Certificação de Firewalls

Daniel Ribeiro Brahm1, SandroAntônio Fernandes1, João Avelino Bellomo Filho 1

1 Universidade Católica de Pelotas

drb@atlas.ucpel.tche.br

saf31@uol.com.br

avelino@conectiva.com.br

Resumo

Este artigo apresenta uma visão geral sobre segurança de redes; onde serão abordados estudos sobre *firewalls* (filtragem de pacotes*)* , suas características e potencialidades, tipos de invasões atuais e, ao final, apresenta−se uma proposta acadêmica de certificação de *firewalls* , através de uma ferramenta que foi desenvolvida a partir desta proposta.

Palavras−chave: Redes, segurança, *firewall* , certificação, *GNU/Linux.*

Abstract

This article presents a preview on networks security; where a study on firewalls (filtering of packages), its current and potentialities, types of attacks and to the end are presented a proposal academic of certifys of firewalls, tool that was developed to leave of this proposal.

Keywords: Networks, security, firewall, certification, GNU/Linux.

1 Introdução

Nos dias de hoje o uso de redes de computadores é, cada vez mais uma realidade comum a um maior número de pessoas.

De acordo com Cheswick [CHE94] o crescimento do e−*business* , o surgimento do *e−cash* , o aumento do número de usuários e conseqüentemente o crescimento de uma categoria de usuários mal intencionados denominados *hackers* /*crackers* que, através da própria *Internet* , possuem um fácil acesso a material que contém informações sobre como invadir redes, roubar dados e uma relativa facilidade em usar redes desprotegidas como base de lançamento de ataques, nos proporcionam uma visão clara das possibilidades e problemas que podem advir dessa situação.

Ao se observar os fatos descritos acima começa a ficar claro que uma das maiores preocupações do administrador de redes é a segurança, conforme Ranun [RAN98], mesmo porque já há uma base legal (nos Estados Unidos), que responsabiliza o administrador da rede por danos causados por negligência quanto à segurança da rede.

A dificuldade começa ao se tentar definir qual o padrão (política) de segurança que se deseja para a rede, e é aí que reside o maior desafio do administrador, implementar uma política efetiva de segurança, a qual perpasse todos os elementos que fazem parte e interagem com essa rede, onde a segurança não se limite a dispositivos específicos, mas que também seja levada em conta a segurança física dos equipamentos e a educação e a conscientização dos usuários.

Tomando essas precauções, o administrador não poderá ser acusado de omissão, o que não nos permite dizer que determinada rede é completamente segura, afinal, com certeza pode−se afirmar que uma estrutura é insegura, porém garantir o inverso é colocar o cargo de administrador de rede a prêmio.

É nesse cenário que situa−se este artigo, uma área crucial para o contínuo desenvolvimento e crescimento das redes, e com necessidade de soluções que permitam que esse crescimento seja mantido.

Durante o período de preparação deste artigo, após buscar modelos de certificação os quais ir−se−ia utilizar e adaptar para as nossas necessidades, verificou−se que apresentavam soluções combinadas de

hardware e *software* ou apenas *software* , que não diziam de que forma se estabelecer um conjunto de regras para certificação. A partir daí começou−se a procurar por um modelo conceitual para certificação de *firewalls* , que não se detenha exclusivamente em uma tecnologia específica, mas que leve vários aspectos em consideração para se chegar ao que se pretendia, ou seja,uma proposta de certificação que utiliza como premissa o uso de *software* livre, uma solução não proprietária e sem custos para sua implantação.

Neste contexto foi desenvolvido e implementado o CertFire que permite ao administrador de rede verificar se o(s) *firewall* (s) e o(s) *host* (*s*) de sua rede estão operando de maneira adequada, ou seja, de maneira segura. Este programa permitirá analisar o funcionamento e garantir um estado de segurança de acordo com os parâmetros definidos pelo administrador via configuração e uso da ferramenta.

Dessa maneira definiu−se os objetivos deste artigo, estudar uma ferramenta importante para as redes de computadores, o *firewall* , e propor e implantar um modelo de certificação para esse dispositivo, de maneira que se possa fazer uma análise do funcionamento de um *firewall/host* e demonstrar se o mesmo está funcionando a contento, ou seja, se ele é capaz de propiciar e fazer parte de uma política de segurança eficiente.

2 Firewalls: IPChains e TIS

Antes de se definir a Certificação, foi necessária a implantação de um *firewall* , para que dessa forma, se pudesse definir uma proposta. Para isso, foram escolhidas duas possíveis soluções: o IPChains e o TIS.

Segundo Chapman [CHA95], o IPChains é uma reconstrução do código do *GNU/Linux* IPv4 *firewalling* (que foi baseado no BSD) e uma reconstrução do *ipfwadm* , que foi uma reescrita do *ipfw* do sistema operacional BSD. Ele necessita, para administrar a filtragem de pacotes de um *kernel* 2.1.102 ou superior.

O motivo dessa ferramenta necessitar de um *kernel* atualizado, deve−se a que, nos antigos códigos de filtragem de pacotes, os fragmentos não eram tratados, não permitiam a especificação de outros protocolos como UDP ou ICMP.

O IPChains insere e apaga regras da parte responsável pela filtragem de pacotes do *kernel* . Isso significa que não há armazenamento permanente das regras pelo *kernel* , sendo perdidas no caso de *reboot* . Para que essa perda de regras não acabe com a filtragem e a interligação entre ambas as redes, será necessário criar um *shell script* que repasse essas regras novamente ao *kernel* . Isso será feito no momento da carga do sistema e no momento anterior da carga das interfaces de rede.

O IPChains é uma ferramenta versátil e que pode ao contrário dos servidores *proxy* ser usada para todos os tipos de serviços de redes. Sua configuração requer certa prática. O ideal seria a combinação entre um servidor *proxy* e um filtro de pacotes.

Conforme foi defindo por [EWY96] o TIS *Internet Firewall Toolkit* (que vamos nos referir como TIS ou *toolkit*) é um conjunto de programas e configurações projetado para facilitar a construção de *firewalls* . Os componentes do TIS, embora desenhados para trabalhar de forma conjunta, podem ser usados isoladamente ou podem ser combinados com outros componentes de *firewall* . O *software* foi projetado para ser usado em sistemas UNIX usando TCP/IP com uma interface *Berkeley style* "*socket* ".

A instalação do TIS pressupõe experiência prática com sistemas UNIX e redes TCP/IP. Pelo menos o administrador do *firewall* deve ter familiaridade para instalá−lo e mantê−lo funcionando em um sistema UNIX. Também é necessário conhecimento de como compilar pacotes usando "*make* ". O *toolkit* não provê uma instalação padrão, já que para cada topologia de rede, *hardware* disponível e práticas administrativas existirão necessidades diferentes.

Dependendo de como o *toolkit* for configurado, diferentes níveis de segurança poderão ser conseguidos. As configurações de segurança mais rigorosas poderão servir para a maioria dos casos, enquanto em outros, algo menos rigoroso, não servirá para a maioria dos casos.

Na verdade, esta é a função do administrador do *firewall* : Entender o que é necessário proteger, entender quais são os riscos aceitáveis e inaceitáveis, e racionalizá−los com as necessidades dos usuários. Esta análise é a tarefa mais difícil na implementação de um sistema de segurança.

O TIS é um conjunto de ferramentas poderoso, que inclui ferramentas de autenticação (que não foram abordadas neste artigo), é flexível e configurável, o seu uso vai depender da habilidade e conhecimento do administrador do *firewall* em definir o que é necessário para sua rede, e como fazê−lo. Como todo servidor

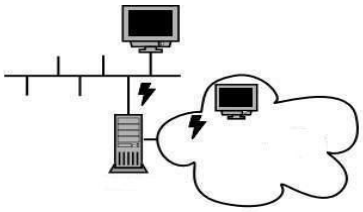
proxy o TIS enfrenta o problema de que nem todos os serviços passam pelo *firewall* , há opções como usar o *plug−gw* que é um servidor *proxy* genérico, o que no entanto acarreta o problema de que o tráfego que está passando não é analisado, o que pode trazer dores de cabeça para o administrador. Outra solução é buscar *proxys* escritas por terceiros que funcionam com o TIS.

Mas o que realmente se destacou neste trabalho com o *toolkit* foi a facilidade de configuração e a quantidade razoável de documentação disponível.

3 A Certificação

Certificação ou o ato de certificar significa dizer que: está se fazendo algo obedecer a um conjunto de parâmetros pré−definidos para que dentro de uma realidade tal, possa−se determinar os resultados esperados. No contexto deste artigo, a certificação atua como um conjunto de regras que serão definidas para certificar um *firewall* , ou seja, garantir que um microcomputador que será responsável pela segurança de uma rede local possa oferecer condições para isto.

Rede protegida
Firewall



Escopo de uma rede protegida por um Firewall

Antes de se definir formalmente este modelo de certificação deve−se afirmar que, este modelo é acadêmico e, portanto, não possui finalidades comerciais. Este modelo conceitual não é a palavra final em certificação de *firewalls* para o S.O *GNU/Linux* (ou outras plataformas), mas pode ser visto como um primeiro passo para tal. Devido as suas características que serão descritas a seguir, este modelo não se restringe apenas à aplicação em *firewalls* mas também a *hosts* .

Este modelo poderá ser útil como referência para futuros trabalhos que envolvam o tópico de segurança em redes de computadores.

Ao se passar pelas dificuldades iniciais, resolveu−se procurar por ferramentas que pudessem viabilizar o modelo de certificação. Primeiramente, buscou−se ferramentas que verificam segurança em nível de portas e logo após uma ferramenta para análise local da máquina que será utilizada como *firewall* .

Este modelo de certificação foi viabilizado através do uso de ferramentas de análise e auditoria, as ferramentas usadas são: *Tiger* , *Nmap* , *Exscan* .

No caso da certificação remota, ao se utilizar as ferramentas *Nmap* e *Exscan* , irá se determinar através de configuração do programa que implementamos a proposta de certificação quais portas deverão estar obrigatoriamente abertas, que serão as portas dos serviços permitidos no *firewall* . Se o resultado desse "*port scaning* " for compatível com o que foi configurado no programa, o *firewall* estará certificado em nível de portas, não existirá meia certificação, isto é, no caso ser retornado um resultado não compatível em apenas uma porta não estará garantida a certificação.

Na certificação local será configurado através do programa os *scripts* de análise de permissão e integridade do *Tiger* . A certificação quando feita na própria máquina abrangerá tanto o estado das portas quando as permissões e integridades relativas ao sistema de arquivos, resultando em uma análise mais profunda quando ao estado geral do *firewall* /*host* .

Poderá acontecer que programa implementado para realizar a certificação atinja os resultados esperados em nível de portas e não obtê−lo quando da análise das permissões e integridades no sistema de arquivos e vice−versa, nesta situação a máquina não será considerada certificada, devendo ser efetuados ajustes necessários, conforme demonstração apresentada pelo arquivo de *log* do programa para certificação.

Após a execução, será possível obter no programa uma descrição detalhada das análises que foram

feitas, permitindo observar o resultado das análises feitas que levariam ou não a certificação, e no caso de falhas permitindo a sua correção.

Estes são os aspectos considerados no modelo de certificação ressaltando sempre, que não está se propondo aqui uma solução definitiva, e sim um modelo conceitual de certificação.

4 A implementação da proposta de Certificação

Conforme foi definido anteriormente, o nosso modelo de certificação passa pela utilização de três ferramentas de auditoria de segurança, tais como: *Nmap* , *Exscan* , responsáveis por análise do estado de portas TCP e o *Tiger* responsável pela análise de segurança no sistema de arquivos. Ambas gerenciadas por um programa escrito em C++ com a biblioteca Qt [DAL99] , denominado CertFire (http://certfire.codigoaberto.org.br).

Esse modelo de certificação foi definido da seguinte forma:

- Usuário do programa de certificação definirá através de configuração no próprio programa quais portas devem apresentar o estado aberto e quais estarão no estado fechado. Caso alguma dessas condições não seja atendida, será assinalada uma falha no processo de certificação.

- Logo após esta análise, começará o processo de auditoria no sistema de arquivos na qual a ferramenta *Tiger* fará uma varredura à procura de falhas, como: tipo de acesso a arquivos (permissões), alteração não autorizada em arquivos de configuração e alguns sinais comuns de intrusos que podem ser deixados no sistema, e que o *Tiger* pode facilmente detectar. Caso seja encontrada uma falha será reportado um problema no processo de certificação.

O processo de certificação só será bem sucedido se ambos os testes descritos anteriormente forem concluídos com sucesso (sem ocorrência de falhas).

As ferramentas utilizadas para o controle na situação de estado de portas e a de auditória no sistema de arquivos não são executados da mesma forma como se estivessem sidos executados no terminal (modo console), pois a sintaxe de utilização das ferramentas, não é visível ao usuário, ficando visível apenas o tipo de análise que será feita, e de que forma será feita. Porém tudo isto é feito com uma vantagem, os parâmetros utilizados para funcionamento destas ferramentas são determinados no próprio programa para certificação, de maneira intuitiva, graças a sua interface gráfica.

As ferramentas que verificam a situação a nível de estado de portas funcionam da seguinte forma, se, por exemplo, as portas marcadas para análise forem: FTP *(File Transfer Protocol*) e TELNET, estas (duas portas devem estar no estado "aberto" simultaneamente, caso uma não esteja, já será apontado um erro pelas ferramentas) devem estar no estado "aberto" para que as ferramentas retornem um resultado favorável, ou seja, sucesso. Este resultado poderá ser confirmado através dos arquivos de *log* gerado por elas, que pode ser visualizado no próprio programa. Por exemplo, se os serviços FTP e TELNET estiverem ativos, no *log* da ferramenta *Nmap* será exibido o seguinte resultado:

```
Port       State       Service
21/tcp     open        ftp

23/tcp     open        telnet
```

Este resultado comprova que os serviços FTP e TELNET estão ativos. Caso não estivessem, logo após os cabeçalhos (*Port* , *State* e *Service*) seriam deixadas duas linhas em branco, comprovando que os serviços não estão disponíveis.

Da mesma forma a ferramenta *Exscan* comprova o estado dos serviços (neste caso o FTP e TELNET), através do seu *log* , que segue abaixo:

```
Port   21 Open: File Transfer Protocol Service Running.
Port   23 Open: Telnet Service Running.
```

Assim como o *Nmap* , se estes serviços não estivessem disponíveis seriam deixadas apenas linhas em branco no arquivo de *log* , comprovando que os serviços não estão disponíveis.

No caso da ferramenta *Tiger* , são feitas várias análises de segurança no sistema de arquivos, caso seja

apontada uma falha, está será parte integrante do seu arquivo de *log* da própria ferramenta e do programa de certificação. Se logo após a descrição da análise que foi realizada, não for assinalado nenhum erro no *log* do CertFire após a especificação do tipo de análise, a próxima linha será deixada em branco, comprovando que aquela análise foi realizada e concluída com sucesso, caso contrário será especificado nesta linha ou nas próximas o tipo de erro e a falha que aconteceu apontado o número da linha em que o erro foi encontrado no *log* da ferramenta *Tiger* . Um exemplo de análise do *lo* g da ferramenta *Tiger* , pode ser visto neste fragmento de seu *log* :

```
# Performing check of anonymous FTP...
−−WARN−− [ftp006w] Anonymous FTP enabled, but directory does not exist.

# Performing check of ’services’ and ’inetd’...

# Checking services from /etc/services.
−−FAIL−− [inet002f] Service echo is assigned to port 4/ddp which should be
```

Neste fragmento, temos uma visão clara do que o programa (CertFire) irá analisar, sempre que é descrita uma análise, a respectiva linha é iniciada com o caractere "#" e sempre que for apontada uma falha a linha começará por dois hífens "− −" e logo após a falha que ocorreu. Toda vez que o CertFire encontrar os dois hífens no início da linha no *log* da ferramenta *Tiger* , será apontada uma falha na análise em questão, e logo após ela será reportada no seu *log* , que irá descrever o tipo de análise que foi efetuada e o número da linha onde houve erro. Se logo após a descrição da análise (linha que começa com o caractere "# ") for deixada uma linha em branco no *log* do *Tiger* , isto comprova que não foi encontrado nenhum problema, portanto no *log* do CertFire, irá apenas aparecer a análise que foi realizada, e logo após será deixada uma linha em branco.

Neste contexto, o programa que foi desenvolvido representa o nosso modelo conceitual de certificação o qual recebeu a denominação de CertFire versão 0.1 e foi por meio dele que expressaram−se todas as regras necessárias para o processo de certificação e onde também empregamos as ferramentas de análise (*Nmap* , *Exscan* e *Tiger*) que deram apoio e sustentação a este processo. Sua essência é escrita em código C++ com suporte gráfico da biblioteca Qt (versão 1.44). Por trabalhar com esta biblioteca, seu funcionamento dependerá de um ambiente gráfico instalado.

O programa é voltado para o ambiente de rede tanto para gerentes e administradores de redes preocupados com os tipos de serviços autorizados no seu *firewall* e a integridade de seu sistema de arquivos, como para usuários domésticos preocupados com a segurança de seu *host* (máquina cuja finalidade é executar as aplicações do próprio usuário). Sua interface gráfica permite um fácil manuseio e interpretação de resultados tanto para usuários de nível básico quanto avançado.

A sua função é permitir que qualquer indivíduo que possua algum razoável conhecimento de redes de computadores possa executar testes de segurança em qualquer computador (*firewall* ou *host*).

Tem por finalidade ser de fácil entendimento por parte do usuário, pois à parte de mais baixo nível, como a interpretação de resultados gerados pelas ferramentas é feita pelo CertFire.

Todas as configurações de análise são feitas pelo usuário, mas com um benefício, todos os itens possuem uma ajuda rápida (*tool tips*), apenas posicionando−se o mouse sobre o item desejado será apresentado uma descrição do mesmo, caso o usuário não tenha idéia de quais configurações utilizar, o próprio programa oferecerá uma configuração padrão (recomendada) que abrangerá todos os principais pontos relevantes para o processo de certificação. O Processo de certificação começa da seguinte forma:

1. Executar primeiramente o programa na máquina alvo dos testes, pois o teste de análise de segurança no sistema de arquivos só poderá ser realizado na máquina local e não remotamente. Desta maneira é feita análise tanto a nível de portas quanto de sistemas de arquivos; este será o principal teste para o processo de certificação.

2. Logo após, deve−se executar o aplicativo numa máquina pertencente à mesma rede que a maquina alvo esteja localizada. Desta maneira será feita a análise de portas cujo resultado deve ser igual ao do teste anterior, sempre que preservando as mesmas configurações do CertFire.

3. Por fim deve−se rodar o programa numa rede distinta da máquina em análise para se poder observar o comportamento da máquina alvo, quando submetida a teste de estado de portas fora do seu ambiente normal de rede, que deverá ser rigorosamente igual ao dos passos anteriores, lembrando sempre que se deve manter

a mesma configuração para a execução dos testes.

Depois de se ter seguido os três passos anteriores, cabe ao usuário a interpretação dos resultados que serão apresentados da seguinte forma: quando o teste for realizado na máquina alvo, serão apresentadas mensagens que irão demonstrar se a máquina está ou não certificada conforme os padrões (de estado das portas TCP e da integridade do sistema de arquivos) que foram configurados pelo usuário, se não for positivo o usuário deve recorrer aos *log's* das ferramentas coadjuvantes no processo de certificação e ao *log* do próprio CertFire para correção do(s) possível(eis) erro(s), logo após deve−se defrontar os resultados dos passos dois e três que demonstrarão a situação da máquina em nível de portas, se caso todos os passos forem de aprovação a conclusão será que a máquina estará certificada de acordo com os padrões conceituais empregados pelo programa (os quais foram descritos anteriormente nas seções anteriores) de certificação e devidamente configurados pelo usuário, caso seja assinalada alguma(s) falha(s) recomenda−se a correção dos erros e uma nova execução da ferramenta para comprovação das correções e para uma possível obtenção do *status* (estado) da máquina certificada.

5 Conclusão

Ao final deste artigo, verificou−se que quando o assunto é segurança não existem soluções prontas ou mágicas. Para diferentes situações existirão diferentes problemas a serem enfrentados, e é neste momento que o administrador de redes deve entrar em ação planejando a política de segurança da rede sob seu domínio.

Uma política de segurança não se resume a dispositivos como o estudado neste artigo (o *firewall*), é óbvio que o *firewall* é uma importante ferramenta de segurança, mas não a resposta para tudo, a segurança passa pela conscientização dos usuários, pelo constante aperfeiçoamento do administrador, pelas atualizações de *software* e *hardware* quando necessárias, ou seja, é um processo que nunca tem fim. Na realidade pode−se estar protegido dos perigos já conhecidos, mas para enfrentar os problemas que porventura surgirem, o administrador tem de estar preparado, como diz a frase de autor desconhecido "o preço da paz é a eterna vigilância".

Neste contexto, esboçou−se uma proposta para garantia de segurança em redes de computadores, isto é, uma proposta (metodologia) de certificação que visa viabilizar um estado de segurança dentro de uma rede. Ao iniciarmos a jornada sobre a Certificação de *Firewalls* , buscamos por modelos de certificação prontos, onde através dos quais poderíamos adaptá−los para as nossas necessidades, mas infelizmente não foi possível, pois todas as soluções neste sentido eram soluções proprietárias.

A partir disso, começou−se a se procurar por ferramentas usadas por administradores de redes para verificar a segurança em suas redes, vide [SEG]. Encontramos três ferramentas; *Nmap* e *Exscan* que foram utilizadas neste artigo, que são do tipo "*Port Scaning* " e uma outra responsável pela auditória e análise de integridade em sistema de arquivos denominada *Tiger* . Com estas ferramentas, vislumbrou−se uma proposta de certificação.

De posse destas ferramentas implementou−se um programa, que esta sob a licença GPL, denominado CertFire para demonstrar na prática a proposta de certificação.

Ao terminar−se esta implementação, constatou−se que o programa possui algumas limitações como, por exemplo: Não ter a capacidade de detectar possíveis tentativas de invasão que possam estar ocorrendo em tempo real, limitação em número de portas para varredura (*scan*), etc.

Por isso, uma possível sugestão para futuras implementações do CertFire, seria portá−lo para uma interface WEB, isto é, onde o seu ambiente natural seria um navegador (*browser*) que permitiria sua operação em qualquer S.O, com suporte ao protocolo HTTP. Aliado a isto, uma outra área que poderia colaborar através de suas características é a inteligência artificial (IA), a qual poderia oferecer subsídios e mecanismos para uma melhor análise e interpretação de *log's* das ferramentas auxiliares. Qualquer pessoa que quiser compartilhar do desenvolvimento do CertFire será bem vinda, bastando apenas visitar o seu site em http://certfire.codigoaberto.org.br

É válido ressaltar que a metodologia de certificação que foi usada, com o passar do tempo pode se tornar ineficaz, para uma realidade futura. Por isso, é importante compreender que possíveis alternativas de certificação podem e devem ser adotadas e adaptadas, para que o processo de certificação em si seja válido. Embora, hoje, não exista de maneira clara e definida metodologias de certificação, cabe ao administrador de rede estar sempre atualizado sobre tópico *security* , para que possa sempre tomar decisões coerentes para proteger os *hosts* que estão sob a sua responsabilidade.

E, então, neste paradigma nossa proposta de certificação, um trabalho acadêmico com suas limitações, voltado para uma situação específica, não se propondo a ter uma abrangência genérica, vem para fornecer

subsídios para futuros trabalhos, podendo ser considerado como um primeiro passo para outros alunos/profissionais que resolvam desenvolver projetos na área de segurança de redes. Neste artigo não foram abordados os tipos de ataques (vírus, worms, e vulnerabilidades) embora tenham sido tratados de modo abrangente em nossa monografia de conclusão de curso.

6 Bibliografia

[CERT] CERTIFICAÇÃO ICSA Firewall Test Criteria disponível em http://www.icsa.net/html/communities/firewalls/certification/criteria/index.shtml

[CHA95] CHAPMAN, Brent, ZWICKY Elizabeth, *Building Internet Firewalls*, 1ª ed., O'Reilly & Associates; 1995. 544p.

[CHE94] CHESWICK, William R., BELLOVIN, Steven M., *Firewalls and Internet Security : Repelling the Wily Hacker* , Addison−Wesley, 1994, 306p.

[DAL99] DALHEIMER, Matthias Kalle, *Programação em Qt* , Ed. Ciência Moderna, Rio de Janeiro, 1999.

[EWY96] EWY, Benjamin, *Creating a Linux Firewall using the TIS Toolkit* , Linux Journal, 25, May 1996.

[IBM] IBM Software Security SecureWay Firewall Library disponível em http://www−4.ibm.com/software/security/firewall/library/

[ICSA98] ICSA, *3rd Annual Firewall Industry Guide*, disponível em http://www.icsa.net/fwbg/, 1998.

[LUZ96] LUZZARDI, Paulo Roberto Gomes: *Linguagem de programação C* , EDUCAT, Editora da Universidade Católica de Pelotas, Pelotas, 1996

[RAD97] RADER, Mark, BIRDWELL, J. D, *Public/Private/Wireless Information Security. A blue print for safeguarding sensitive informations.* In: ONDPC/CTAC International Symposium, Chicago, Illinois, August 18−22, 1997 pp16−28. (Paper)

[RAN98] RANUN, Marcus, CURIN, Matt, *Internet Firewalls Frequently Asked Questions*, disponível em http://www.clark.net/pub/mjr/pubs/fwfaq/, 1998.

[SEG] Segurança de redes disponível em http://penta.ufrgs.br/gere96/segur/ferram.htm#satan

[TAN96] TANENBAUM, Andrew S. Tanenbaum, *Computer Networks* , 3ª ed., Upper Saddle River, Prentice Hall, 1996. 814p.