to protect the contents of all host-to-host communication via encrypted data channels (called virtual private groups (VPGs)) while also preventing unauthorized communications. Guided by the following warning found in RFC2647, our validation effort focused on testing the firewalls' flood tolerance.

> Further, certain forms of attack may degrade performance. One common form of denial-of-service (DoS) attack bombards a firewall with so much rejected traffic that it cannot forward allowed traffic. DoS attacks do not always involve heavy loads; by

> definition, DoS describes any state in which a firewall is offered rejected traffic that prohibits it from forwarding some or all allowed traffic. Even a small amount of traffic may significantly degrade firewall performance, or stop the firewall altogether. Further, the safeguards in firewalls to guard against such attacks may have a significant negative impact on performance [17].

Surprisingly, we found that a successful denial-of-service attack could be launched against either the EFW or ADF with the smallest "default allow all" rule-set using only 30% of the maximum frame rate on a 100 Mbps Ethernet network. Although we were specifically looking for denial-of-service vulnerabilities, this paper also presents the performance characteristics of the EFW and ADF. Our results show that, unlike modern software-based firewalls or standard NICs, both the EFW and ADF had a significant impact on network performance, even when enforcing small rule-sets.

As was shown with DPASA, the EFW/ADF can be deployed with low risk when the proper safeguards are implemented. The continual gain in support for embedded firewall solutions [2] indicates that the results in this paper are applicable outside of the DPASA project. In addition, the methodology presented in this paper is flexible enough to be applied to other host-based, distributed firewalls as they become available.

2. Related Work

The distributed firewall concept, first introduced by Bellovin in 1999 [4], provides firewall protection at the network end-points via a centrally defined policy. Unlike traditional firewalls, which only provide protection at the network perimeter, distributed firewalls can provide host protection for internal threats. Distributed firewalls are topology-independent, provide fine-grained access control, and reduce global performance bottlenecks.

Distributed firewalls are available as either software or hardware solutions. Early software-based distributed firewall implementations existed as research projects; a preliminary OpenBSD implementation [12] based on Bellovin's original concept, and later the StrongMan [13] framework, are two such examples. There also exist a few commercial software implementations by Green Bow and FSecure. We are only aware of two hardware solutions, the 3Com EFW and the ADF.

The EFW and ADF are hardware-based distributed firewalls that enforce the rule-set on the NIC [16, 18]. Both implementations share a common ancestral code-base and similar underlying hardware. The EFW was developed first, providing stateless packet filtering and a central policy server. The ADF later added the ability to create encrypted communication channels, called VPGs [6,15,16], which provide confidentiality, integrity, and sender authentication.

One of the primary goals of the EFW project was to remain cost-effective for large networks. To achieve this the device must be "fast, simple, and cheap" [18]. By implementing the EFW functionality on top of an inexpensive existing network card (3CR990) the hardware costs were kept low enough for normal deployment. Although more expensive, hardware designed especially for packet filtering may have provided higher performance and possibly would have been able to withstand a packet flood attack.

Performance data, similar to that found in this paper, has been presented for the two most common open-source firewall software packages in [10] and [8]. The results in both papers can be used to compare the performance of NIC-based firewalls to software firewalls. In [14] the Linux TIS firewall toolkit performance is assessed using HTTP and FTP scenarios. Unlike NIC-based firewalls, the performance of software-based firewalls is directly related to the computational performance of the underlying host.

For the EFW (but not the ADF), basic performance data has been presented in [20]. Unlike [20], our methodology directly measures flood tolerance by initiating a packet flood, much like an attacker would. This difference in perspective allowed us to identify the denial-of-service vulnerability, that was not identified in [20], during the DPASA validation effort.

Two request for comments (RFC) papers [5, 9] provide recommendations for analyzing network interconnect devices and firewalls. Whenever possible we attempted to follow the guidelines in each RFC paper, deviating only when the particular nature of the EFW and ADF firewalls demanded such modifications.

As previously mentioned, the experiments in this paper were carried out in the context of a larger validation effort that aimed to validate a survivable PSQ system in response to a 2002 DARPA challenge. The candidate solution, called DPASA, was designed and developed by a team led by BBN Technologies. The DPASA system is documented in [3, 7, 19, 21].

3. Experimental Methodology

All experiments were performed on an isolated network, eliminating extraneous packets and thus preventing the results from being skewed. Our configuration required four hosts connected via a standard 100 Mbps switch (3COM 3C16734A): the EFW policy server, flood generator (i.e., attacker), client, and target. We assumed that the Ethernet switch itself would not affect the results in any significant manner, and verified the assumption by performing identical tests against a standard non-filtering NIC (Intel EEPro 100). The performance loss, if any, for the standard non-filtering NIC would be attributed to the network switch and infrastructure. In all of our tests the EFW and ADF experienced much greater losses than those found with the standard NIC. Thus we were confident that the switch was not causing the performance loss.

The hosts in our experiment used 1 GHz Pentium III processors with 256MB of RAM. Except for the policy server, which required Microsoft Windows 2000, the hosts ran Redhat Linux. For the EFW host, a 2.4 Linux kernel was used instead of the more recent 2.6 kernel because the EFW lacked official driver support for the 2.6 Linux kernel. It was as-

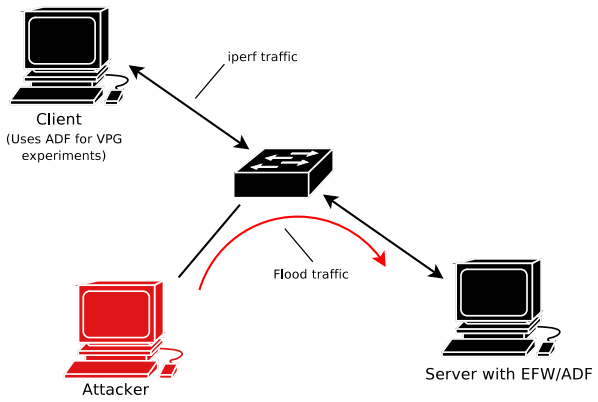


Figure 1. Experimental Network Setup

sumed that no major performance differences existed between the 2.4 and 2.6 kernels.

The rule-sets used in the experiments were configured to act on the packets at a particular rule in the rule-set, which we call the "action rule." We found that any rules following the "action rule" did not affect the flood tolerance or performance. This was expected, because as soon as a matching rule is found for an incoming packet, no further processing is required. Therefore, when we refer to rule-set length (or depth) we are technically referring to the number of rules up to and including the "action rule."

For VPGs, the "action rule" is the pair of rules that fully define one VPG. The depth of the rule-set is increased by adding additional non-matching VPGs above the "action rule"; thus, a rule-set with four VPGs has three VPGs that do not match the desired incoming traffic and one VPG that does match the incoming traffic.

We measured bandwidth between two hosts using `iperf` ,a cross-platform client-server software tool capable of measuring both transmission control protocol (TCP) and user datagram protocol (UDP) bandwidth. In order to measure available bandwidth, it was necessary for the firewall policy to allow communication between the `iperf` client and server, as seen in Figure 1.

Hyper-Text Transfer Protocol (HTTP) load tests were performed using `http_load` to repeatedly request a web page from an `apache2` web server. The web server was configured with the default Gentoo configuration. To achieve the goal of measuring performance loss, `http_load` was configured to use at most one connection at a time with an unlimited rate for 30 s. Alternatively, `http_load` could have been configured to measure the number of parallel connections supported by the server at a given connection rate.

The flood tolerance of the EFW/ADF was tested using an additional machine as the hypothetical attacker, as seen in Figure 1. Tests were carried out by sending a packet flood at the target and then taking bandwidth measurements between the client and the target. If the flood was able to prevent the measurement from succeeding (i.e., 0 Mbps), then the denial-of-service attempt was deemed successful. The implementation of our flood generator is documented in [11].

4. Results

4.1. Available bandwidth

Our first experiment measured the maximum bandwidth supported by the embedded firewalls. In DPASA, it was assumed that the embedded firewalls would support the full network bandwidth because, in general, NICs do not significantly impact network performance. If it were shown that embedded firewalls did not support full network bandwidth, then it would be possible to packet-flood an embedded firewall to initiate a denial-of-service attack. In other words, if the maximum throughput of the embedded firewall was less than the theoretical maximum packet rate of the network, then the host would be vulnerable to a denial-of-service attack.

Ideally, we would have measured maximum throughput directly via the methods detailed in RFC2544 [5]. However, the methods in RFC2544 are better-suited to traditional firewalls, which have separate incoming and outgoing interfaces. Attempting to use the same measurement techniques for distributed firewalls would have required that the EFW/ADF host forward packets out of a second interface, adding additional overhead and potential complications in the experiment. As an alternative, available bandwidth can be measured using only a single network interface with no packet forwarding. If there is bandwidth loss the maximum throughput can be calculated with the simple relation $Max.Throughput = BW/FrameSize$. If no bandwidth loss is measured, it does not imply that the maximum throughput of the firewall is greater than the maximum frame rate of the network. It only means that the firewall can support the maximum packet rate for large packets. Smaller packets, which can be transmitted at a higher rate, may still cause a denial-of-service.

During the available bandwidth tests the frames were the maximum size supported by Ethernet (1518 bytes per frame); thus, the EFW/ADF was only able to process approximately 4100 packets/s when the policy contained 64 rules. For smaller policies, it was impossible to determine whether a smaller, higher-rate packet stream would overload the firewall card. With one rule the EFW/ADF was able to support the full network bandwidth. However, the frame rate was much lower than the maximum frame rate achievable with smaller frames (but lower bandwidth). Thus, the maximum throughput of the firewall could not be determined from the bandwidth experiments for all rule-set configurations.

The results are presented in Figure 2. We anticipated that as the rule-set size was increased the EFW/ADF would suffer some performance loss. The amount of performance loss, however, was surprising. When configured with the largest possible rule-set the EFW and ADF respectively lost 45% and 65% of nominal bandwidth capacity. However, when configured with smaller rule-sets (those with less than 20 rules) there was no significant performance loss. Based on this experiment alone, it seems wise to limit rule-set depth or place bandwidth-sensitive traffic early in the rule-set.

For comparison purposes, we performed identical tests against `iptables`. Our results were identical to Hoffman et al.'s [10] results. We found `iptables` had no bandwidth loss

for rule-sets containing less than 64 rules on a 100 Mbps network.

When the ADF was configured to use VPGs, the performance drop was more significant than that seen with a non-VPG rule-set. We believe this is due to the additional encryption/decryption overhead for all VPG packets processed by the ADF. When additional non-matching VPGs (those that did not match the packets of the VPG under test) were inserted into the rule-set, the performance did not decrease by any appreciable amount. That implies that the ADF is able to avoid decrypting incoming packets until they reach the matching VPG rule.

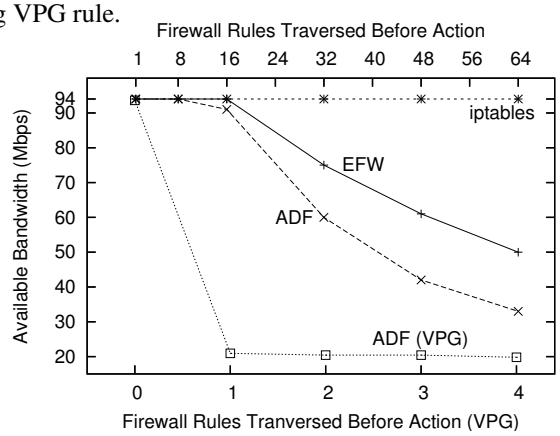


Figure 2. Available Bandwidth as Rules Are Added to the Rule-Set

4.2. Available bandwidth during floods

The poor performance of the EFW/ADF indicated that a packet flood may easily overload the EFW/ADF card. To measure flood tolerance, another experiment was used. First, a packet flood was directed at the firewall, and then the available bandwidth was measured. If the flood consumed all of the firewall resources, then there would be no bandwidth available. As with the previous bandwidth measurements, this measurement indirectly measures maximum throughput. In this experiment we measured the maximum throughput of the minimal, one rule, rule-set.

At each of nine flood rates, three bandwidth measurements were taken and averaged. The results are shown in Figure 3(a). For both the EFW and ADF, a major portion of bandwidth was lost with a flood of 16 000 packets per second. A flood with 20 000 packets per second caused the available bandwidth to drop to almost zero, thus creating a successful denial-of-service attack. The drastic bandwidth loss seen in the EFW/ADF did not occur for either the standard NIC or `iptables`, which both supported 77 Mbps when flooded with 20 000 packets per second. That leads to the conclusion that the EFW and ADF are alone responsible for the loss. In addition, the flood tolerance of a single VPG was interesting due to the near-linear relation between bandwidth and flood rate.

4.3. Minimum flood rate

From the previous experiment, it was clear that even the simple one-rule rule-set is vulnerable to denial-of-service attacks. However, it would be rare to find an embedded firewall that was deployed with such a simple rule-set. Therefore, it is important to determine if the addition of rules to the rule-set decreases the minimum required flood rate.

We define the minimum flood rate to be the minimum packet rate with which an attacker must flood the firewall in order to successfully cause a denial of service. The minimum packet rate was determined by incrementally increasing the flood rate until the measured bandwidth fell to approximately 0 Mbps. We tested two different rule-set classes: one with the flood packets being allowed by the rule-set and another with the flood packets denied. In each case, the action (allow or deny) was taken on at rules 1, 8, 16, 32, and 64. The results are presented in Figure 3(b).

With only eight rules, the performance was low enough that an attacker on a 10 Mbps network could easily cause a denial-of-service attack if the flood packets were being allowed by the rule-set. When the largest rule-set was enforced, the attacker host only needed to generate 4500 packets/s to create a denial-of-service.

We found that some flood tolerance could be gained by denying the flood packets. This effect, though, was actually due to the lack of any outgoing TCP responses that were being generated when the flood packets reached the host. When attack packets are dropped, the host will not receive the packet; thus, no outgoing response packets are sent. As a result, in the experiment, total traffic through the firewall was halved, doubling the required flood rate.

In conflict with the earlier recommendation to place bandwidth-sensitive services early in the rule-set, it is also important for the policy to deny any potential sources of attack early in the rule-set. However, early denial is only partially effective in preventing flood attacks, given the attacker's ability to spoof packets that will traverse deeper into the rule-set.

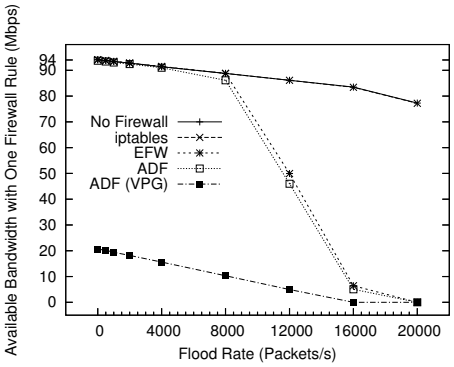
During the experiments it was not possible to capture any data for the EFW Deny-All case, because the card would stop processing packets when it was flooded with over 1000 packets/s. Restarting the firewall agent software restored functionality to the NIC until the next flood test. No solution was found.

As expected, `iptables` was able to withstand any packet flood attack directed at it. Hoffman found that `iptable`'s performance [10] had 22% network utilization with a 100 rule rule-set on a 100 Mbps network (with 64-byte frames). This utilization translates to approximately 33 000 packets/s; thus, with only 64 rules, it is unlikely that our flood generator was able to achieve a rate sufficient to flood the firewall.

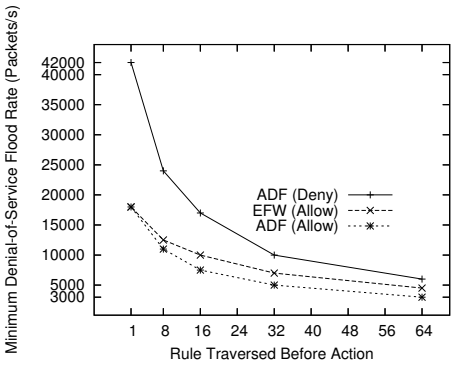
4.4. HTTP performance

The denial-of-service experiments indicate that the EFW/ADF may have a significant effect on application performance. Because there is no easy way to convert raw packet performance to application-level performance, we performed an additional experiment measuring web server performance.

HTTP performance tests were run against an `apache2` web server. The measurements provided direct insight into the performance decrease associated with the firewall filtering. As



(a) Available Bandwidth During Packet Flood With a Single-Rule Rule-Set



(b) Minimum Flood Rate Required to Cause Denial-of-Service as Rule-set Depth Increases

Figure 3. Testing for Denial-of-Service Tolerance

anticipated, if the rule allowing HTTP traffic was placed deep in the rule-set, performance decreased.

Three measurements are provided by `http_load`: throughput, connection latency, and response latency. The throughput of the server, measured in page fetches per second, provides a rough estimate of how many users the server can support simultaneously. Connection latency is the time required to complete the 3-way TCP handshake. Response latency is the time required to complete the entire transfer of the requested web page.

Figure 1 shows that the ADF offered lower performance than a standard NIC in all configurations. As the action rule was placed deeper in the rule-set, web-server throughput was reduced. At its worst, the ADF was responsible for a 41% performance decrease compared to a standard NIC.

The connection time and response time are latency metrics that are important for interactive applications. Figure 1 shows that both latency measures increased as the rule-set size increased, but the additional delay was not excessive. Any additional latency would hardly be noticeable for Internet service, which typically has a latency greater than 50 ms. The additional latency might be noticeable for local area networks, but would only be problematic for the most demanding real-time applications.

Use of VPGs also significantly affected HTTP performance. Figure 1 shows that the addition of a VPG dropped performance significantly, but that the insertion of other non-matching VPG rules did not alter the performance. This is similar to the effect seen for the available bandwidth experiments.

4.5. Analysis of results

The experimental results show that neither the EFW nor the ADF performs well enough to be used safely on a 100 Mbps network. In our opinion this is problematic for a couple of reasons. First, given the proliferation of 100 Mbps networks, it is unlikely that the devices will be used only on 10 Mbps networks. Second, we have found no publicly available EFW documentation that provides performance data or vulnerability warnings for packet flood attacks. Finally, due to the cost of the devices and the nature of their implementation, we believe that they will most likely be used in environments where security is of the utmost importance. In those environments, it is important to be aware of all vulnerabilities so that proper mitigations can be implemented.

Even on a 10 Mbps network, the EFW/ADF can be safely used *only if* the rule-set is kept to under eight rules. In general it would be very difficult to provide a useful rule-set in under eight rules. For example, to protect an Oracle database server, 3COM recommends a rule-set that requires at least 31 rules to protect the appropriate ports [1].

If we had access to the source code and hardware schematics, it would have been possible to determine the exact reason for the performance bottleneck. Unfortunately, without such access, it is only possible to make conjectures about the exact implementation used on either the EFW or the ADF. Further discussion of our results is available in [11].

5. Conclusion

In this paper, we have presented the validation of two NIC-based distributed firewalls, the Embedded Firewall and the Autonomic Distributed Firewall. Specifically, we were interested in determining if either of the devices are susceptible to packet flood denial-of-service attacks. Our results indicated that both devices can easily be flooded, preventing valid traffic from being processed.

Our experiments indicate that an attacker can easily overload the NIC with packets, even if the device is enforcing the simplest "default allow" rule. An attacker only needs to generate a flood of 42 000 packets per second, a rate easily achievable on a 100 Mbps network. As rules were added to the rule-set we found the minimum flood rate declined; for a full rule-set (64 rules) with the attack packets being allowed by the firewall, the minimum required flood rate was only 4 500 packets per second. Because the flood packets in the rule-set were blocked, some denial-of-service tolerance was gained. However, the attacker only needs to spoof packets with the right IP address and ports to have the packets pass through the firewall.

In the process of determining the flood tolerance, we have also measured the performance characteristics of the firewall devices. Our performance tests measured the available

Table 1. HTTP Performance of Apache Webserver Protected by an ADF

		ADF with Standard Rules						ADF with VPG Rules				
Experiment	Standard NIC	1	8	16	32	48	64	0	1	2	3	4
HTTP Fetches/s	380	362	349	330	285	248	222	341	205	202	200	199
ms/Connect	0.215	0.269	0.342	0.440	0.813	1.140	1.340	0.326	0.994	1.03	1.06	1.09
ms/First-Response	2.167	2.242	2.276	2.325	2.402	2.499	2.615	2.316	2.825	2.852	2.861	2.869

bandwidth of the EFW/ADF as a function of rule-set size. We found that for rule-sets containing fewer than 20 rules there was no significant performance loss. However, with the largest rule-set of 64 rules, the EFW was only able to operate at 50 Mbps, half of full network speed. The ADF was only able to provide around 33 Mbps throughput, likely the result of a less efficient packet filtering algorithm, as both devices are built on the same hardware platform.

Despite the EFW/ADF's vulnerability to denial-of-service attacks, we believe the benefits provided by a NIC-based distributed firewall outweigh the drawbacks. As part of a strong defense-in-depth strategy, distributed firewall NICs provide defense at the network edge. If they are deployed with the above limitations in mind, the network administrator can safely utilize the EFW and ADF. It is our hope that this research encourages the development of new embedded firewall devices that have sufficient tolerance to simple packet flood attacks.

Acknowledgments

This paper was supported, in part, by DARPA contract number F30602-02-C-0134. We would like to thank Charlie Payne and Dick O'Brien, of Adventium Labs, for their invaluable help verifying our results. We would also like to thank all members of the DPASA team.

References

[1] CERT coordination center threats and 3Com embedded firewall protection, May 2003.

[2] 3COM. Department of homeland security selects 3com and adventium labs to secure critical infrastructure, September 2005.

[3] M. Atighetchi, P. Rubel, P. Pal, J. Chong, and L. Sudin. Networking aspects in the DPASA survivability architecture: An experience report. In *The 4th IEEE International Symposium on Network Computing and Applications*, 2005.

[4] S. M. Bellovin. Distributed firewalls. *;login:*, pages 39–47, Nov 1999.

[5] S. Bradner and J. McQuaid. Benchmarking methodology for network interconnect devices. RFC 2544, Internet Engineering Task Force, March 1999.

[6] M. Carney, R. O. Hanzlik, and T. R. Markham. Virtual private groups. presented at Third Annual IEEE Information Assurance Workshop, June 2002.

[7] J. Chong, P. Pal, M. Atigetchi, P. Rubel, and F. Webber. Survivability architecture of a mission critical system: The DPASA example. In *Proceedings of the 21st Annual Computer Security Applications Conference*, December 2005.

[8] D. Hartmeier. Design and performance of the OpenBSD stateful packet filter (pf). In *Proceedings of the USENIX Annual Technical Conference, Freenix Track*, pages 171–180, June 2002.

[9] B. Hickman, D. Newman, S. Tadjudin, and T. Martin. Benchmarking methodology for firewall performance. RFC 3511, Internet Engineering Task Force, April 2003.

[10] D. Hoffman, D. Prabhakar, and P. Strooper. Testing iptables. In *Proceedings of the 2003 Conference of the Centre for Advanced Studies on Collaborative Research*, pages 80–91, October 2003.

[11] M. Ihde. Experimental evaluations of embedded distributed firewalls: Performance and policy. Master's thesis, University of Illinois at Urbana-Champagin, 2005.

[12] S. Ioannidis, A. D. Keromytis, S. M. Bellovin, and J. M. Smith. Implementing a distributed firewall. In *Proceedings of the Seventh ACM Conference on Computer and Communications Security*, pages 190–199, November 2000.

[13] A. D. Keromytis, S. Ioannidis, M. B. Greenwald, and J. Smith. The strongman architecture. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition*, volume 1, pages 178–188, April 2003.

[14] M. R. Lyu and L. K. Y. Lau. Firewall security: Policies, testing and performance evaluation. In *Proceedings of the 24th International Computer Software and Applications Conference*, pages 116–121, October 2000.

[15] T. Markham, L. Meredith, and C. Payne. Distributed embedded firewalls with virtual private groups. In *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition*, volume 2, pages 81–83, April 2003.

[16] L. M. Meredith. A summary of the autonomic distributed firewalls (ADF) project. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition*, volume 2, pages 260–265, April 2003.

[17] D. Newman. Benchmarking terminology for firewall performance. RFC 2647, Internet Engineering Task Force, August 1999.

[18] C. Payne and T. Markham. Architecture and applications for a distributed embedded firewall. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 73–80, December 2001.

[19] P. Rubel, M. Ihde, C. Payne, and S. Harp. Generating policies for defense in depth. In *Proceedings of the 21st Annual Computer Security Applications Conference*, December 2005.

[20] S. Rumelioglu. Evaluation of the embedded firewall system. Master's thesis, Naval Postgraduate School, Montery, CA, 2005.

[21] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders, , and P. Pal. Model-based validation of an intrusion-tolerant information system. In *Proceedings of the 23rd Symposium on Reliable Distribued Systems*, pages 184–194, October 2004.