Aplicação de Redundância em Firewall: Análise do CARP como Alternativa a Ataque do Tipo DoS

Fábio Brotto Flores, Marco Antônio Trentin, and Adriano Canabarro Teixeira

Curso de Ciência da Computação, Universidade de Passo Fundo-RS, Brasil
{67558,trentin,teixeira}@upf.br

Resumo. Este artigo tem a finalidade de apresentar os principais conceitos na área de segurança de informação, no que diz respeito as ameaças iminentes à disponibilidade, além da confiabilidade de uma rede de computadores quando trata-se de ataques DoS. É apresentando o conceito de Firewall e analisado o protocolo CARP como alternativa para implementação de redundância em um Firewall e contenção de ataques DoS. Através de simulações de ataques DoS a Firewalls, buscou-se descobrir até que ponto a redundância em um sistema de Firewalls é suficiente para inibir ataques de negação de serviço.

Abstract. This article aims to present the main concepts in the area of information security, especially the imminent threats to availability, and reliability of a computer network when it suffers DoS attacks. Also, the CARP protocol is analyzed as an alternative to redundancy implementation in a Firewall and containment of DoS attacks. Through simulations of DoS attacks to firewalls, we study to what extent the redundancy in a system of firewalls is sufficient to inhibit denial of service attacks.

Keywords: Segurança, Firewall, Redundância, DoS, CARP

1 Introdução

Quando se iniciou a popularização dos computadores pessoais, as redes de computadores também conseguiram um amplo espaço no mercado, provendo soluções que popularizaram ainda mais o uso de computadores no segmento industrial, comercial e doméstico. Esta expansão das redes de computadores fez com que surgissem questões como a exposição de falhas em sistemas que não foram projetados para um ambiente de rede. Atualmente as organizações têm cada vez mais dados críticos trafegando através de suas redes de computadores, além de possuírem dados a serem protegidos. Existem serviços considerados críticos a serem providos pelas organizações que dependem de uma grande disponibilidade destes serviços.

Um dos mecanismos criados para conter as ameaças presentes na Internet foi o *firewall*, que acabou por tornar-se a única via de acesso da rede interna para a Internet. Através desta topologia que transforma o *firewall* em um ponto único de falha, existem medidas para contornar esta questão. Uma delas é aplicação de redundância em equipamentos de redes considerados críticos para o funcionamento de uma rede. Visando a implementação de redundância, surgiu o CARP - *Common Address Redundancy Protocol*, que busca fornecer redundância transparente entre dois ou mais Firewalls.

Esse trabalho objetivou a implementação de um sistema de *firewall* redundante, buscando a disponibilidade tanto em momentos de falhas de hardware, quanto na sobrecarga no caso de ataques do tipo DoS. A partir disto foi analisada a viabilidade e confiabilidade de se ter *firewalls* redundantes através do protocolo CARP como alternativa na contenção de ataques DoS.

2 Conceitos Fundamentais

2.1 Segurança de uma Rede

Segurança em tecnologia da informação é um amplo, complexo e, algumas vezes, um confuso campo de estudo, pois envolve diferentes áreas (redes, sistemas operacionais, aplicações, banco de dados, entre outros), cada uma com seus próprios riscos, ameaças e soluções. Afirma-se que segurança não é uma tecnologia que se pode comprar ou criar, capaz de tornar a rede segura. Isto se baseia na falsa implicação de que “segurança é um estado que se pode alcançar”. De fato o que realmente pode ser feito é administrar um nível aceitável de risco (Hansteen, 2011).

Na última década, na área de TI, aumentaram significativamente o número de incidentes relacionados à segurança da informação nas organizações. Cada vez mais as pessoas e organizações estão dependentes de tecnologias de redes e computadores para diversos propósitos. Porém, ao mesmo tempo, a tecnologia potencializa atividades ilegais ou maliciosas, como a invasão de redes para roubo de números de cartões de créditos, uso de sistemas telefônicos de forma fraudulenta, transmissão de segredos comerciais e propriedade intelectual, *defacement* em *web sites* por diversas razões.

Dentre os cinco conceitos da segurança da informação, segundo os padrões internacionais (ISO/IEC 17799:2005), que são Confidencialidade, Autenticidade, Integridade, Disponibilidade e Não repúdio, a Disponibilidade costuma ser um dos conceitos mais simples de ser comprometido. São facilmente encontrados casos de ataques bem sucedidos, tendo como consequência a indisponibilidade dos serviços na Internet de empresas ou organizações. Segundo Burnett (2002), existem três tipos de perdas que as organizações podem experimentar por causa de lapsos na segurança, são elas:

- **Perda de dados ou segredos:** perda de dados financeiros, comprometimento da integridade dos dados, tendo como consequência o comprometimento de relatórios de dados e acesso não autorizado às informações;
- **Perda de reputação:** após ocorrer uma quebra bem sucedida na segurança, os usuários finais podem abandonar o serviço ou produto por receio de utilizá-los. O efeito que isso tem na avaliação de uma corporação por analistas financeiros faz com que uma avaliação negativa cause um impacto tão grande quanto a própria invasão;
- **Perdas financeiras:** umas das perdas mais difíceis de quantificar, em razão de que ninguém sabe exatamente quantos clientes atuais retornarão após a ocorrência de uma invasão ou, ainda, quantos novos clientes em potenciais nem ao menos tentarão.

2.2 DoS - *Denial of Service*

Ataques de negação de serviço consistem em tornar indisponível um *host* a usuários legítimos, através da inundação de solicitações de serviços enviados por um atacante, na maioria das vezes de forma automatizada e em grande escala. Objetiva sobrecarregar o alvo com requisições que ultrapassem a capacidade de processamento do mesmo.

No início dos anos 90 até os dias atuais, ataques DoS têm amadurecido desde meros aborrecimentos a graves ameaças a disponibilidades de sistemas na área governamental, financeira e econômica. As técnicas de DoS, no final dos anos 90, em sua maioria exploravam falhas em sistemas operacionais relacionadas a implementações do TCP/IP. Algumas explorações conhecidas como "ping da morte", ataque *Smurf*, *Fraggle*, *Boink* e *Teardrop*, foram efica-

zes em derrubar *hosts* individuais com uma sequência simples de pacotes, até que as vulnerabilidades fossem em grande parte corrigidas.

Uma variação do DoS, denominada DDoS (*Distributed Denial of Service*), consiste em ataques DoS distribuídos, utilizando-se de vários computadores que disparam requisições falsas ao mesmo tempo para um determinado *host*, visando tornar indisponível os serviços deste *host*. Nas realizações de ataques DDoS é comumente utilizado computadores zumbis. Estes zumbis são utilizados por redes denominadas *Botnets*, que coordenam e executam ataques DDoS de forma automatizada na Internet.

McClure (2010) comenta sobre um acontecimento significativo de DDoS, ocorrido em conflitos entre a Rússia e a Estônia. É atribuído a Rússia um devastador ataque de DDoS, direcionado a maioria dos serviços *online* da Estônia, causando sérios transtornos.

2.3 Taxonomia de ataques DDoS

RioRey (2011), especializada em ataques DDoS, realizou um estudo que fornece a descrição e a estrutura para classificação e compreensão de ataques DDoS. A seguir são apresentados os principais ataques DDoS:

- **Ataques DDoS baseados em TCP**: SYN *Flood*, SYN-ACK *Flood*, ACK & PUSH ACK *Flood*, ACK Fragmentado, RST ou FIN *Flood*, IP Sinônimo, Sessão Falsa, Ataque de Sessão, Mau Uso de Aplicativos;
- **Ataques DDoS baseados em UDP:** *Flood* UDP, Fragmentação UDP, *Flood* DNS, *Flood* VoIP, *Flood* Dados de Media, *Flood* UDP *Non-Spoofed*;
- **Ataques DDoS baseados em TCP-HTTP:** Fragmentação HTTP, VERB Excessivo, VERB Excessivo de Sessão Simples, Múltiplas requisições de VERB Simples, GET Recursivo, GET Recursivo Aleatório, Aplicação Defeituosa;
- **Ataques DDoS baseados em ICMP**: *Flood* ICMP, Fragmentação, *Flood* Ping.

3 Trabalhos Relacionados

Pesquisas relacionadas a sistemas de *firewalls* redundantes vem crescendo consideravelmente nos últimos anos. A seguir são descritos alguns trabalhos relacionados, em especial os que julgamos serem os mais relevantes.

Em Kraemer (2011) é destacada a importância da tolerância a falhas utilizando protocolos de *Gateway* Redundantes, com o objetivo de comparar e aplicar protocolos de tolerância a falhas dentro do contexto de uma rede corporativa. Para tanto, foram utilizados equipamentos redundantes e protocolos específicos, tais como HSRP (*Hot Standby Router Protocol*), VRRP (*Virtual Router Redundancy Protocol*) e GLBP (*Gateway Load Balancing Protocol*). Nesse trabalho é apresentado um caso de uso real dessas tecnologias em uma empresa do setor público, fazendo um comparativo entre os três protocolos usados na implementação de redundância em roteadores. Ao final, concluem que, entre os protocolos explorados, o GLBP é mais avançado, o VRRP é menos custoso e o HSRP é o mais simples. No caso de uso do setor público, onde não era necessário balancear carga e ao mesmo tempo era importante ter alto desempenho, a implantação foi feita com o HSRP.

Com um propósito um pouco diferente, Botelho (2006) implementou e avaliou, em um ambiente real, técnicas de *Cluster* de Alta Disponibilidade para *Firewall*, utilizando *pfsync* e CARP (*Common Address Redundancy Protocol*) no sistema operacional FreeBSD. Esse trabalho objetivou manter os serviços virtuais disponibilizados o máximo de tempo possível onde, para isso, foram utilizadas técnicas de replicação de arquivos e serviços, e redundância de hardware. Após a implementação do *Cluster* houve um aumento na disponibilidade dos serviços.

O trabalho de Chen (2011) objetivou identificar uma classe de ataques DDoS *Stateful* que ultrapassam soluções baseadas em *cookies*, como em casos de ataques por meio de fragmentação HTTP. Para combater esses ataques, foi proposto um mecanismo de defesa, denominado *Targeted Filtering*, que estabelece filtros em um *firewall,* automaticamente convertendo os filtros para os IP fontes da inundação, deixando o restante do tráfego desbloqueada. Foram experimentadas correções em um mecanismo de defesa proposto, avaliando a eficácia através de análise e simulações para estabelecer os limites de pior desempenho em resposta aos ataques DDoS. Também foi implementado um protótipo em Linux com resultados experimentais que demonstram a eficácia da filtragem por alvo. Ainda, nesse trabalho são comentados vários algoritmos e otimizações, além de provar matematicamente o tempo de convergência do pior caso em relação a um número de sistema por parâmetros de ataques. Os resultados obtidos demonstram a eficácia do mecanismo proposto na defesa contra ataques DDoS *Stateful* em tempo real.

4 Implementando Redundância em Firewall

A fim de analisar a viabilidade e eficiência de um sistema de *firewalls* redundantes, em especial para a contenção de ataques de DoS, foram realizadas experiências a fim de descobrir até que ponto a redundância em um sistema de *firewall* é suficiente para inibir ataques de negação de serviço.

4.1 Protocolo CARP - Common Address Redundancy Protocol

A maioria das redes locais tem seu acesso à Internet através de um único *firewall*, transformando-o no SPOF (*Single Point of Failure*) da rede, de modo que se o mesmo for comprometido, torna indisponível o acesso para dentro da rede e para fora da rede. O CARP é um protocolo que auxilia na criação de redundância, onde vários *hosts* possuem uma única interface de rede virtual entre eles, de modo que se qualquer um dos nós falhar, outro irá responder no lugar, permitindo, além disto, um grau de balanceamento de carga entre os nodos. Assim, o CARP possibilita a implantação de um segundo ou mais *firewalls*. Trabalha da seguinte forma: quando há queda de um *firewall*, com o auxílio do protocolo *pfsync*, todo o estado da conexão é transferido para o *firewall* de *backup*. Nenhuma conexão ativa do TCP será interrompida de forma visível ao usuário final dos serviços de rede. Em um ambiente operacional esta transparência é necessária, pois fornece meios eficazes do administrador da rede tomar uma ação a respeito do problema.

O CARP funciona através de técnicas de replicação passiva, onde são criados um ou mais *backups* de um componente com o objetivo de substituí-lo em caso de falha. Sendo um protocolo *multicast*, são agrupados vários *hosts* físicos em um ou mais endereços IP virtuais. Destes, um sistema é o *master* e responde a todos os pacotes destinados para o grupo de redundância, enquanto o sistema *backup* fica em estado de espera.

O *host* definido como *master* no grupo envia regularmente anúncios à rede local, assim os *hosts backup* sabem qual o estado do *master*; se ele está ou não ativo. Se o *master* tornar-se *offline*, inicia-se o processo de *failover;* os outros hosts dentro do grupo de redundância iniciam os avisos. O *host* que estiver apto a atender mais frequentemente torna-se o novo *master*. Quando o sistema considerado principal volta a tornar-se ativo (*faillback)*, este por

padrão torna-se um host *backup*. A responsabilidade do CARP é na criação e gerência de uma interface de rede virtual. O restante da implementação de redundância, como a sincronização de dados entre aplicações, são utilizados protocolos como *pfsync*, *rsync* ou qualquer outro protocolo apropriado para uma aplicação específica. No caso deste trabalho foi utilizado o protocolo *pfsync* em conjunto com o CARP.

4.2 Protocolo *pfsync*

O protocolo *pfsync* é utilizado pelo *Packet Filter* (*firewall* do OpenBSD) para gerenciar e atualizar o estado das tabelas do *firewall* tipo *Stateful*. O *pfsync* transfere entre os *firewalls* mensagens de inserção, atualização e exclusão de estados. Por padrão, as mensagens de mudanças de estado são enviadas para uma interface de sincronização utilizando pacotes IP *multicast*. O *firewall* que envia as mensagens também fica ouvindo as conexões na interface do *pfsync* por mensagens similares provenientes de outros firewalls na rede.

Nesse experimento foram utilizados três elementos, sendo eles o CARP, o PF - *Packet Filter*, e o protocolo *pfsync* que é utilizado pelo PF para gerenciar e atualizar o estado das tabelas do *firewall*. O ambiente dos testes foi implementado em um ambiente virtual (VMware Workstation versão 7.0.1), composto de quatro *hosts* virtuais, sendo que a topologia do ambiente implementado está de acordo com a Figura 1, com as seguintes especificações:

- **Firewall-1 (Master)**: Nome da máquina Virtual: FW1; Sistema Operacional: OpenBSD 4.8 i386; Memória RAM: 256 MB; Três interfaces de rede virtualizadas: 1º Interface: vic0, Ethernet, em modo *Bridge*; 2º Interface: vic1, Ethernet, em modo *Host-only*; 3º Interface: vic2, Ethernet, em modo *Host-only*.
- **Firewall-2 (Backup)**: Nome da máquina Virtual: FW2; Sistema Operacional: OpenBSD 4.8 i386; Memória RAM: 256 MB; Três interfaces de rede virtualizadas: 1º Interface: vic0, Ethernet, em modo *Bridge*; 2º Interface: vic1, Ethernet, em modo *Host-only*; 3º Interface: vic2, Ethernet, em modo *Host-only*.
- **Cliente da LAN**: Nome da máquina Virtual: Ubuntu; Sistema Operacional: Linux Ubuntu 10.04.3 i686, Kernel 2.6.32; Memória RAM: 1024 MB; Interface de rede: Virtualizada sobre a interface da máquina hospedeira, denominada eth0, Ethernet, em modo *Host-only*.

- **Atacante Externo**: Nome da máquina Virtual: BT5-GNOME-VM-32; Sistema Operacional: Linux Back Track 5 i686, Kernel 2.6.38; Memória RAM: 768 MB; Interface de rede: Virtualizada sobre a interface da máquina hospedeira, denominada eth0, Ethernet, em modo *Host-only*.

A escolha do OpenBSD como sistema operacional dos *firewalls* foi feita com base nos recursos que o mesmo oferece para implementação para alta disponibilidade, como os protocolos CARP e *pfsync* que já vêm nativos na instalação do OpenBSD. Possui o *kernel* com grande foco em segurança e estabilidade, sendo o mais seguro sistema operacional do tipo UNIX, resultado de uma intensa e continua auditoria de segurança no código fonte, além de já ter sido utilizado em outros trabalhos com propósitos semelhantes como os apresentados pelos autores Botelho (2006) e Danhieux (2004).

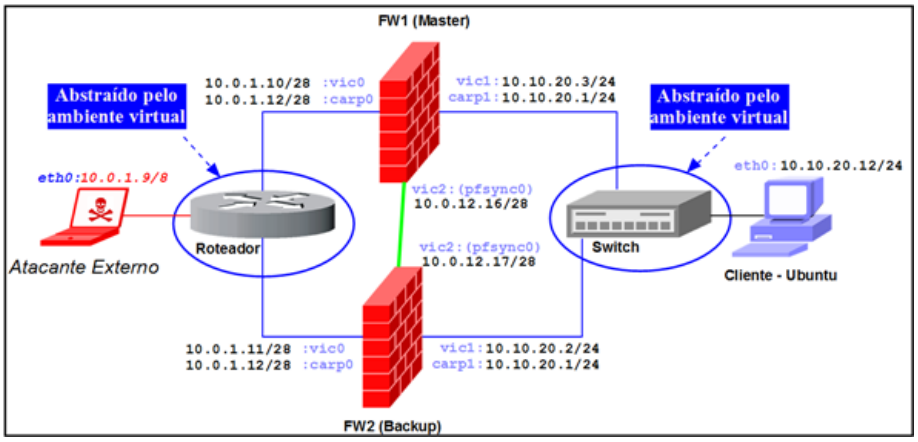

Fig. 1. Topologia da implementação

5 Análise dos Dados e Discussão

Os testes foram realizados com o objetivo de conhecer o limite que em um sistema redundante de *firewall* tem ao estar sob ataque de negação de serviço. Foram realizados testes de *stress*, simulando ataques de DoS, utilizando a ferramenta T50, que é uma ferramenta de injeção de pacotes. Os testes consistiram em disparar o ataque através de um cliente com visibilidade ao *firewall*, buscando com isto estressar o *firewall master*.

5.1 Ataques Realizados e Resultados Obtidos

Seguindo a topologia apresentada na Fig. 1, o ataque foi realizado a partir do *host* do atacante externo, com os seguintes parâmetros no T50:

- Realizado um *flood* com pacotes dos protocolos ICMP, IGMP, TCP, EGP, UDP, DCCP, RSVP e OSPF, sendo disparados simultaneamente;
- Alvo foi o IP do Firewall 10.0.1.12;
- Primeiro disparo de 1.000.000 de pacotes, com média de 40.000 pps;
- Segundo disparo, executando SYN-Flood, a uma média de 40.000 pps com duração de 5 minutos;
- Terceiro disparo, executando um SYN-Flood com múltiplas threads, a uma média de 40.000 pps com duração de 5 minutos;
- Quarto disparo, executando um SYN-Flood com múltiplas threads, a uma média de 40.000 pps com duração de 10 minutos;
- Quinto disparo, executado um SYN-Flood com múltiplas threads, a uma média de 40.000 pps com duração de 15 minutos.

Os resultados foram coletados a partir das ferramentas *iptraf*, *syslog*, *systat* e *pftop*, que são utilizadas para monitorar os recursos do *firewall master* e *firewall backup*. Os resultados obtidos são apresentados abaixo.

Ataque	Nº de Pacotes	PPS	Duração	Falha do Master	Nº de Trocas	Uso de Memória	Bytes IN	Bytes OUT	CPU	Tempo de resposta LAN	Tempo de resposta WAN
1º	998.781	31.088	1 min	Não	0	15,2%	62,52 MB	421B	24,5%	3.709ms	5.421ms
2º	10.329.824	38.495	5 min	Sim	3	32,1%	646,75 MB	0	54,8%	11.996ms	11.996ms
3º	10.695.131	34.837	5 min	Sim	5	67,3%	669,64 MB	0	65,4%	11.136ms	15.623ms
4º	21.928.476	37.977	10 min	Sim	8	72,4%	1,34 GB	0	67,1%	15.632ms	18.649ms
5º	32.779.397	37.421	15 min	Sim	14	87,7%	2,00 GB	0	73,9%	18.159ms	25.223ms

Table 1. Dados obtidos nos testes

5.2 Análise e Considerações dos Resultados

Ao encerrar os testes, análises dos resultados obtidos foram feitas e chegou-se as seguintes considerações:

- durante o 1º ataque, o *firewall master* estava executando um *ping* a um endereço na Internet, o que gerou um tráfego no valor de 421 Bytes de saída, diferente dos ataques restantes onde não houve tráfego de saída;

- foi constatado que houveram várias trocas entre os *firewalls master* e *backup*; isto aconteceu principalmente pelo fato de que, segundo os registro de *log* do *firewall backup*, após o *firewall backup* assumir o tráfego que entrava na rede, o mesmo levou em torno de 15 segundos para receber e tratar o tráfego gerado pelo ataque. Tempo suficiente para que o *firewall master* descartasse os pacotes e reassumisse o tráfego da rede;
- O CARP demonstrou que este comportamento ágil de 15 segundos ocorreu pelo fato de que devido a não ser um ambiente real e apesar de ter sido um ataque com alta intensidade de pacotes, foi tempo suficiente para a limpeza da tabela de estado e descarte dos pacotes maliciosos pelo PF do *firewal master*;
- Para cada troca que ocorreu, onde *firewall backup* assumiu o tráfego durante o ataque, houve uma troca inversa devolvendo o tráfego ao *firewall master*, podendo ser contabilizado o número de trocas total, como sendo o dobro dos valores apresentados na tabela acima;
- Foi constatado que o CARP, sob um ataque de até aproximadamente 10.000.000 de pacotes, a uma média de 40.000 pps, o sistema de *firewall* conseguiu tratar o ataque sem aumento no tempo de resposta, através de trocas de tráfegos constantes com intervalos médio de 15 segundo, entre os *firewalls* pertencentes ao grupo de redundância;
- Existiu uma degradação de desempenho, aumentando o tempo de resposta mais significativo na LAN e WAN, quando o sistema de *firewall* sofreu um ataque com mais de 30.000.000 pacotes a uma média de 40.000 pps.

Apesar da limitação de 40.000 pps imposto pelo enlace da rede, testes com maiores intensidade poderão retratar de forma mais fidedigna a capacidade do CARP em conter ataques do tipo DoS.

A partir dos resultados analisados constatou-se de que para uma melhor contenção de um ataque DoS, além de uma redundância com o objetivo de tolerar as falhas do *firewall*, também são essenciais ter uma largura de banda suficiente para que o enlace de rede não se torne o gargalo antes dos pacotes chegarem ao *firewall*, além de que exigirá recursos significativos de hardware por parte do equipamento que estiver suportando o protocolo CARP, assim como o restante da redundância do *firewall*.

A implementação de um *firewall* redundante para conter um ataque do tipo DoS é uma medida paliativa, fornecendo um tempo necessário para o administrador da rede buscar conter o ataque sem causar uma indisponibilidade aos usuários da rede. Como, por exemplo, através de aplicação de regras mais restritivas no *firewall* durante o tratamento de incidentes, buscando descartar tráfegos provenientes de IP externos com atividades consideradas maliciosas pelo administrador.

A redundância dos *firewalls* pode vir a falhar, caso sofra um ataque massivo em uma proporção que ultrapasse a capacidade da largura de banda da rede ou ainda extrapole os recursos de *hardware* de ambos os *firewalls*. Existe a possibilidade de isto acontecer, pois não há maneiras de prever o quão forte será um ataque a uma determinada rede.

6 CONCLUSÃO

Ataques do tipo DoS são considerados fáceis de serem executados, não exigindo conhecimentos avançados por parte do atacante. Porém, esses ataques tem a capacidade de causarem a degradação do desempenho de uma rede ou ainda uma indisponibilidade completa.

Foram identificados mecanismos capazes de implementar um sistema de *firewall* redundante, buscando manter a disponibilidade da rede. Com base nos mecanismos pesquisados foram escolhidos os protocolos CARP e *pfsync*, além do PF como filtro de pacotes, com o objetivo de implementar uma redundância em um *firewall* buscando conter ataques do tipo DoS.

O protocolo CARP demonstrou ser um protocolo robusto e eficiente para implantação de soluções de redundância em sistemas que provêm serviços de rede, como roteadores e *firewalls*. Com base nestas informações, foi analisado a utilização de redundância de *firewalls* através do protocolo CARP, como uma opção para ataques do tipo DoS. A partir de um ambiente virtual de *firewalls* redundantes foram testados ataques do tipo DoS e analisado o comportamento de um sistema redundante de *firewall* sob estes ataques.

A partir da implementação e dos testes realizados, pode-se verificar que é possível obter algum sucesso em conter ataques DoS, desde que exista um

ambiente preparado para uma resposta efetiva em momentos que estiverem ocorrendo incidentes desta natureza.

No decorrer dos testes, o CARP demonstrou ser uma alternativa paliativa, pois para que haja uma melhor contenção de um ataque DoS, além da redundância para tolerar as falhas do *firewall*, também são essenciais fatores como, largura de banda de rede no enlace da conexão e equipamentos suportando CARP com recursos significativos de hardware.

Conclui-se então que a implementação de redundância para conter ataques do tipo DoS utilizando o CARP, serve mais como medida temporária, fornecendo um tempo necessário para o administrador da rede buscar conter o ataque sem prejudicar a disponibilidade da rede.

Referências

1. AYUSO, Pablo; GASCA, Rafael. Demystifying Cluster-Based Fault-Tolerant Firewalls. In: IEEE Internet Computing, Vol. 13, Nr. 6, 2009.
2. BOTELHO, Marco A. F. A*lta Disponibilidade em Firewall utilizando PFSYNC e CARP sobre FreeBSD.* 2006. Monografia (Especialização em Administração de Redes Linux), Universidade Federal de Lavras, 2006.
3. BURNETT, S; PAINE, S. *Criptografia e Segurança*: O guia oficial RSA. Rio de Janeiro: Campus, 2002.
4. CHEN, Shigang; TANG, Yong; DU, Wenliang. *Stateful DDoS Attacks and Targeted Filtering*. Disponível em: <http://www.cise.ufl.edu/~sgchen/papers/JNCA2006.pdf>. Acesso em: 18/05/2011.
5. DANHIEUX, Pieter. *CARP The Free Fail-over Protocol:* Global Information Assurance Certification Paper. USA: SANS Institute, 2004.
6. HANSTEEN, Peter N. M. *Firewalling with OpenBSD's PF packet filter.* Disponível em: http://rlworkman.net/howtos/OpenBSD_pf_guide.html#CARP. Acesso em: 24/08/2011.
7. KRAEMER, Alessandro ; VILAR, Kaio; GOLDMAN, Alfredo. *Tolerância a Falhas utilizando Protocolos de Gateway.* Disponível em: <http://www.ime.usp.br/~gold/publications/pdf/erad2010.pdf>. Acesso em: 10/07/2011
8. MCCLURE, Stuart M.; SCAMBRAY, Joel S.; KURTZ, George K.; *Hacking* Exposed. 6. ed. USA: Mc Graw Hill, 2010.
9. RIOREY, Inc. *Taxonomy of DDoS Attacks.*2011. Disponível em: <http://www.riorey.com/x-resources/2011/>. Acesso em: 02/09/2011.