



Edição 06 :: Ano I



**ESPECIAL: Prepare-se para as certificações LPI**  
A certificação Linux mais procurada na atualidade

**Acesso Remoto e Seguro também nas PMEs**  
A importância das soluções de VPN

**Salve seus dados com Bacula**  
Backup centralizado com  
uma ferramenta open source

**Segurança e IPv6**  
O IPv6 é mais seguro que o IPv4?

# SNORT

## Detectando e prevenindo intrusos

**Soluções com o  
Windows PowerShell**  
Uma poderosa ferramenta  
para automatização

**Supercomputação  
e virtualização**  
Processadores com múltiplos  
núcleos no suporte à virtualização



# TENHA ACESSO ILIMITADO A TODO CONTEÚDO DO SITE DEVMEDIA

## SEJA UM ASSINANTE **MVP!**

Tenha acesso garantido a qualquer tema  
que você possa necessitar!



### VIDEOS - AULAS

O assinante MVP acessa todas as vídeo-aulas do site DevMedia. Nenhum outro produto dá essa possibilidade. São mais de **7.000 vídeo-aulas**.



### CURSOS

Nosso site possui atualmente **+ de 250 cursos online** com a qualidade que você já conhece e confia. Com o MVP você terá acesso a todos eles.



### REVISTAS

Assinante MVP acessa todas as nossas revistas na versão digital, incluindo todas as edições anteriores. Quando pintar aquela dúvida você terá mais de **3.500 artigos para consultar!**



A assinatura MVP te dá a oportunidade de aumentar seu conhecimento em diversos assuntos, com qualidade e economia.

**Invista na sua carreira se torne um profissional ainda mais completo!**



DevMedia

POR APENAS  
**R\$59,90** POR MÊS

Acesse: [www.devmedia.com.br/mvp](http://www.devmedia.com.br/mvp)

# Sumário

Olá, eu sou o **DevMan**! Desta página em diante, eu estarei lhe ajudando a compreender com ainda mais facilidade o conteúdo desta edição. Será um prazer contar com sua companhia! Confira abaixo o que teremos nesta revista:



Infra

## 06 – Prepare-se para as certificações LPI

A certificação Linux mais procurada na atualidade

[ *Leonardo Afonso Amorim* ]

Infra

## 12 – Soluções com o Windows PowerShell

Uma poderosa ferramenta para automatização

[ *Leandro Marcos* ]

Infra, Segurança,  
Web

## 19 – Segurança e IPv6

O IPv6 é mais seguro que o IPv4?

[ *Bruno Bogaz Zarpelão* ]

Segurança

## 26 – Detectando e prevenindo intrusos com Snort

Ferramentas de detecção de intrusão adicionam uma nova camada de segurança à rede

[ *Renato Carneiro Pacheco* ]

Segurança, Infra

## 35 – Salve seus dados com Bacula

Configuração e instalação – Parte 2

[ *Flávio Alexandre dos Reis e Eduardo Pagani Julio* ]

Segurança, Infra

## 45 – Acesso Remoto e Seguro também nas Pequenas e Médias Empresas

A importância das soluções de VPN

[ *Roberto Henrique de Sousa* ]

Infra

## 51 – Supercomputação e virtualização

A utilização de processadores com múltiplos núcleos no suporte à virtualização

[ *André Leon S. Gradvohl* ]

**[Infra]** Artigo que apresenta tecnologias relacionadas a questões de Infraestrutura.

**[Segurança]** Artigo dentro do contexto de Segurança: redes, servidores, sistemas operacionais, infraestrutura em geral.

**[Web]** Artigos sobre ou que envolvam tecnologias de infraestrutura para WEB.

## Edição

### Editor

Eduardo Spínola ([eduspinola@gmail.com](mailto:eduspinola@gmail.com))

### Sub Editores

André Leon S. Gradwohl ([gradwohl@ft.unicamp.br](mailto:gradwohl@ft.unicamp.br))

Marco Antônio Pereira Araújo ([maraujo@devmedia.com.br](mailto:maraujo@devmedia.com.br))

Patrícia Lima Quintão ([pquintao@devmedia.com.br](mailto:pquintao@devmedia.com.br))

## Produção

Jornalista Responsável Kaline Dolabella - JP24185

## Atendimento ao leitor

A DevMedia possui uma Central de Atendimento on-line, onde você pode tirar suas dúvidas sobre serviços, enviar críticas e sugestões e falar com um de nossos atendentes. Através da nossa central também é possível alterar dados cadastrais, consultar o status de assinaturas e conferir a data de envio de suas revistas. Acesse [www.devmedia.com.br/central](http://www.devmedia.com.br/central), ou se preferir entre em contato conosco através do telefone 21 3382-5038.

## Publicidade

[publicidade@devmedia.com.br](mailto:publicidade@devmedia.com.br) – 21 3382-5038

**Anúncios** – Anunciando nas publicações e nos sites do Grupo DevMedia, você divulga sua marca ou produto para mais de 100 mil desenvolvedores de todo o Brasil, em mais de 200 cidades. Solicite nossos Media Kits, com detalhes sobre preços e formatos de anúncios.

### Fale com o Editor!

É muito importante para a equipe saber o que você está achando da revista: que tipo de artigo você gostaria de ler, que artigo você mais gostou e qual artigo você menos gostou. Fique a vontade para entrar em contato com os editores e dar a sua sugestão! Se você estiver interessado em publicar

um artigo na revista ou no site Easy Java Magazine, entre em contato com o editor, informando o título e mini-resumo do tema que você gostaria de publicar:

Eduardo Spínola - Editor da Revista  
[eduspinola@gmail.com](mailto:eduspinola@gmail.com)

# Editorial

**M**anter uma rede segura e sem dores de cabeça é o sonho de todo administrador. Infelizmente, por diversos motivos, muitas pessoas acabam tomando atitudes que visam prejudicar o funcionamento da rede de uma empresa. Por causa disso, precisamos estar preparados para os problemas que possam surgir. Assim como destacamos no último editorial, não espere que algo de ruim aconteça para reagir. Esteja preparado para os desafios, e um deles, evitar ataques à rede, pode ser amenizado com a ferramenta Snort.

Deste modo, no artigo **Detectando e prevenindo intrusos com Snort**, serão expostos os conceitos da ferramenta Snort, seu histórico e definições de IDS. Também serão mostradas suas funcionalidades, exemplos práticos e, por fim, recomendações de boas práticas de segurança no uso em produção.

**Soluções com o Windows PowerShell** retrata uma abordagem simplificada de primeiro contato com o novo shell e linguagem de script da Microsoft, com a finalidade de apresentar e exemplificar sua utilização no dia a dia do administrador de redes.

A matéria **Prepare-se para as certificações LPI** procura esclarecer o leitor quanto às dúvidas em torno da certificação LPI, abordando desde seu histórico de evolução no Brasil, como são compostas e divididas as provas por nível, o local em que se pode realizá-las e também dicas para quem quer começar a estudar para as provas.

**Segurança e IPv6** abordará as principais questões relacionadas à segurança em redes IPv6. Primeiramente, serão descritas as principais novidades do IPv6 com relação ao IPv4. Também serão apresentados o framework de segurança IPSec, cujo suporte é obrigatório nos nós das redes IPv6, e as principais vulnerabilidades encontradas nestas redes. Por fim, serão analisadas soluções de segurança que podem ser adotadas para amenizar o risco trazido pelas vulnerabilidades presentes nas redes IPv6.

O artigo **Salve seus dados com Bacula – Parte 2** dará sequência ao estudo sobre a ferramenta Bacula, demonstrando suas formas de instalação e configuração. Também serão apresentados os modos de instalação e configuração com o MySQL e apresentadas as variantes para instalação do PostgreSQL e SQLite como bases de dados para gerenciar as rotinas de backup.

Em **Acesso Remoto e Seguro também nas Pequenas e Médias Empresas**, analisaremos as principais razões para a adoção massiva do acesso remoto seguro nas Pequenas e Médias Empresas por meio das soluções de VPN, as características técnicas dos diferentes tipos de acessos e exemplos de produtos acessíveis a este mercado.

Por fim, a matéria **Supercomputação e virtualização** destacará a utilização de processadores com múltiplos núcleos no suporte à virtualização. Imagine os problemas que um ambiente virtualizado com baixo desempenho pode provocar. A supercomputação pode ser a solução para todos eles.

Eduardo Oliveira Spínola

[eduspinola@gmail.com](mailto:eduspinola@gmail.com)

[twitter.com/Java\\_Magazine](https://twitter.com/Java_Magazine)



# Seu conhecimento vale muito na DevMedia

Publique artigos e vídeos gratuitamente no site e participe do programa **PayContent**.

Sua participação pode ser premiada: quanto mais você postar, mais chances de ganhar.



Para saber mais acesse:  
[www.devmedia.com.br/paycontent](http://www.devmedia.com.br/paycontent)



# Prepare-se para as certificações LPI

## A certificação Linux mais procurada na atualidade

A certificação Linux mais procurada na atualidade é a criada pela LPI – Linux Professional Institute. A LPI é uma organização sem fins lucrativos, que possui sede na Califórnia e que foi constituída em 1999 pela comunidade Linux com a finalidade de desenvolver um programa de certificações GNU/Linux reconhecido mundialmente por empresas e profissionais de TI. Foi a primeira organização a defender e ajudar o uso profissional de Linux, do Open Source e do Free Software.

O grande diferencial da certificação LPI é que a mesma não foca exclusivamente em uma distribuição Linux, sendo independente de fabricantes e fornecedores de solução. É neutra, pois se baseia no padrão LSB – Linux Standard Base (Base Padrão do Linux), criado para diminuir os custos envolvidos com suporte em plataformas Linux. Para conseguir isso a ideia é diminuir as diferenças entre as distribuições Linux, pois se sabe que essas diferenças requerem suportes personalizados, necessitando de profissionais que saibam trabalhar nos diversos sistemas, o que encarece ainda mais o suporte.

Se a certificação LPI não é focada em uma distribuição Linux específica, quer dizer então que não são cobrados assuntos que podem variar de distribuição para distribuição como gerenciadores de pacote, localizações de arquivos em diretórios diferentes e alguns comandos específicos?

Pode sim, mas isso é a exceção, pois o foco principal são os conceitos e o entendimento das tecnologias envolvidas no Linux. No entanto fique atento, pois detalhes sobre gerenciadores de pacotes e arquivos de configurações de distribuições como Red Hat e Debian podem cair. Isso acontece pelo fato de essas distribuições serem pioneiras (matrizes que deram origem a outras) e bem estabelecidas no mercado. Entretanto, a maior parte do que é cobrado na certificação é realmente neutro, valendo para as diversas distribuições.

Para verificar os detalhes de cada distribuição que podem cair nas provas basta consultar o programa de certificação no site oficial da LPI (<http://www.lpi.org>).

### Resumo DevMan

#### *De que se trata o artigo:*

Este artigo explica como é e como funciona a certificação LPI com o intuito de motivar os profissionais de Linux que não sabem como são as provas e querem dar seu pontapé inicial no mundo das certificações Linux.

#### *Em que situação o tema é útil:*

O conteúdo apresentado neste artigo fornecerá subsídios para os profissionais de Linux que estão planejando obter a certificação Linux LPI, que é independente de fornecedores e reconhecida mundialmente.

#### *Prepare-se para as certificações LPI:*

Este artigo procura esclarecer o leitor quanto às dúvidas em torno da certificação LPI, abordando desde seu histórico de evolução no Brasil, como são compostas e divididas as provas por nível, o local em que se pode realizá-las e também dicas para quem quer começar a estudar para as provas.

Existem outras certificações Linux no mercado que são focadas em distribuições específicas, como, por exemplo, as certificações da Red Hat (Red Hat Certified Engineer - RHCE, Red Hat Certified Architect - RHCA e Red Hat Certified System Administrator - RHCSA) e as certificações da Novell (CLA – Certified Linux Administrator). A CompTIA, principal fornecedor de certificações em TI independentes de fornecedor nos Estados Unidos, também tem um programa de certificação voltado para o Linux, chamado CompTIA Linux+.

A Novell e a CompTIA atualmente são parceiras da LPI, aumentando ainda mais o valor da certificação. Atualmente a certificação LPI é tão importante que quem já tirou o primeiro nível já leva junto a certificação Novell Certified Linux Administrator (CLA) e também será considerado certificado CompTIA Linux+ powered by LPI.

### Um pouco de história sobre a certificação LPI no Brasil

A aplicação das provas da certificação LPI no Brasil ocorreu graças ao esforço de pessoas influentes na comunidade Linux do

Brasil. Os diretores da empresa 4Linux, especializada em consultoria e treinamento em softwares livres, Marcelo Marques e Rodolfo Gobbi, deram início à ONG LPI Brasil em 2003.

A primeira prova LPI foi aplicada no Brasil em 2002 com a vinda do John “Maddog” Hall [2], bastante conhecido da comunidade de software livre, à sede da 4Linux em São Paulo. Este foi um passo significativo para que o Brasil tivesse uma grande quantidade de profissionais certificados.

Hoje, a ONG LPI Brasil não existe mais, pois ocorreram mudanças organizacionais na LPI. A 4Linux, que foi pioneira na aplicação das provas no Brasil, se tornou uma afiliada LPI, e novos aplicadores (proctors) de provas foram capacitados e autorizados. A LPI no Brasil, este ano, comemora 10 anos de vida.

## Provas

Cada prova da LPI tem 60 questões, e os assuntos são organizados por tópicos. Neste formato, cada tópico possui um peso, o que especifica a quantidade de questões a serem cobradas sobre um determinado assunto. Por exemplo, o tópico “Instalar um gerenciador de inicialização (boot)” da prova 101 possui peso 2, o que significa que terão duas questões sobre ele na prova. O respectivo peso de cada assunto pode ser encontrado no site oficial da LPI [4], dentro da seção de objetivos detalhados.

A pontuação varia de 200 a 800 pontos, não sendo possível determinar a quantidade mínima de questões que o candidato deve acertar para obter êxito. O mínimo para conseguir aprovação é 500 pontos. Além disso, o candidato tem 90 minutos para responder a prova, formada por questões de múltipla escolha e dissertativas, que cobram comandos e geralmente suas opções, arquivos de configuração ou conceitos em geral.

A seguir apresentamos alguns exemplos de questões:

1. Qual comando pode ser usado para colocar em funcionamento as mudanças feitas no arquivo `/etc/inittab` sem ter que reiniciar o sistema?

- a) `init q`
- b) `init now`
- c) `runlevel`
- d) `inittab q`

2. Você precisa alterar as permissões do diretório `/home/leo` e todo seu conteúdo para as seguintes permissões: `drwxr--r--`. Quais os comandos abaixo com seus respectivos argumentos fazem isso?

- a) `chmod 0744 /home/leo/ -R`
- b) `chmod 744 /home/leo/*`
- c) `chmod -R u=rwx,g=r,o=r /home/leo/`
- d) `chmod -R u=rwx,g=rw,o+r /home/leo/`

3. O comando \_\_\_\_\_ com 3 letras no Bash mostrará todas as variáveis de ambiente.

4. Você deseja achar instâncias da palavra “Maria” no começo das linhas com o editor de textos VI. Qual a combinação de teclas para

fazer isso? Inclua o caractere que indica a busca no VI e considere que você já está no modo de comando. \_\_\_\_\_

Para quem já deseja ir se preparando, eis o gabarito: 1) Letra A; 2) Letras A e C; 3) Comando `env`; 4) `/^Maria`.

## Níveis da LPI

A LPI possui três níveis profissionais atualmente, a saber: LPIC-1, LPIC-2 e LPIC-3. É importante destacar “níveis profissionais”, pois a LPI lançará em Junho de 2012 uma certificação chamada “Linux Essentials”, voltada para o meio acadêmico e a usuários iniciantes em Linux. Para mais detalhes, consulte [16].

O intuito da LPI ao criar esta nova certificação é melhorar o currículo de pessoas que querem entrar no mercado de Linux, mas nunca tiveram uma experiência profissional com o mesmo.

Vale ressaltar que a certificação “Linux Essentials” não é considerada um pré-requisito para as certificações profissionais (LPIC-1, LPIC-2 e LPIC-3).

A LPIC-1 e a LPIC-2 são compostas por duas provas cada. Já a LPIC-3 possui uma quantidade de provas maior. Ao todo são seis, sendo que duas delas ainda estão em fase de desenvolvimento. É neste último nível que o administrador de sistemas sênior escolherá uma área para se especializar, que pode ser segurança, ambientes mistos, virtualização e alta disponibilidade, correio eletrônico ou servidor web e intranet.

## LPIC-1 (Linux Professional Institute Certification – 1 / Junior Level Linux Certification)

A LPIC-1 é voltada para administradores Linux nível júnior e é composta pelas provas 101 e 102. Esta é a porta de entrada das certificações LPI e o pré-requisito para a certificação LPIC-2.

Pré-requisito: Nenhum.

### A prova 101

A prova 101 é focada nos seguintes assuntos: arquitetura do sistema; instalação do Linux; gerenciamento de pacotes; comandos GNU/Linux; dispositivos; e Filesystem Hierarchy Standard (ou Padrão Hierárquico do Sistema de Arquivos).

É reconhecida pela maioria dos candidatos a LPIC-1 como a prova mais difícil deste nível, por requerer detalhes de comandos e até assuntos pouco vivenciados no dia a dia do administrador de sistema.

Nesta prova, o candidato deve estar bem familiarizado com os comandos GNU/Linux e deve ficar atento quando uma questão cobra somente o comando ou o comando e seus parâmetros. Ela requer que o candidato tenha bastante intimidade com a linha de comando no Linux e saiba os principais parâmetros dos comandos mais importantes. Muitos candidatos consideram a prova 101 mais difícil do que a 102 por este motivo.

O candidato não deve focar apenas nos comandos mais usuais como **cat**, **grep**, **chmod**, **chown** e **cp**. Comandos para instalação e remoção de pacotes dos gerenciadores **aptitude**, **dpkg**, **yum** e **RPM** são essenciais e também devem ser dominados para conseguir uma boa pontuação na prova.

Também se deve prestar atenção aos comandos de gerenciamento dos sistemas de arquivos, que por não serem tão usuais podem fazer com que o candidato não dê a devida importância. São exemplos deste tipo de comando: **dumpe2fs**, que mostra detalhes sobre uma partição do disco (tipo da partição, número de inodes, blocos livres, tamanho do bloco, intervalo entre checagens automáticas); **e2fsck**, que é usado na checagem de integridade de sistemas de arquivos EXT2; **debugfs**, que é usado para examinar e mudar o estado do sistema de arquivos EXT2; e **tune2fs**, que pode ser usado para modificar a quantidade de vezes que um sistema de arquivos EXT será checado.

Muitos candidatos estão acostumados a lidar apenas com sistemas de arquivos EXT3. Mas estude também sobre o sistema de arquivos XFS e seus utilitários, pois também são cobrados na prova.

Outro ponto importante é dominar no mínimo o básico do editor de textos VI, que é o “calcanhar de Aquiles” de muitos administradores de sistema iniciantes. Esse editor causa estranheza a muitos profissionais de Linux, por ser bastante diferente na hora de editar um arquivo quando o comparamos com outros editores de textos, como o **nano**, o **pico** e o **mcedit**.

Apesar do peso do tópico sobre o editor de textos VI ser considerado baixo (peso 3), não é um assunto tão complicado para que o candidato não acerte as questões.

O editor de textos VI existe em praticamente todas as distribuições Linux. Portanto, o candidato deve dominar os comandos para movimentar o cursor pelo texto, copiar, colar, apagar, salvar, sair do editor e procurar ocorrências com expressões regulares básicas.

Operações avançadas como trabalhar com macros no VI não costumam cair na prova. Uma boa dica para estudar o VI de maneira divertida é o site <http://aurelio.net/vim/>, que possui materiais sobre o editor de textos VIM que é um clone 100% compatível com o VI, ou seja, o que se faz no VI pode-se realizar no VIM.

Outra dica importante é dominar os gerenciadores de pacotes (**dpkg**, **aptitude**, **apt-get**, **RPM** e **yum**). Se você está acostumado apenas com os gerenciadores de pacotes do Debian ou Red Hat, crie uma máquina virtual com o VirtualBox e uma distribuição derivada de uma das que não se tem costume. Para treinar os comandos do RPM e do yum, por exemplo, você pode optar pela distribuição CentOS ou Fedora. E para quem não quer testar o Debian puro, tem a opção da sua distribuição descendente mais famosa atualmente, o Ubuntu.

Um assunto que os candidatos novatos no trabalho com o Linux costumam sentir dificuldades é com as expressões regulares. A princípio parece ser um assunto realmente complicado, mas na prova 101 este assunto tem peso baixo e é algo que um bom administrador de sistemas, mesmo que júnior, deve dominar. Uma dica interessante de livro sobre esse tema é “Expressões regulares – Uma abordagem divertida”, do autor Aurélio Marinho Jargas, editora Novatec, que aborda um assunto muito importante inclusive para programadores. Assim o candidato já se prepara também para o tópico de Shell Script, que é cobrado na prova 102.

## A prova 102

A prova 102 é focada nos seguintes assuntos: Shell, Shell Script, gerenciamento de dados com SQL, interfaces de usuário, tarefas administrativas, serviços essenciais ao sistema, fundamentos de rede e segurança básicos.

A LPI trouxe um assunto novo para a prova 102 em 2009. Este contempla os comandos SQL. A princípio isso causou estranheza para a maior parte dos usuários que já tinham prestado a prova anteriormente. Mas fique tranquilo, o que é cobrado sobre a linguagem SQL são comandos básicos como inserção, remoção, pesquisa de dados e seleção de tabelas. O peso deste requisito é baixo. Assim, o máximo de questões que podem cair sobre isso são duas, já que o seu peso é 2.

A linguagem SQL começou a ser cobrada porque atualmente é necessidade básica de um administrador de sistemas saber lidar ao menos com o básico dela. O administrador precisa saber instalar aplicações web como o Moodle, utilizada para ensino a distância, fóruns, etc., e essas aplicações geralmente requerem um banco de dados como o MySQL ou o PostgreSQL. Apesar disso, a LPI não foca em nenhum SGBD específico, e sim na linguagem SQL.

A seguir apresentamos uma questão de exemplo sobre SQL:

1. Tenho uma base de dados (tabela) chamada clientes e gostaria de selecionar somente os clientes que são da cidade de São Paulo. Selecione a opção correta:
  - a) `select * from clientes where cidade like 'São Paulo';`
  - b) `select 'São Paulo' from clientes;`
  - c) `select * from 'São Paulo' where clientes;`
  - d) `select all from clientes where cidade like 'São Paulo';`

Gabarito: a.

O candidato que faz a prova 102 deve saber automatizar tarefas com Shell Script e usar as variáveis de ambiente e o agendamento de tarefas, já que são assuntos intimamente ligados. Se o administrador da rede monta um script para fazer backup periodicamente, ele deve saber os comandos envolvidos para isto e para agendá-lo. Uma observação importante é que a LPI não vai pedir para que se desenvolva um script na prova, mas sim saber os comandos envolvidos para criar um laço de repetição, uma estrutura condicional, etc.

Um assunto que pode pegar o candidato de surpresa com questões que cobrem coisas pouco utilizadas é o requisito sobre o servidor X, isto é, trabalhar com a configuração da interface gráfica. Comandos como *xhost*, *xwininfo* e *X* são cobrados. Não seja pego de surpresa!

Saber administrar usuários é essencial. É assunto que o candidato deve dominar 100%, pois faz parte da rotina do administrador, seja ele júnior, pleno ou sênior. Nesse caso as questões não costumam pegar quase ninguém de surpresa, mas fique atento aos arquivos de configuração e comandos envolvidos neste requisito tão importante!

Um assunto que não costuma ser do cotidiano do administrador de sistemas infelizmente é acessibilidade. Seu peso é mínimo,

caindo apenas uma questão sobre o assunto. Portanto, é importante conhecer os softwares de acessibilidade cobrados na LPI, como por exemplo, o ORCA, que é um software livre para deficientes visuais.

Além de saber trabalhar com logs no sistema, o candidato também deve dominar o básico sobre configurações e serviços de rede, como: o NTP para ajuste de data e hora em rede, o servidor de mensagens, criar aliases de e-mail e configurar o redirecionamento de mensagens. Apesar disso, não é cobrada a configuração de um MTA (*Mail Transfer Agent*). O SSH é outro assunto cobrado, sendo preciso saber trabalhar com os dados de maneira segura com o software livre chamado *gnupg*, que passou a ser cobrado a partir de uma mudança feita na prova da LPI em 2009. Mais detalhes sobre os objetivos do exame 102 podem ser obtidos em [9].

É importante destacar que a LPIC-1 sofrerá uma atualização no seu conteúdo a partir de 2 de Julho de 2012. Ela passará a cobrar conhecimentos sobre o gerenciador de inicialização GRUB2, o sistema de arquivos EXT4, conhecimentos básicos do *systemd* e *upstart*, conhecimentos básicos do protocolo IPv6 e LVM (Logical Volume Manager – Gerenciador de Volumes Lógicos) e não cairão mais questões sobre o gerenciador de inicialização LILO.

Observação! Para ser considerado certificado LPIC-1, é necessário obter aprovação nas provas 101 e 102. Caso seja reprovado em uma delas, o candidato deverá refazer apenas aquela em que não obteve sucesso.

Se o candidato realizar uma prova e não for aprovado, é possível fazer uma nova tentativa após sete dias. Caso seja reprovado novamente, deverá aguardar noventa dias para repetir pela terceira vez a mesma prova. Caso o candidato faça a prova antes desse prazo, o exame será rejeitado e o mesmo será suspenso de prestar a prova pelos próximos dois anos. Além disso, a segunda tentativa não é gratuita e a prova deverá ser paga novamente.

## LPIC-2 (Linux Professional Institute Certification – 2 / Advanced Level Linux Certification)

A LPIC-2 é voltada para administradores Linux nível pleno e é composta pelas provas 201 e 202. O foco principal das provas deste nível é certificar que os candidatos são capazes de trabalhar com servidores de pequeno e médio porte com ambientes mistos (Linux e Windows), saber configurar um firewall, um Proxy, um servidor Web, um servidor FTP, configurar e manter um servidor DNS com segurança.

Pré-requisito: LPIC-1.

### A prova 201

A prova 201 cobra assuntos como compilação de Kernel e gerenciamento de módulos, sistema de inicialização, sistemas de arquivos e dispositivos, administração avançada de dispositivos de armazenamento como RAID nível 0, 1 e 5 (via software), LVM e tuning de sistemas de arquivos, particionamento, configuração de rede a cabo e sem fio, roteamento, OpenVPN, configuração de um servidor DNS usando o BIND, comandos utilitários de rede e resolução de problemas envolvendo a rede.

Compilar o Kernel e gerenciar módulos no sistema deve ser tarefa habitual para quem deseja obter aprovação nesta prova, já que existirão várias questões envolvendo estes assuntos.

Conhecer como é a inicialização do sistema, além de saber gerenciar quais serviços serão ativados no momento da inicialização, é um exemplo de assunto que não se pode perder pontos, porque se lida com isso bastante no dia a dia do administrador pleno. Detalhes sobre os sistemas Red Hat e Debian são cobrados aqui, pois possuem comandos diferentes para realizar trabalhos em comum.

Lidar com sistemas de arquivos é algo que é bastante cobrado também. Deste modo, é importante saber criar, ativar e desativar uma memória swap, criar e fazer checagem de integridade em sistemas de arquivos EXT3, XFS e para o formato ISO. Ademais, o candidato deve ter domínio sobre como ajustar o acesso aos dispositivos de armazenamento com os comandos *hdparm*, *sdparm*, *tune2fs* e *sysctl*.

Na parte de redes, não basta conhecer como configurar uma rede com cabos ou wireless. O candidato deve saber como lidar com problemas comuns que podem ocorrer em uma rede e como sair deles. Questões sobre “troubleshooting” despenham na prova 201!

A seguir apresentamos uma questão de exemplo:

1. Uma entrada correta foi adicionada no arquivo */etc/hosts.allow* para permitir que certos clientes se conectem ao servidor, mas isso não surtiu efeito. O que poderia estar causando isto?

- a) *tcpd* precisa receber um sinal HUP;
- b) O serviço precisa ser reiniciado;
- c) A máquina precisa ser reiniciada;
- d) Existe um conflito na entrada em */etc/hosts.deny*;
- e) O serviço não suporta *tcpwrappers*.

Gabarito: e.

Para saber os objetivos desta prova detalhadamente, acesse [10].

### A prova 202

A prova 202 cobra assuntos como servidor Web Apache 2, Proxy SQUID, compartilhamento de arquivos com SAMBA e NFS, configuração de servidor DHCP, autenticação com o PAM, LDAP como cliente, servidor de e-mail (Postfix/Sendmail), gerenciamento de regras de firewall com *iptables*, servidor FTP com *VSFTPD* e *TCP Wrapper*.

Essa prova costuma ser mais difícil que a 201, pois envolve raciocínio para descobrir soluções ou a causa de problemas que atrapalham o correto funcionamento de um serviço ou da rede.

Os assuntos mais complexos da prova 202 costumam ser a configuração do servidor DNS e do servidor de e-mail. Esta prova é recheada de questões para resolução de problemas, o que exige experiência na configuração e manutenção destes serviços.

É importante ressaltar que a LPIC-2 sofrerá uma atualização no seu conteúdo a partir de 1 de Agosto de 2012. Ela passará a cobrar conhecimentos sobre o sistema de arquivos EXT4, o protocolo IPv6,

conhecimentos básicos em sistemas de arquivos encriptados, além dos comandos *xfsdump* e *xfsrestore*, e detalhes do Kernel versão 3.0.

Para obter mais detalhes sobre os objetivos dessa prova, acesse [11].

Observação! Para ser considerado certificado LPIC-2 é necessário obter aprovação nas provas 201 e 202. Em caso de reprovação em uma delas, o candidato deverá refazer apenas aquela em que foi reprovado.

## LPIC-3 (Linux Professional Institute Certification – 3 / Senior Level Linux Certification)

A LPIC-3 é voltada para administradores Linux nível sênior e composta pelas provas 301 a 306.

Ao contrário das certificações de nível 1 e 2, para ser considerado LPIC-3 basta obter aprovação na prova 301.

Vamos a um detalhamento sobre as provas 301 a 306.

### A prova 301

A prova 301 qualifica o profissional como Core, e as demais do nível LPIC-3 cobram assuntos específicos, como ambientes mistos, segurança e virtualização.

Esta prova testa conhecimentos sobre as tecnologias do OpenLDAP como servidor e não só como cliente, como é cobrado na prova da LPIC-2. Além disso, pode ser cobrado como é a integração do OpenLDAP ao PAM, como integrá-lo com serviços de acesso remoto SSH, de transferência de arquivos por FTP, servidor WEB, FreeRADIUS, impressão, Samba, Active Directory, Postfix, Sendmail etc. Também busca explorar a capacidade de planejamento de recursos de hardware e banda de rede para um servidor, além de saber mensurar o uso de memória, entrada e saída de disco e rede, roteamento e regras de firewall, e as famosas questões sobre resolução de problemas conhecidos como “troubleshooting”.

Pré-requisito: LPIC-2.

Para mais detalhes sobre o objetivo desta prova, consulte [12].

### A prova 302

A prova 302 é voltada para ambientes mistos. Seu principal assunto é o servidor SAMBA com compartilhamento de arquivos, servidor de impressão e controlador primário de domínio. Também é necessário saber como trabalhar com CIFS, NetBIOS, WINS e Active Directory, e conhecer detalhes sobre segurança e performance.

Para mais detalhes sobre o objetivo desta prova, consulte [13].

### A prova 303

A prova 303 é voltada para segurança e envolve assuntos como criptografia, controle de acesso, SELinux, servidores de rede, SNORT, escaneamento de dados em rede, OpenVPN e monitoração de rede.

Para mais detalhes sobre o objetivo da prova, consulte [14].

### A prova 304

A prova 304 é voltada para virtualização (KVM e XEN) e alta disponibilidade (balanceamento de carga, gerenciamento de cluster e cluster storage).

Para mais detalhes sobre o objetivo da prova, consulte [15].

### A prova 305

A prova 305 ainda não possui conteúdo detalhado no site oficial da LPI, mas será sobre servidor de e-mail e se encontra em desenvolvimento. Lembre-se que o assunto servidor de e-mail também é cobrado na prova 202 da LPIC-2 de forma básica. A diferença é que a prova 305 se aprofunda no assunto, já que é um nível mais avançado sobre correio eletrônico.

### A prova 306

A prova 306 também não possui conteúdo detalhado no site oficial da LPI, mas será sobre servidor Web e Intranet e se encontra em desenvolvimento.

Você é considerado especialista em um determinado assunto quando possui a prova 301 mais uma prova especialista, que varia da 302 a 306, de acordo com a **Tabela 1**.

| Especialidade                               | Combinação de provas |
|---------------------------------------------|----------------------|
| LPIC-3 Ambientes mistos                     | 301 Core + 302       |
| LPIC-3 Segurança                            | 301 Core + 303       |
| LPIC-3 Virtualização e Alta Disponibilidade | 301 Core + 304       |
| LPIC-3 Correio eletrônico                   | 301 Core + 305       |
| LPIC-3 Web e Intranet                       | 301 Core + 306       |

**Tabela 1.** Especialidades da LPIC-3

Observação! O termo Core, usado na prova 301, é apenas para deixar claro que ao obter êxito nela, o candidato já será considerado LPIC-3. Esta é a prova principal da LPIC-3. O termo especialista é conquistado tendo sido aprovado nela e em mais outra prova (que pode variar da 302 a 306). Realizando apenas a prova 301, o candidato será considerado sênior. Para ser especialista em segurança, por exemplo, o candidato precisa obter êxito nas provas 301 e 303.

## Formato das provas

As provas da certificação LPI podem ser aplicadas em papel ou no meio eletrônico.

As provas eletrônicas são realizadas em centros autorizados VUE ou Prometric, e as PBTs (Paper Based Tests – Provas em Papel) em eventos de software livre e em locais em que não existe um centro autorizado a aplicá-la de forma eletrônica. As provas em papel feitas no Brasil são enviadas para os Estados Unidos, onde são corrigidas. Após o prazo de 15 a 30 dias o candidato recebe o resultado através do e-mail. Se aprovado, mais dias serão necessários para que o mesmo receba o certificado via Correios.

Para saber onde fazer uma prova LPI em PBT basta acompanhar o site <http://www.lpi-brasil.org> e procurar um LATP (LPI Approved Training Partner – Parceiro Aprovado de Treinamento) com seu calendário de provas. O site <http://www.lpi-brasil.com.br> traz o calendário de provas aplicadas pelo LATP 4Linux e o site <http://www.lpi-brasil.org> destaca o calendário de provas aplicadas por outros LATPs.

O preço de cada prova em papel, cobrada em reais, gira em torno de 290,00 reais, exceto a 301, que custa 470,00 reais. Mas este valor pode variar, sendo necessário consultá-lo no site da LPI Brasil na época em que for realizá-la.

As provas em papel dos níveis 1 (LPIC – 1) e 2 (LPC2) são em português e as provas do nível 3 (LPIC – 3) são em inglês.

As CBTs (Computer Based Tests – Provas eletrônicas) podem ser agendadas no horário e dia desejados através do site da Prometric (<http://www.prometric.com>) ou VUE (<http://www.pearsonvue.com>). Além disso, será necessário cartão de crédito internacional para fazer o pagamento das provas, pois este é realizado em dólares.

O preço de cada prova eletrônica custa aproximadamente 173 dólares, mas pode variar de acordo com a região. Já a prova 301 geralmente custa mais caro, sendo cobrado aproximadamente 260 dólares. As demais provas de nível 3 (302, 303, 304, 305 e 306) costumam ter o mesmo valor das de nível 1 e 2. Portanto, é importante consultar o site da Prometric ou da VUE no momento do cadastro para saber o valor atual. O resultado da prova eletrônica é mostrado assim que o candidato finaliza o exame e por isso tem se tornado preferência entre os candidatos.

Observação! Para agendar a prova eletrônica ou se inscrever para fazer a prova em papel, o candidato deverá ter o LPI ID criado através do seguinte endereço: <https://cs.lpi.org/cafi/Xamman/register>. A criação do LPI ID é gratuita.

## Conclusão

A grande dica para quem quer conquistar as certificações LPI é consultar os objetivos detalhados no site oficial da LPI (<http://www.lpi.org>) e buscar material sobre os assuntos, se possível matriculando-se em cursos preparatórios.

Existem excelentes opções de cursos no nosso país. No entanto dê preferência aos cursos que fornecem simulados e dicas para a prova. É comum encontrar pessoas com excelente conhecimento técnico que foram reprovadas nas provas por não conhecerem o estilo das questões e não terem o hábito de responder as provas das certificações como exercício.

Na opinião do autor é melhor investir em simulados e treinamentos do que em livros em específico, pois são formas mais eficientes e eficazes de treinar para obter a certificação.

Outra dica importante é que o candidato deve ficar atento ao prazo de validade da certificação quando obtiver êxito nos exames. Atualmente a validade é de cinco anos. Antes de finalizar este prazo, ele deve refazer as provas do nível mais alto que possui ou obter a certificação de um nível maior. Dessa forma automaticamente estará renovando a certificação do nível anterior.

Para finalizar, dê um passo de cada vez, se planeje e você ganhará mais confiança em cada prova que fizer. Boa sorte e bons estudos!



### Leonardo Afonso Amorim

[leonardoafonso@gmail.com](mailto:leonardoafonso@gmail.com)

Tem 26 anos, é bacharel em Engenharia de Computação pela PUC Goiás e atua como Analista de Sistemas Pleno pela empresa 4Linux com software livre desde 2006 na área de consultoria e treinamentos. Ministrou mais de 70 cursos EAD em assuntos com Linux, serviços de rede, segurança e atualmente também com Java. É certificado LPIC-1 e LPIC-2.



#### [1] Padrão LSB

<http://www.linuxfoundation.org/collaborate/workgroups/lsb>

#### [2] Relato de John “Maddog” Hall sobre a certificação LPI e a vinda da mesma para o Brasil

<http://blog.lpi.org/index.php/2010/10/lpi-and-my-first-international-proctoring-job/>

#### [3] Site da LPI Brasil mantido pela 4Linux

<http://www.lpi-brasil.com.br>

#### [4] Site oficial da LPI no Brasil

<http://www.lpi-brasil.org/>

#### [5] Site Oficial da LPI

<http://www.lpi.org>

#### [6] Certificação CompTIA Linux

<http://certification.comptia.org/getCertified/certifications/linux.aspx>

#### [7] Programa LPIC-3

[http://www.lpi-brasil.com.br/sites/default/files/lpic-3\\_program.pdf](http://www.lpi-brasil.com.br/sites/default/files/lpic-3_program.pdf)

#### [8] Detalhamento do Exame 101

<http://www.lpi.org/linux-certifications/programs/lpic-1/exam-101>

#### [9] Detalhamento do Exame 102

<http://www.lpi.org/linux-certifications/programs/lpic-1/exam-102/>

#### [10] Detalhamento do Exame 201

<http://www.lpi.org/linux-certifications/programs/lpic-2/exam-201>

#### [11] Detalhamento do Exame 202

<http://www.lpi.org/linux-certifications/programs/lpic-2/exam-202>

#### [12] Detalhamento do Exame 301

<http://www.lpi.org/linux-certifications/programs/lpic-3/exam-301>

#### [13] Detalhamento do Exame 302

<http://www.lpi.org/linux-certifications/programs/lpic-3/exam-302>

#### [14] Detalhamento do Exame 303

<http://www.lpi.org/linux-certifications/programs/lpic-3/exam-303>

#### [15] Detalhamento do Exame 304

<http://www.lpi.org/linux-certifications/programs/lpic-3/exam-304>

#### [16] Detalhamento da prova Linux Essentials

<http://www.lpi.org/linux-certifications/introductory-programs/linux-essentials>



# Soluções com o Windows PowerShell

## Uma poderosa ferramenta para automatização

Por muitos anos certamente administradores gerenciaram seus servidores e produtos Microsoft utilizando GUI (Graphical User Interface – Interface Gráfica de Usuário), *cmd.exe*, VBScript e talvez outras tecnologias complementares.

Em função de maior flexibilidade e extensão, a Microsoft vem tentando fazer com que o Microsoft Windows PowerShell seja a ferramenta de gerenciamento de melhor escolha. Para isso, os novos produtos de servidores requerem Windows PowerShell para administração em função da sua facilidade no gerenciamento e na manutenção.

O Microsoft Windows PowerShell é um novo prompt de comando do Windows que comparado ao antigo *cmd.exe* é muito mais poderoso. É voltado ao uso de scripts para manutenção de sistemas por parte de administradores, oferecendo maior controle do sistema e agilidade na automatização de tarefas como backup, transferência de arquivos, parametrização de aplicações como Microsoft Exchange, entre outras.

Criado em uma estrutura do Framework .NET, o Windows PowerShell ajuda os profissionais de Tecnologia da Informação (TI) e usuários a controlar e automatizar a administração do Sistema Operacional Windows e os aplicativos que são executados no mesmo. É sobre este tema que o conteúdo deste artigo é baseado.

### Windows PowerShell Versão 2.0

Até o momento da escrita deste artigo, o Windows PowerShell encontrava-se na versão 2.0, que oferece diversas melhorias, a saber:

- 1. Comunicação remota:** Permite executar comandos em um ou mais computadores remotamente, a partir de um computador que esteja executando o Windows PowerShell;
- 2. ISE (Integrated Scripting Environment – Ambiente de Script Integrado):** Permite executar comandos interativos, editar e depurar scripts em um ambiente gráfico, incluindo ao ambiente sintaxe realçada por cor, depuração gráfica, suporte a Unicode e ajuda contextual do código;

### Resumo DevMan

#### De que se trata o artigo:

Este artigo retrata uma abordagem simplificada de primeiro contato com o novo shell e linguagem de script da Microsoft, com a finalidade de apresentar e exemplificar sua utilização no dia a dia do administrador de redes.

#### Em que situação o tema é útil:

As novas versões do Windows Server 2008 e produtos Microsoft estão vindo com um gerenciamento via Windows PowerShell. Nestas condições, faz-se necessário o entendimento e aprendizado do mesmo para a administração de uma rede com ambiente Microsoft.

#### Soluções com o Windows PowerShell:

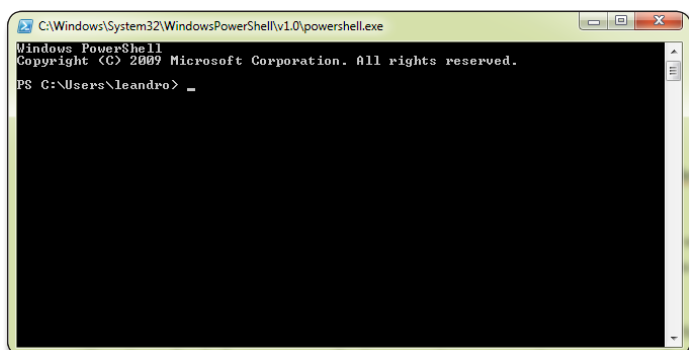
O Microsoft Windows PowerShell é um novo prompt de comando do Windows, mais poderoso e flexível para administração de servidores, aplicações e scripts. É sobre ele que abordaremos neste artigo.

- 3. Módulos:** Permite ao desenvolvedor e ao administrador dividir e organizar o código escrito em Windows PowerShell em unidades reutilizáveis e independentes;
- 4. Trabalhos em background:** Permite a execução de comandos PowerShell em segundo plano, sem interação com o console;
- 5. Novos cmdlets** (pronuncia-se *command-lets*): São comandos integrados que estão incluídos no Windows PowerShell. Assemelham-se aos comandos internos em outros shells, como por exemplo, o **dir**, encontrado no *cmd.exe*. Podem ser chamados diretamente da linha de comando do Shell e executados no contexto desse interpretador, e não como um processo separado.

### Primeiro contato

A instalação do Windows PowerShell é feita a partir do Windows XP com Service Pack 3, e, também, nas versões superiores do Windows, tendo como pré-requisito a instalação do Microsoft .NET Framework 2.0 com Service Pack 1 ou superior.

Após a instalação, começaremos então a entender como funciona o Windows PowerShell. Para inicialização do mesmo, basta executá-lo a partir do menu *Iniciar*. Para isto, clique em *Iniciar > Executar*. Observe que aparecerá uma caixa solicitando a digitação do comando a ser executado, que é *Powershell.exe*. Feito isso, será aberto um *shell* para execução dos comandos e criação de scripts, conforme ilustrado na **Figura 1**.



**Figura 1.** Prompt de comando do Windows PowerShell quando inicializado

## Comandos internos do PowerShell

Como foi dito, o Windows PowerShell é repleto de comandos chamados *cmdlets*. Existem mais de 250 comandos do Windows PowerShell para aprender. No entanto, veremos que pela estrutura com que o Windows PowerShell foi desenhado, facilmente encontraremos os comandos que precisamos com uma ótima documentação para ajuda.

Ao contrário da maioria das outras interfaces de linha de comando, o Windows PowerShell foi padronizado utilizando verbos auxiliares no início dos comandos. Uma convenção de nomenclatura para saber como um *cmdlet* irá agir. Esta padronização é importante pois simplifica a curva de aprendizado, e a documentação do comando fornece uma descrição do que o *cmdlet* faz e formas de utilizá-lo.

As palavras reservadas dos *cmdlets* do Windows Power Shell são baseadas no inglês, por isto tem-se no início dos comandos verbos como **Get**, **Set**, **Add**, dentre outros. Por exemplo, se desejar ver uma lista de *cmdlets* disponíveis no Windows PowerShell, basta digitar o *cmdlet*: `Get-Command <enter>`

Na saída de **Get-Command**, apresentada na **Figura 2**, é possível notar que existe uma série de outros comandos que começam com **Get**, **Set**, **Clear**, **Add**, **Start**, etc. Isso simplifica bastante a curva de aprendizado em função da padronização dos comandos iniciados sempre com um verbo auxiliar.

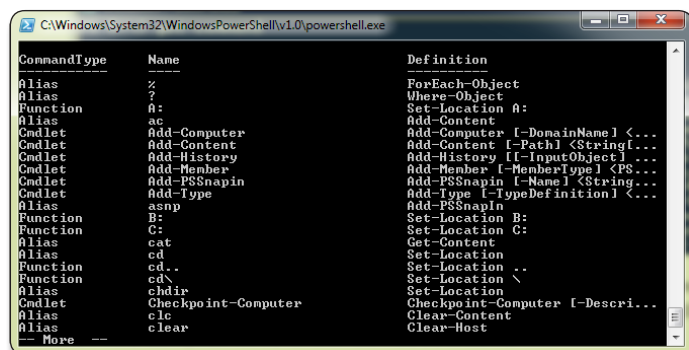
Ainda com a janela aberta, vamos dar sequência ao nosso estudo, familiarizando-se ainda mais com o Windows PowerShell e entendendo melhor esse *cmdlet* chamado **Get-Command**.

Para quem já utilizou outra linguagem de script e está acostumado com comandos, percebe que muitos retornam informações sem a necessidade de parâmetros.

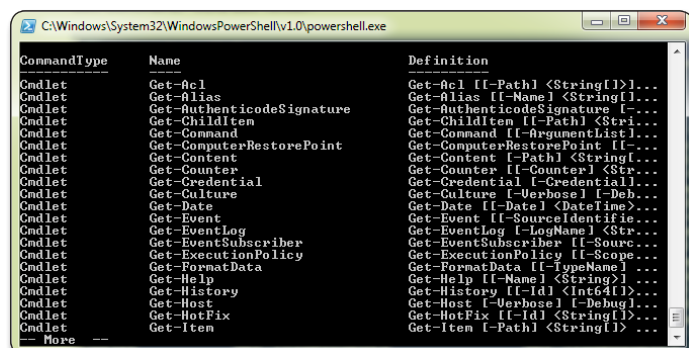
No uso do comando **Get-Command** pode-se incrementá-lo passando, por exemplo, um parâmetro que ajude a listar comandos

com verbos específicos, ou seja, se houver necessidade de resgatar uma lista de comandos iniciados com algum verbo, como **Get**, **Set**, **New** ou outro qualquer, basta informá-lo via parâmetro.

Tomando como exemplo que se queira uma lista de comandos que contenham o verbo **Get**, basta executar o mesmo comando utilizando o parâmetro “-Verb” e o verbo a ser listado. Por exemplo: `Get-Command -Verb Get <enter>`



**Figura 2.** Lista de cmdlets exibidos com o uso do comando `Get-Command`



**Figura 3.** Lista de cmdlets iniciados com o verbo auxiliar `Get`

Como se pode notar na **Figura 3**, com pouco contato e alguns minutos dedicados ao Windows PowerShell, facilmente pode-se identificar uma lista de comandos que anteriormente era difícil de ser encontrada no *cmd.exe*.

Bom, já sabemos como listar comandos e filtrá-los de acordo com nossa necessidade. Deste modo, se em algum momento precisarmos adquirir alguma informação do sistema operacional, como por exemplo, a data, é necessário obter (verbo **Get**) uma informação que neste caso trata-se da data. Então basta procurar na lista de saída do **Get-Command -Verb Get** se tem algo relacionado à data. Nesse caso, veremos na listagem o comando **Get-Date** e facilmente receberemos a informação da data.

## Help

Um ponto fundamental e indispensável para o aprendizado e uso da ferramenta é o sistema de ajuda, e neste contexto novamente o Windows PowerShell foi muito bem arquitetado.

Seguindo a linha de raciocínio e uso dos verbos auxiliares empregado no começo do comando, nota-se que para obter

(Get) ajuda (Help) podemos utilizar novamente o comando **Get-Command -Verb Get**. Neste caso já é possível deduzir algo como **Get-Help**, e, sem muita surpresa, verificamos que o comando existe. Para facilitar, podemos digitar **Get-<TAB>** para completar o comando.

Até agora utilizamos o comando **Get-Command** para listar os *cmdlets* internos do PowerShell que contenham o verbo auxiliar **Get**. Aprendemos também que este e outros comandos podem receber parâmetros. Nesse ponto uma dúvida pode ser levantada com relação a quais parâmetros um comando espera receber. Para isto, utiliza-se o *cmdlet* **Get-Help**.

Deste modo, no prompt de comando do PowerShell, execute: **Get-Help**

Na saída do mesmo é possível identificar outro padrão interessante do Windows PowerShell que também visa reduzir a sua curva de aprendizado. Ao digitar **Get-Help**, uma saída na tela é exibida com uma série de informações sobre como utilizá-lo.

Nestas informações está contida uma pequena descrição do que faz o comando, a sintaxe, exemplos e comentários, mantendo sempre esta ordem e padrão para qualquer comando.

Como exemplo, digite o código abaixo no prompt do PowerShell: **Get-Help Get-Command <enter>**

Com a execução do comando **Get-Help Get-Command**, conforme ilustrado na **Figura 4**, é exibida uma saída com todas as informações relacionadas ao *cmdlet*. Nesse contexto, pode-se ter até links de acesso à web, que oferecem mais detalhes sobre as opções do comando.

Com apenas dois *cmdlets* observamos que é possível aprender sobre diversos outros sem a necessidade de dedicar muito tempo navegando pela Internet em busca de informações. Com esta estrutura o Windows PowerShell proporciona mais objetividade na busca de comandos que resolvam nossos problemas ou melhore nossos resultados.

Desta forma, quando se deseja obter informações mais detalhadas sobre um *cmdlet*, basta usar o **Get-Help** passando como parâmetro o nome do *cmdlet*.

Para finalizar esta etapa, digite no prompt do PowerShell o comando: **Get-Help Get-Command -Examples**. Com ele pode-se obter diversos exemplos de uso do *cmdlet* especificado. Veja a **Figura 5**.

A **Tabela 1** apresenta um breve comparativo entre *cmdlets* e outros comandos internos do *cmd.exe* e do Unix Shell.

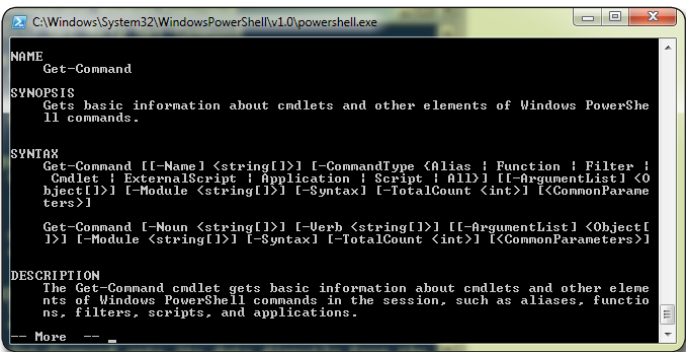


Figura 4. Informações obtidas para ajuda no uso do comando Get-Command

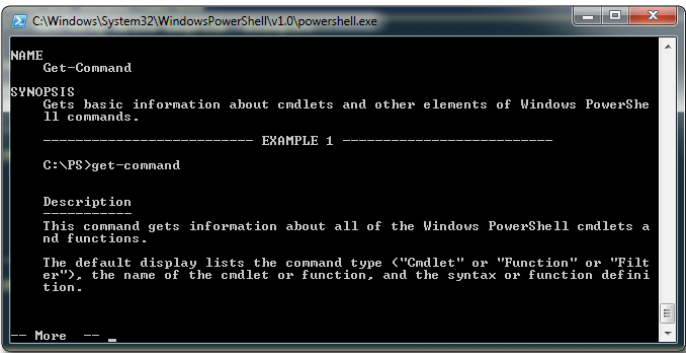


Figura 5. Lista de exemplos para uso do comando Get-Command

| Power Shell (Cmdlet) | Power Shell (Alias)           | Cmd.exe               | Unix Shell | Descrição                                              |
|----------------------|-------------------------------|-----------------------|------------|--------------------------------------------------------|
| Get-ChildItem        | gci, dir, ls                  | dir                   | ls         | Lista todos os arquivos do diretório atual.            |
| Get-Content          | gc, type, cat                 | type                  | cat        | Lista o conteúdo do arquivo.                           |
| Get-Command          | gcm                           | help                  | help       | Lista os comandos disponíveis.                         |
| Get-Help             | help, man                     | help                  | man        | Comando de ajuda.                                      |
| Clear-Host           | cls, clear                    | cls                   | clear      | Limpa a tela.                                          |
| Copy-Item            | cpi, copy, cp                 | copy                  | cp         | Copia um ou mais arquivos.                             |
| Move-Item            | mi, move, mv                  | move                  | mv         | Move um arquivo/diretório para um novo local.          |
| Remove-Item          | ri, del, erase, rmdir, rd, rm | del, erase, rmdir, rd | rm, rmdir  | Deleta um arquivo/diretório.                           |
| Rename-Item          | rni, ren, mv                  | ren, rename           | mv         | Renomeia um arquivo/diretório.                         |
| Get-Location         | gl, pwd                       | cd                    | pwd        | Mostra o diretório atual.                              |
| Set-Location         | sl, cd, chdir                 | cd, chdir             | cd         | Troca o diretório atual.                               |
| Write-Output         | echo, write                   | echo                  | echo       | Mostra o valor de uma variável no prompt de comando.   |
| Get-Process          | gps, os                       | tlist, tasklist       | ps         | Lista todos os atuais processos em execução.           |
| Stop-Process         | spp, kill                     | kill, taskkill        | kill       | Finaliza um processo em execução.                      |
| Select-String        | n/a                           | find, findstr         | grep       | Encontra uma linha utilizando parâmetros comparativos. |

Tabela 1. Cmdlets comparados com outros comandos internos do cmd.exe e do Unix Shell

## Uma linguagem inteligente

Até o momento foi apresentado bem pouco do que o PowerShell pode fazer para os administradores. Vimos apenas dois *cmdlets* que nos mostraram uma lista gigante de outros *cmdlets*. De qualquer maneira, ainda não interagimos com muita inteligência, ou seja, apenas obtemos informações. No dia-a-dia da administração de redes, manipulando dados ou aplicações, tem-se a necessidade de interagir de forma inteligente, aplicando lógica na estrutura.

## Scripts

O Windows PowerShell além de possuir comandos internos é uma linguagem de script do tipo dinâmica, em que não é necessário declarar o tipo de variável que será utilizada. Com isto, permite programar complexas operações usando *cmdlets*.

Por se tratar de uma poderosa linguagem de script, o Windows PowerShell suporta o uso de variáveis, constantes, funções, verificações (**if-then-else**), iterações (**while**, **do**, **for** e **foreach**), tratamentos de erros/exceções e interação com o Framework .NET.

Com base nisso, serão apresentados a seguir alguns aspectos de desenvolvimento para melhor interação e aproveitamento do Windows PowerShell na criação e execução de scripts.

## Variáveis

Antes de iniciar a criação de scripts, é importante destacar que uma variável tem a funcionalidade de armazenar um determinado valor (não necessariamente numérico) durante algum tempo ou até o final da execução do script. Isto vai depender da finalidade com que a mesma foi criada.

Para criar uma variável basta acrescentar no início do nome o caractere “\$”. Desta maneira a mesma armazenará um valor. Considerando que nossa variável se chama nome, devemos criá-la como “\$nome”.

Para criar e atribuir um valor à variável **\$nome**, por exemplo, digite dentro do prompt de comando do Windows PowerShell o seguinte: `$nome = "Leandro"`

Para testarmos o valor armazenado na variável, digite no prompt do Windows PowerShell o nome da variável (“\$nome”). O valor obtido é uma string (conjunto de caracteres) contendo o nome “Leandro”. Na prática poderia conter, por exemplo, o nome de *login* de um usuário. `$nome`

Para atribuir a data do dia de hoje à variável **\$data** utilizando o *cmdlet* **Get-Date**, digite no prompt do Windows PowerShell o seguinte: `$data = Get-Date`

Como já informado, o Windows PowerShell não é uma linguagem fortemente tipada, em que se faz necessário declarar a variável antes de ser utilizada, ou seja, não é necessário informar que a variável **\$nome** é do tipo string. Vê-se claramente que em nenhum momento foi necessário declarar a variável e informar qual será o seu tipo (**int**, **string**, **date**, etc.). Nesse caso, foi executado o *cmdlet* **Get-Date**, atribuindo a saída à variável **\$data**, deixando que o Windows PowerShell defina que tipo será a variável dinamicamente.

Se desejar escrever na tela o valor armazenado na variável, pode-se utilizar o *cmdlet* **Write-Output**, da seguinte forma: `Write-Output $data`

Até então estamos manipulando valores sem nenhum tratamento antes de atribuir o mesmo à variável.

Uma atividade comum na criação de scripts é fazer a verificação de um resultado antes de armazená-lo e/ou manipulá-lo. Para isto, veremos declarações que ajudam a comparar o valor obtido com o valor desejado e então tomar alguma ação, antes de atribuir o valor à variável.

## If, Then e Else

Para tratarmos a informação comparando-a com algum outro valor desejado, utilizam-se as declarações **If**, **Then** e **Else**, que podem ser entendidas como **Se**, **Então** e **Senão**, respectivamente. Com essas declarações é possível comparar um resultado obtido com um resultado esperado e então armazená-lo.

Utilizando a variável criada anteriormente chamada **\$data** (onde se encontram armazenadas informações relacionadas à data de hoje, que foi atribuída através do *cmdlet* **Get-Date**), pode-se comparar o valor contido na variável com um dia da semana para identificar se os dias são iguais. Em caso afirmativo, pode ser exibida na tela uma mensagem dizendo que o dia é igual, caso contrário nada será exibido.

Para isso, digite no prompt de comando do Windows PowerShell o seguinte:

```
$data = Get-Date <enter>
if ($data.DayOfWeek -eq (Get-Date).DayOfWeek) {Write-Output "O dia armazenado na
variável é o mesmo dia da data de hoje"}
```

Verifica-se que se trata de algo bem simples, porém identificamos algumas particularidades na linguagem do Windows PowerShell muito interessantes.

Pelo fato das variáveis serem do tipo dinâmico, **\$data** passa a ser do tipo **Date** ao receber o retorno do *cmdlet* **Get-Date**. Se porventura atribuíssemos uma string, **\$data** seria do tipo string, ou seja, dinamicamente se modificaria para atender o valor recebido. Como se trata de uma variável do tipo **Date**, é possível manipulá-la como tal, chamando assim propriedades como **DayOfWeek**, que informa o dia da semana.

Se por acaso explorarmos a variável **\$nome**, que recebeu uma string, veremos que esta, de forma semelhante, contém métodos e propriedades que nos permitem manipulá-la como uma string.

Para exemplificar, entre com o seguinte comando no prompt do Windows PowerShell: `$nome.ToUpper()`

Tem-se como resultado o valor da variável com todas as letras em maiúsculo.

Por acaso, se você colocar um ponto depois do nome da variável **\$data** e pressionar o TAB, poderá verificar que a mesma possui uma série de propriedades, tais como: **Day**, **Hour**, **Year**, entre outras.

O mesmo vale para o *cmdlet* **Get-Date**, que desta vez foi chamado entre parênteses para garantir que o Windows PowerShell obtenha a data e mostre apenas a propriedade **DayOfWeek**.

De posse das duas informações, podemos comparar se elas são iguais. No Windows PowerShell tem-se uma lista de operadores,

conforme destacado na **Tabela 2**, que fazem essas comparações baseadas em palavras reservadas. No exemplo apresentado, foi comparado se os valores das variáveis são iguais, utilizando para isto a palavra reservada **-eq**.

Para entender o uso destas palavras reservadas, na **Tabela 2** será atribuído o valor 5 a uma variável chamada **\$variavel**, e esta será comparada a um valor com o objetivo de se obter Verdadeiro ou Falso como resultado em várias situações.

Embora esta seja uma particularidade do Windows PowerShell, vimos novamente que a arquitetura da linguagem permite facilmente entender e aprender o que estamos fazendo.

Até este ponto vimos como comparar uma única vez a variável, esperando um resultado falso ou verdadeiro. No entanto é muito comum interagir ao ponto de fazer mais que duas comparações. Para isso devemos avaliar o uso do **else** e do **then**.

Em nosso exemplo, a data armazenada na variável **\$data** é comparada ao dia da semana para identificar se são iguais. Em caso afirmativo, uma mensagem é exibida dizendo: “O dia armazenado na variável é o mesmo dia da data de hoje”. Caso contrário, nada é informado.

Vamos então aprender o que podemos fazer para tratar o caso da condição ser falsa. Para isso utiliza-se a declaração **“else”**. Neste caso dizemos que se (**if**) o dia armazenado na variável **\$data** for igual a hoje, devemos emitir uma mensagem, senão, (**else**) emitir outra mensagem.

Para tanto, dentro do prompt de comando do Windows PowerShell, entre com o código:

```
$data = Get-Date 02/26/1985 <enter>
if ($data.DayOfWeek -eq (Get-Date).DayOfWeek) {Write-Output "O dia armazenado
na variável é o mesmo dia da data de hoje"} else {Write-Output "O dia armazenado na
variável não é o mesmo dia da data de hoje"}
```

É interessante notar que desta vez foi enviado como parâmetro ao **cmdlet Get-Date** uma data, e a partir de então se utilizou esta data para obter informações, como o dia da semana da mesma.

Neste caso a data armazenada na variável foi diferente da data de hoje e por isso o comando emitiu a mensagem: “O dia armazenado na variável não é o mesmo dia da data de hoje”.

Dadas essas condições, é possível fazer diversas comparações a fim de se obter um resultado esperado, ou tratar uma condição não prevista.

Ambiente de Script Integrado (ISE)

Certamente ficou claro que se acrescentarmos mais informações em nossa linha de comando começa a ficar mais complexo o entendimento e consequentemente a manutenção dos scripts. Além disso, caso o prompt do Windows PowerShell seja fechado, as informações serão perdidas, pois não são persistidas ou salvas em nenhum arquivo. Para evitar este problema conheceremos o Ambiente de Script Integrado (ISE), que permite a criação de scripts de forma organizada e com ferramentas de teste e manipulação.

O ISE do Windows PowerShell permite criar, executar e depurar comandos e scripts através de um ambiente integrado completo. Veja a **Figura 6**.

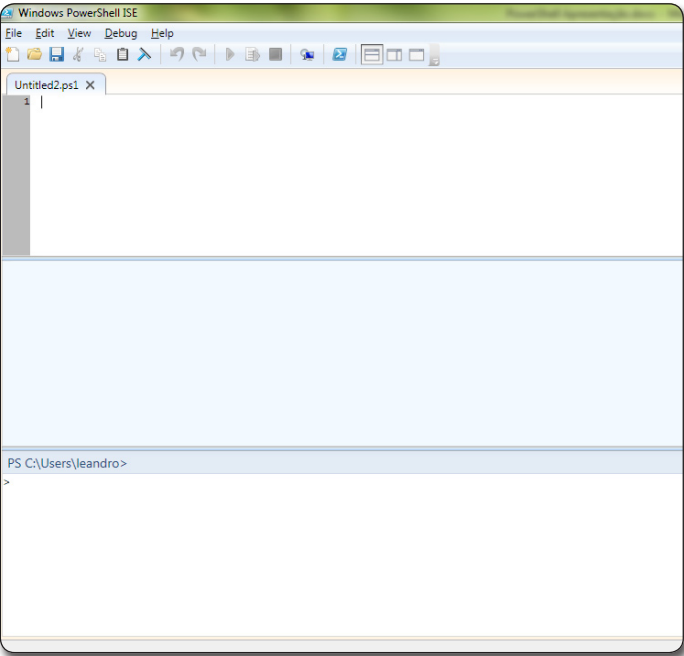


Figura 6. Ambiente de Script Integrado quando inicializado

| Parâmetro | Ação                                      | Exemplo                                                      | Resultado  |
|-----------|-------------------------------------------|--------------------------------------------------------------|------------|
| -eq       | Equals (igual)                            | \$variavel = 5 ; \$variavel -eq 4                            | Falso      |
| -ne       | not equal (diferente)                     | \$ variavel = 5 ; \$variavel -ne 4                           | Verdadeiro |
| -gt       | greater than (maior)                      | \$ variavel = 5 ; \$variavel a -gt 4                         | Verdadeiro |
| -ge       | greater than or equal to (maior ou igual) | \$ variavel = 5 ; \$variavel -ge 5                           | Verdadeiro |
| -lt       | less than (menor)                         | \$ variavel = 5 ; \$variavel -lt 5                           | Falso      |
| -le       | less than or equal to (menor ou igual)    | \$ variavel = 5 ; \$variavel -le 5                           | Verdadeiro |
| -like     | wildcard comparison                       | \$ variavel = "This is Text" ; \$variavel -like "Text"       | Falso      |
| -notlike  | wildcard comparison                       | \$ variavel = "This is Text" ; \$variavel -notlike "Text"    | Verdadeiro |
| -match    | regular expression comparison             | \$ variavel = "Text is Text" ; \$variavel -match "Text"      | Verdadeiro |
| -notmatch | regular expression comparison             | \$ variavel = "This is Text" ; \$variavel -notmatch "Text\$" | Falso      |

Tabela 2. Exemplos de uso de operadores lógicos

Note que o ambiente está organizado em três seções horizontais, sendo a primeira seção o arquivo onde será criado o script, a segunda é onde será apresentado o resultado da execução do mesmo e a última seção representa o prompt do Windows PowerShell.

Agora, ao invés de utilizar o prompt de comando do Windows PowerShell, vamos criar nosso script dentro do ISE. Assim sendo, dentro do arquivo *Untitled1.ps1*, que foi aberto ao executar o ISE, escreva o código da **Listagem 1**.

**Listagem 1.** Exemplo de criação de um script no ISE.

```
$data = Get-Date 02/26/1985
if ($data.DayOfWeek -eq (Get-Date).DayOfWeek)
{
    Write-Output "O dia armazenado na variável é o mesmo dia da data de hoje"
} else {
    Write-Output "O dia armazenado na variável não é o mesmo dia da data de hoje"
}
```

Logo é possível notar (na **Figura 7**) que as palavras ganham destaque por meio de cores. Isto ajuda a identificar palavras reservadas e nomes de variáveis. Outro ponto interessante dentro do ambiente é o uso da tecla <TAB>, que assim como no PowerShell, facilita a interação dos comandos com o usuário.

Após a digitação do código, clique no botão *Run Script* ou simplesmente aperte a tecla *F5*.

Pode-se observar na **Figura 8** que na segunda parte horizontal a saída do comando foi impressa. Já na terceira parte horizontal, pode-se utilizar um prompt para manipular a variável e/ou comandos sem a necessidade de alterar o script, permitindo assim fazer melhores testes.

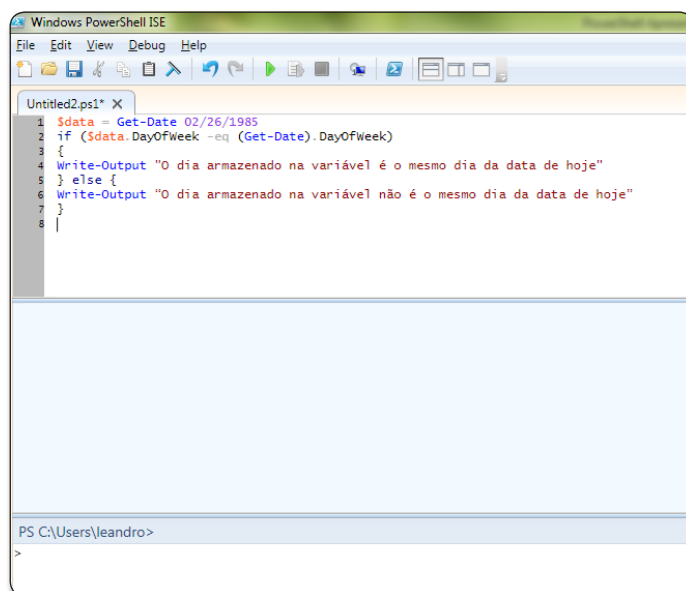
Para finalizar, o script pode ser salvo em um arquivo para ser executado sempre que necessário.

Diferentemente de quando utilizávamos apenas o prompt do Windows PowerShell, com a ferramenta ISE é possível salvar o script com a extensão *.ps1*. Para isso, basta acessar o menu *File*, clicar na opção *Save* e dar um nome ao arquivo.

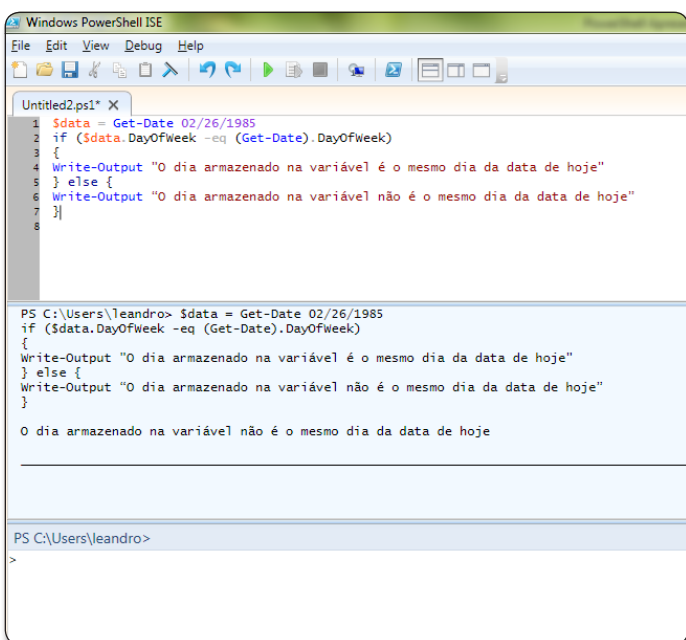
Antes de tentar executar o script fora do Ambiente de Script Integrado, é importante saber que por padrão o Windows PowerShell possui a política de execução de script restrita ou desabilitada, ou seja, diferente dos comandos digitados ao longo do artigo, o Windows PowerShell não permite executar scripts (arquivos preparados com uma sequência de comandos). Com isso, mesmo que você esteja como administrador, não será possível executar o script.

Para verificar a configuração da política de execução de script, digite no Windows PowerShell o *cmdlet* **Get-ExecutionPolicy**. Por padrão esta vem parametrizada como **Restricted** (desabilitada). Para modificá-la basta usar o comando **Set-ExecutionPolicy** passando como parâmetro a nova regra para execução dos scripts. A seguir são destacadas as regras mais utilizadas:

- **Restricted:** Não permite a execução de nenhum script;
- **Unrestricted:** Permite rodar todos os scripts e arquivos de configuração;



**Figura 7.** Ambiente de Script Integrado utilizado para criação de um script



**Figura 8.** Ambiente de Script Integrado utilizado para execução de um script

- **AllSigned:** Requer que todos os scripts e, inclusive arquivos de configuração, sejam assinados por um autor de confiança com um certificado digital;
- **RemoteSigned:** Requer que todos os scripts baixados da Internet sejam assinados por um autor de confiança.

Como estamos trabalhando em um ambiente de teste, podemos utilizar a regra de execução **Unrestricted**, que permite executar qualquer script sem fazer a verificação de confiança. Assim, dentro do prompt de comando do Windows PowerShell, informe: **Set-ExecutionPolicy Unrestricted**.

Agora, basta executar o script criado. Para isso, no prompt do Windows PowerShell, digite o comando: `powershell.exe MeuScriptPS.ps1`

### Conclusão

Como dito no artigo, o Windows PowerShell é hoje a ferramenta escolhida pela Microsoft para gerenciamento dos servidores e também dos novos produtos como Microsoft Exchange, Microsoft Lync Server, SharePoint, Windows Server Core, entre outros.

Ao longo do artigo foi possível observar que o Windows PowerShell nos permite um aprendizado muito rápido, em função do mesmo utilizar verbos auxiliares no início dos *cmdlets*. Com isto os comandos aproximam-se da nossa linguagem, facilitando o entendimento dos mesmos para a criação e manutenção de scripts.

Finalizando, cabe destacar que esse artigo abordou o primeiro contato com o Windows PowerShell, exemplificando em pequenos códigos o uso de *cmdlets* que ajudam no entendimento de outros comandos, facilitando o conhecimento e exploração da linguagem. Abordou também o uso de scripts para automatização de tarefas e apresentou uma ferramenta capaz de ajudar na confecção e gerenciamento dos scripts.



**Leandro Marcos**

[leandrofmarcos@gmail.com](mailto:leandrofmarcos@gmail.com)

Atuo no ramo de tecnologia e desenvolvimento há mais de 7 anos. Instrutor Microsoft desde 2007, ministrando treinamentos em diversos centros autorizados. Certificado MCP, MCTS Windows 7, MCSE, MCTS .NET, MCDBA 2000, MCITP Server Administrator on Windows Server 2008, MCT, ITIL v2 Foundation e LPI 101.



# FÓRUM

# DEV MEDIA

O lugar perfeito para você ficar por dentro de tudo o que acontece nas tecnologias do mercado atual



No novo Fórum da DevMedia você vai encontrar canais específicos de Delphi, ASP.NET, Java, Banco de Dados e Engenharia de Software; além de ter contato com profissionais qualificados da área para troca de informações, sugestões e muito mais.

**ACESSE AGORA**

[www.devmedia.com.br/forum](http://www.devmedia.com.br/forum)

# Segurança e IPv6

## O IPv6 é mais seguro que o IPv4?

O IPv6 (Internet Protocol version 6) tem sido um dos temas mais discutidos na área de redes de computadores e telecomunicações desde 2010. Em fevereiro de 2011, a IANA (Internet Assigned Numbers Authority) distribuiu os últimos endereços IPv4 (Internet Protocol version 4) disponíveis, alarmando a todos sobre a necessidade imediata de migração do IPv4 para uma versão mais nova do protocolo. O IPv6 traz diversas melhorias com relação ao IPv4 como maior número de endereços, autoconfiguração, mobilidade e suporte à segurança. Porém, o IPv4 tem sido usado desde o início da Internet, o que o caracteriza como um protocolo maduro e bastante testado, trazendo preocupação com relação à sua transição para uma solução mais imatura como o IPv6.

Um dos pontos chave desta transição é a segurança. Muitos têm apostado que o IPv6 é naturalmente mais seguro que o IPv4, se baseando, provavelmente, no suporte obrigatório que o IPv6 oferece ao IPSec. Contudo, o IPv6 traz novidades para as quais as equipes técnicas e os equipamentos de segurança ainda não estão bem preparados. Este artigo apresentará os maiores riscos de segurança envolvidos na implantação do IPv6, assim como algumas soluções e precauções que podem ser tomadas para controlar os riscos que surgirão com a transição entre as duas versões do IP.

### Evolução do IP

Em 1981, quando foi publicado o RFC 791 com a especificação do IPv4, não tinha sido vislumbrada a real dimensão que a utilização deste protocolo iria tomar em aproximadamente 15 anos. Os endereços tinham 32 bits e os pesquisadores envolvidos no desenvolvimento do protocolo não imaginavam que isto seria insuficiente. Também não houve preocupação com segurança à época. A consolidação de uma arquitetura de segurança para o IP ocorreu apenas em 1994, início da comercialização de acessos à Internet.

Atualmente, o conjunto de protocolos TCP/IP é um dos principais atores de uma mudança de paradigma na indústria e no mercado de telecomunicações. A Internet cresceu de maneira vertiginosa nos últimos 20 anos, atingindo mais de 2 bilhões de usuários. Inicialmente, o TCP/IP e a Internet eram utilizados apenas para distribuição de conteúdo baseado em textos e

### Resumo DevMan

#### *De que se trata o artigo:*

Este artigo apresenta as principais questões relacionadas à segurança em redes IPv6. Primeiramente, serão descritas as principais novidades do IPv6 com relação ao IPv4. Também serão apresentados o framework de segurança IPSec, cujo suporte é obrigatório nos nós das redes IPv6, e as principais vulnerabilidades encontradas nestas redes. Por fim, serão analisadas soluções de segurança que podem ser adotadas para amenizar o risco trazido pelas vulnerabilidades presentes nas redes IPv6.

#### *Em que situação o tema é útil:*

O suporte obrigatório que os nós IPv6 devem fornecer ao IPSec faz com que muitos técnicos acreditem que o IPv6 é automaticamente mais seguro que o IPv4. Contudo, isto não é verdade. A utilização de IPSec em todas as conexões é inviável devido a sua complexidade. Além disso, novidades inclusas no IPv6 como a autoconfiguração criaram novas vulnerabilidades que podem ser exploradas por agentes maliciosos. Desta forma, este artigo tem por objetivo apresentar as principais vulnerabilidades presentes nas redes IPv6 e algumas soluções que podem ser adotadas para diminuir os riscos de ataques.

#### *Segurança e IPv6:*

Com o esgotamento dos endereços IPv4, a adoção do IPv6 é obrigatória para que as redes IP e a Internet continuem seu forte movimento de expansão. No entanto, esta transição traz muitos desafios para os especialistas, e uma das principais preocupações diz respeito à segurança. Deste modo, o projeto do IPv6 dedicou mais atenção a este requisito, tornando obrigatório o suporte ao IPSec em nós IPv6. Entretanto, o IPv6 traz novidades como a autoconfiguração, que introduzem novas vulnerabilidades. Portanto, é importante conhecer quais são as possíveis vulnerabilidades nas redes IPv6 e como solucioná-las, fazendo com que a transição do IPv4 para o IPv6 não cause problemas de segurança.

figuras. Hoje, o TCP/IP e a Internet são veículo para transporte de voz e vídeo em sessões de comunicação de tempo real. Esta evolução está levando à convergência das redes de transmissão de dados, voz e vídeo, resultando em uma rede multiserviços. O ponto de convergência que torna isto possível é o IP, que foi criado justamente para permitir que haja comunicação entre redes baseadas em diferentes tecnologias de camada física e camada de enlace.

Todo este sucesso alcançado pelo IP trouxe um problema: o IPv4 não foi preparado para ser o ator principal de um cenário no qual temos mobilidade, um número altíssimo de usuários (humanos ou não), necessidade de conectividade fim a fim, qualidade de serviço e segurança. A necessidade de transição do IPv4 para uma nova versão do protocolo se acentuou quando, no início de 2011, a IANA (Internet Assigned Numbers Authority) distribuiu os últimos endereços IPv4 disponíveis para as entidades de registro regionais.

A versão mais nova do IP, conhecida como IPv6, foi projetada para atender as demandas mais modernas que têm sido apresentadas às redes IP. Primeiramente, o IPv6 traz endereços de 128 bits, que proporcionam um espaço de endereçamento inesgotável. Também houve preocupação em projetar a nova versão do protocolo para que ela oferecesse suporte à segurança, mobilidade e qualidade de serviço. Entretanto, a troca do IPv4 pelo IPv6 não é uma tarefa simples. O IPv4 vem sendo utilizado desde o início da Internet, e por isso constitui uma solução madura, testada nas mais diversas situações. Além disso, as equipes técnicas e todo aparato de gerência e segurança estão preparados para trabalhar com IPv4. Por isso, há pelo menos uma área em que a entrada do IPv6 deve trazer maiores preocupações: a segurança.

Há a disseminação de um mito no qual se afirma que o IPv6 é mais seguro que o IPv4 por dar obrigatoriamente suporte ao IPSec. Mesmo oferecendo maior suporte à segurança do que ocorria com o IPv4, o IPv6 não proporciona automaticamente mais segurança que seu antecessor. A transição entre os dois protocolos, por exemplo, vai criar situações inéditas para os especialistas em segurança, que já estavam acostumados a trabalhar com o IPv4 e agora terão que manter estruturas de segurança simultâneas para duas versões do IP. Todos os equipamentos utilizados para defender as redes de computadores, como firewalls e sistemas de detecção de intrusão, precisarão ser reestruturados para lidar com o novo protocolo. Além disso, o protocolo IPv6 traz funcionalidades novas, como a mobilidade e a auto configuração, que podem ser usadas como brechas para ataques.

IPv6

O IPv6 é bastante diferente do IPv4. Os responsáveis pelo projeto se preocuparam em aumentar o número de endereços possíveis, em possibilitar o provimento de qualidade de serviço e em melhorar o suporte à segurança e à mobilidade. O cabeçalho do IPv6 é apresentado na **Figura 1**.

O cabeçalho IPv6 é bem diferente do cabeçalho IPv4. O cabeçalho IPv6 não possui campos para controle da fragmentação. Este controle é feito nos cabeçalhos de extensão, uma novidade do IPv6. Os cabeçalhos de extensão são extensões que podem ser aplicadas ao cabeçalho IPv6 e são transportadas como payload. Desta forma, os roteadores só abrem o cabeçalho de extensão se for algo que lhes interesse. A utilização dos cabeçalhos de extensão permite que o cabeçalho principal do IPv6 tenha tamanho fixo, o que facilita o trabalho dos roteadores. No caso da fragmentação, não há intervenção dos roteadores. Caso o pacote atinja um ponto

da rede que tenha o MTU (*Maximum Transmission Unit*) do enlace menor que o tamanho do pacote, é gerada uma mensagem ICM-Pv6 do tipo Packet too big e o pacote é descartado. Estas soluções visam facilitar o trabalho dos roteadores, trazendo melhoras de desempenho.

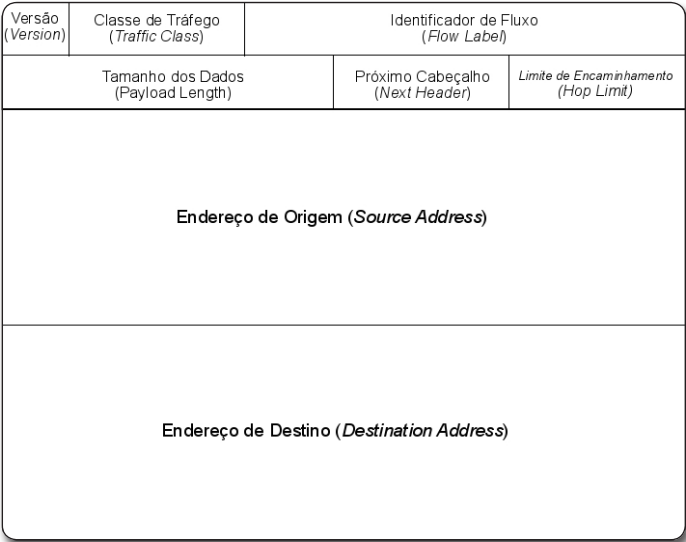


Figura 1. Cabeçalho do IPv6

O cabeçalho IPv6 tem campos para identificador de classe de tráfego e de fluxo, que permitem a identificação dos pacotes para que eles sejam tratados de maneira diferenciada, de forma a garantir qualidade de serviço para aplicações mais sensíveis, como transmissão de vídeo e voz. Estes campos não existiam no cabeçalho IPv4.

Outra mudança importante no cabeçalho IPv6 é o tamanho dos campos de endereço, que possuem 128 bits. Nas redes IPv6, o endereço é representado por 8 grupos de 16 bits separados por “:”. Cada grupo de 16 bits é representado como um hexadecimal. Um exemplo de endereço IPv6: 2001:0DB8:AD1F:25E2:CADE:CAF E:F0CA:84C1. A identificação da sub rede e do host no endereço é realizada da mesma forma que no IPv4. É utilizada a notação CIDR (Classless Inter-Domain Routing) <endereço IP>/<número de bits>. O número de bits colocado após a barra indica a quantidade de bits do endereço utilizada para representar a sub rede.

Os tipos de endereços também mudaram do IPv4 para o IPv6. Há três tipos de endereço no IPv6: unicast, anycast e multicast.

O endereço unicast identifica uma única interface e pode ser classificado em global unicast, link local e site local. Os endereços global unicast são similares aos endereços públicos do IPv4, ou seja, são globalmente roteáveis. Os endereços link local são utilizados apenas localmente, no mesmo enlace onde a interface está conectada. O endereço site local abrange todo o local onde o nó está localizado, mas não é roteável globalmente, assim como os endereços privados do IPv4.

Os endereços anycast são utilizados para identificar um grupo de interfaces. Portanto, os endereços anycast identificam grupos

de endereços unicast. Os endereços anycast podem ser utilizados para identificar grupos de servidores na rede como os servidores DNS ou DHCP. Quando um pacote tem como destino um endereço anycast, ele será roteado até a interface pertencente ao grupo anycast que esteja mais próxima da origem do pacote. Um endereço anycast não pode ser utilizado como endereço de origem de um pacote.

Assim como um endereço anycast, um endereço multicast identifica um grupo de interfaces. A diferença é que um pacote para um endereço multicast é entregue a todas as interfaces pertencentes ao grupo. No IPv6 não há endereço de broadcast. Quando se deseja enviar um pacote para vários computadores, a única opção é enviar este pacote para um grupo multicast.

Outra novidade do IPv6 é a auto configuração, que pode ser dividida em duas categorias: stateless e stateful. A auto configuração stateless permite que endereços sejam atribuídos aos hosts sem a necessidade de serviços como o DHCP. Primeiramente, o host cria um endereço unicast link local e verifica com os seus vizinhos se este endereço não é duplicado. Caso o endereço escolhido não esteja duplicado, o host pode utilizá-lo para enviar uma mensagem Router solicitation com destino ao endereço multicast all-routers. Nesta mensagem, o host requisita informações dos roteadores que serão utilizadas para determinar qual é o roteador padrão e gerar endereços com base no prefixo deste roteador e na identificação da interface. A auto configuração stateful utiliza o DHCPv6 para obter os endereços que podem ser utilizados e as informações do roteador padrão e de serviços essenciais como o DNS.

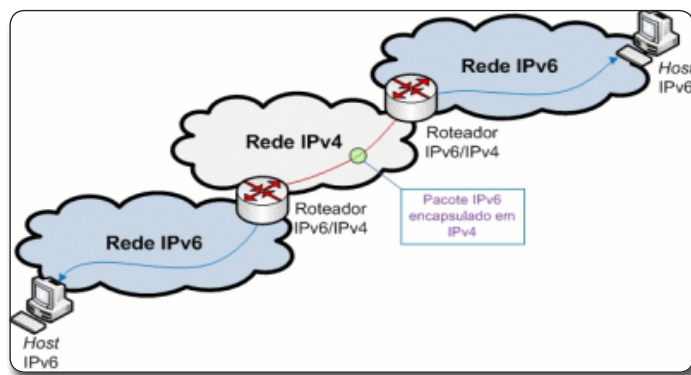
A adoção do IPv6 e o conseqüente abandono do IPv4 não serão completados em um curto espaço de tempo. Durante um longo período, redes baseadas em IPv4 e IPv6 terão de conviver. Portanto, é necessário definir estratégias de transição entre os dois protocolos para que as redes continuem operando normalmente mesmo em cenários heterogêneos. Há três técnicas mais comuns:

- **Pilha dupla:** os dispositivos são compatíveis com o IPv4 e o IPv6. O Windows 7 e o Apple OS X, por exemplo, oferecem suporte para estas duas versões do protocolo;
- **Tunelamento:** o tunelamento consiste do transporte de um cabeçalho IPv4 como payload do IPv6 em redes IPv6 e do transporte do cabeçalho IPv6 como payload do IPv4 em redes IPv4. A **Figura 2** mostra um cenário possível de tunelamento. Quando o pacote chega ao roteador posicionado na fronteira entre as redes IPv6 e IPv4, é inserido um cabeçalho IPv4 abaixo do cabeçalho IPv6, permitindo que este pacote seja roteado na rede IPv4 até que atinja o roteador que leva o pacote de volta a uma rede IPv6;
- **Tradução:** a primeira possibilidade é que cabeçalhos IPv6 sejam inteiramente traduzidos para cabeçalhos IPv4. O mesmo ocorrerá com APIs de programação que lidam com IPv4 e passarão a lidar com IPv6.

## IPSec

Quando o IPv4 foi criado, no início da década de 80, não houve preocupação em desenvolver soluções de segurança para o protocolo. As primeiras aplicações que utilizavam o IP envolviam a

conexão de laboratórios acadêmicos e, provavelmente, os pesquisadores não imaginavam que o IPv4 seria a base de uma rede de escala mundial como a Internet. Com o início da popularização da Internet, já na década de 90, os pesquisadores perceberam que o IP poderia ser usado em transações sensíveis do ponto de vista de segurança.



**Figura 2.** Tunelamento na transição IPv4/IPv6

Em 1994, a IAB (Internet Architecture Board) publicou um relatório denominado Security in the Internet Architecture (RFC 1636). O relatório estabeleceu os principais pontos que deveriam ser tratados para a melhoria de segurança na Internet. Em resumo, o documento determinava que era necessário utilizar técnicas de cifragem e autenticação para evitar que conexões fim a fim fossem controladas ou monitoradas por agentes não autorizados. As soluções levantadas no documento seriam incluídas na versão 6 do IP. Entretanto, elas foram projetadas de forma que já pudessem ser utilizadas pela versão 4. A estes padrões de segurança para o IP damos o nome de IPSec.

O IPSec é um framework de segurança, do qual devemos destacar dois protocolos: o AH (Authentication Header) e o ESP (Encapsulating Security Payload). O AH oferece autenticação e integridade das mensagens, mas não proporciona confidencialidade. O ESP oferece uma combinação de autenticação, integridade das mensagens e confidencialidade. Por oferecer confidencialidade, o ESP é bem mais utilizado que o AH. Porém, o AH pode ser utilizado em conexões que não precisam de confidencialidade, como a descoberta de vizinhos do IPv6.

Há duas formas de utilização do IPSec: o modo túnel e o modo transporte. No modo transporte, a preocupação é oferecer segurança para as camadas superiores à camada de rede. Neste caso, temos a cifragem de todo o payload, mas não temos a cifragem do cabeçalho IP. No modo túnel, tanto o cabeçalho quanto o payload IP são cifrados.

No IPv4, o suporte a IPSec não é obrigatório e os cabeçalhos do IPv4 e dos protocolos do IPSec se combinam de maneiras diferentes para os modos transporte e túnel. A **Figura 3** ilustra como os cabeçalhos do ESP e do IPv4 são posicionados no modo transporte. Podemos observar que temos, primeiramente, o cabeçalho IPv4 original do pacote a ser transmitido, que não é cifrado. Logo após o cabeçalho IPv4, vem o cabeçalho ESP, seguido pelo payload do

pacote IP, que está cifrado. A **Figura 4** mostra como os cabeçalhos do ESP e do IPv4 se posicionam no modo túnel. É criado um novo cabeçalho IPv4, cujo endereço de origem é a entrada do túnel e o endereço de destino é a saída do túnel. O cabeçalho ESP é posicionado após o novo cabeçalho IPv4 e é seguido pelo cabeçalho original IPv4 do pacote a ser transmitido. Logo após o cabeçalho IPv4 original, temos o payload do pacote IP. Tanto o cabeçalho original IPv4 como o seu payload são cifrados.

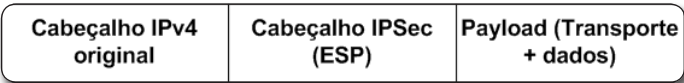


Figura 3. IPSec e IPv4: modo transporte

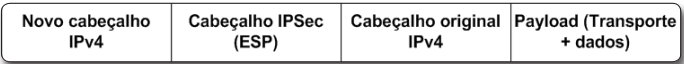


Figura 4. IPSec e IPv4: modo túnel

No IPv6, o suporte ao IPSec é obrigatório. Os cabeçalhos dos protocolos IPSec, tanto o AH como o ESP, são inseridos nos cabeçalhos de extensão do IPv6, que são posicionados como payload do pacote IP. No caso do AH, o campo Next Header do cabeçalho IPv6 vai receber o valor 51. Quando o cabeçalho de extensão contiver um cabeçalho ESP, o valor do campo Next Header será igual a 52.

O AH, no modo transporte, é adicionado após as extensões Hop by hop options, Routing options e Fragmentation options, caso elas existam. O cabeçalho de extensão Destination options pode ser posicionado após o cabeçalho de extensão AH. No modo túnel, é criado um novo cabeçalho IPv6 e podem ser criadas também novas extensões. O cabeçalho IPv6 original e os cabeçalhos de extensão originais são posicionados após o AH. A **Figura 5** mostra como fica a disposição dos cabeçalhos com a utilização do AH no modo transporte e no modo túnel com o IPv6.

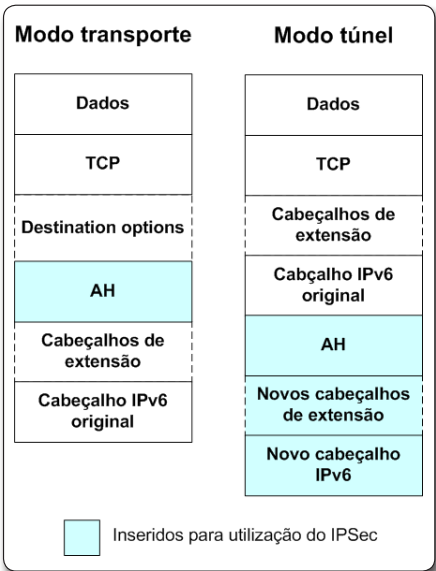


Figura 5. AH com IPv6

O ESP, no modo transporte, também é adicionado após os cabeçalhos de extensão Hop by hop options, Routing options e Fragmentation options, caso eles estejam sendo utilizados. O cabeçalho de extensão Destination options pode ser adicionado depois do ESP. No modo túnel, é criado um novo cabeçalho IPv6 e cabeçalhos de extensão novos também podem ser criados. O cabeçalho e as extensões originais são posicionados após o ESP. A **Figura 6** apresenta como fica a disposição dos cabeçalhos com a utilização do ESP no modo transporte e no modo túnel.

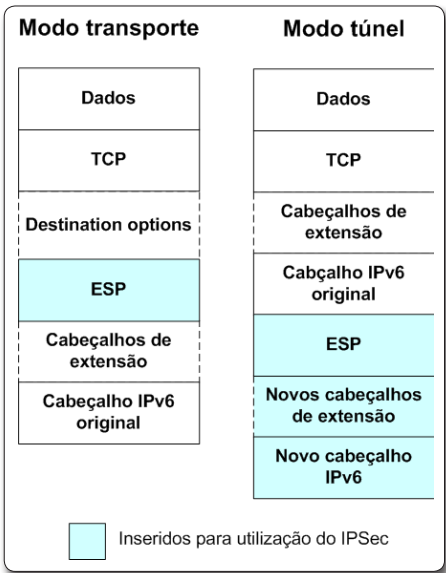


Figura 6. ESP com IPv6

As conexões IPSec são construídas por meio das associações de segurança. Uma associação de segurança é uma conexão simplex, isto é, ela tem apenas um sentido. Se os dois nós desejam enviar pacotes por meio de uma conexão segura, são necessárias duas associações de segurança, uma em cada sentido. No estabelecimento da associação de segurança são definidos os parâmetros de segurança da conexão como algoritmo de criptografia que será utilizado, chaves de segurança, entre outros. A definição das chaves pode ser manual ou automática. Caso ela seja manual, o administrador de rede deve cadastrar os parâmetros de segurança nos nós que participam da conexão segura, tarefa impraticável para situações onde há muitos nós utilizando IPSec. Na configuração automática, é utilizado o protocolo de troca de chaves IKE (Internet Key Exchange), previsto no próprio framework IPSec.

É possível observar que a utilização do IPSec no IPv4 e no IPv6 é ligeiramente diferente, já que no IPv6, o IPSec utiliza a estrutura de cabeçalhos de extensão, não existente no IPv4. No IPv6, o suporte ao AH e ao ESP já esta previsto e é obrigatório.

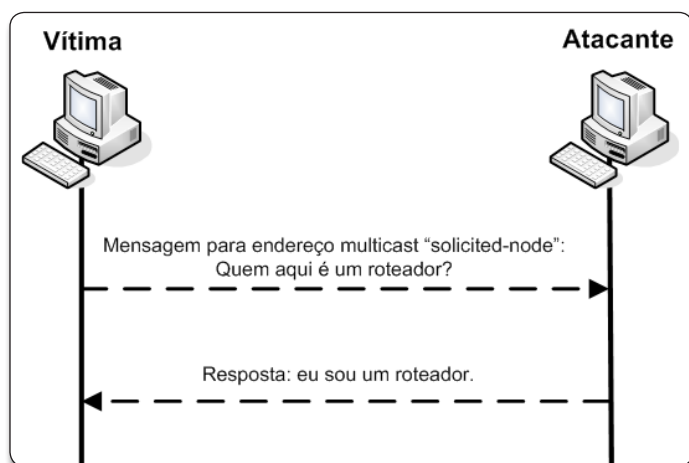
Ameaças à segurança das redes IPv6

O suporte obrigatório que o IPv6 deve oferecer ao IPSec não o torna um protocolo automaticamente mais seguro que o IPv4. Portanto, os problemas de segurança nas novas redes IPv6 merecem especial

atenção. O primeiro ponto que deve ser esclarecido é que o IPSec não deve ser utilizado em todas as conexões baseadas em IPv6. Isto seria impossível, pois a utilização do IPSec em larga escala é complexa. A segunda razão é que a utilização de IPSec não corrige todos os problemas de segurança possíveis. Mesmo se ele pudesse ser utilizado em todas as conexões IPv6, ainda restariam alguns tipos de ataques que não seriam evitados. Além disso, há novas formas de ataque que devem ser criadas em cima das novas funcionalidades do IPv6 e ataques que já existiam no IPv4 que devem tomar novas formas para continuarem presentes no IPv6. A falta de mão de obra especializada e de ferramentas de monitoramento e proteção voltadas para redes IPv6 pode tornar o cenário ainda mais complicado. Mesmo que tenha havido uma preocupação maior com segurança no desenvolvimento do IPv6, se não entendermos bem as implicações da adoção do IPv6 na área de segurança, não conseguiremos desfrutar dos benefícios que a nova versão do protocolo nos traz.

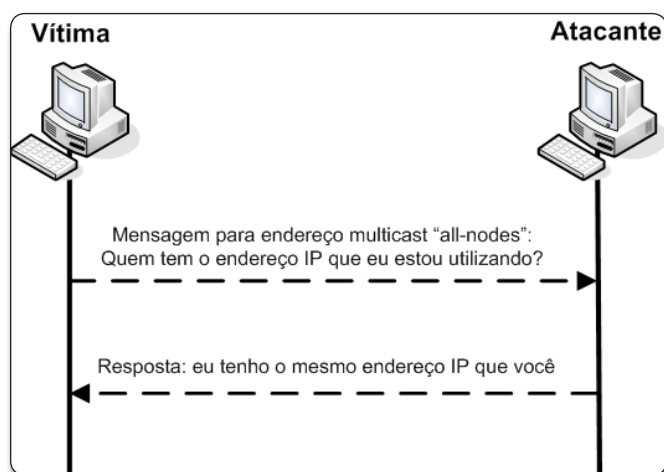
Alguns atributos que são novidade no IPv6 podem ser utilizados como brechas de segurança. A primeira delas são os endereços multicast e anycast. Por meio destes endereços, é possível identificar rapidamente os hosts que hospedam serviços importantes como DHCP e DNS ou dispositivos como roteadores. O projeto do IPv6 buscou a diminuição da intervenção humana na configuração dos hosts, tornando automática a localização dos serviços e dispositivos mais importantes para o bom funcionamento da rede.

A auto configuração stateless trazida pelo IPv6 é um ponto de vulnerabilidade para outros tipos de ataques. As mensagens ICMPv6, utilizadas para a troca de informações entre hosts durante a auto configuração, podem ser utilizadas por agentes maliciosos, mas não podem ser bloqueadas por serem essenciais ao funcionamento da rede. Um host mal intencionado pode gerar mensagens ICMPv6 fazendo com que outros nós passem a tratá-lo como roteador, conforme apresentado na **Figura 7**. Uma solução é utilizar o DHCPv6 para alimentar os hosts com informações sobre os roteadores, ao invés de usar a solução stateless, que é totalmente automatizada. Ainda utilizando o ICMPv6, um agente mal intencionado pode gerar um grande volume de mensagens para causar sobrecarga em toda a rede.



**Figura 7.** Ataque no qual agente malicioso se passa por roteador

Uma possível forma de ataque de negação de serviço explora o processo de descoberta de endereço duplicado utilizado na auto configuração. O host que está se configurando envia à rede uma mensagem ICMPv6 do tipo Neighbor Solicitation (NS) com o endereço que pretende usar. Um agente mal intencionado pode mentir, enviando uma mensagem ICMPv6 do tipo Neighbor Advertisement (NA) na qual diz que já está utilizando o endereço. Quando recebe esta mensagem, o *host* que havia enviado a mensagem NS gera outro endereço e envia nova mensagem NS para conferir se não há duplicação de endereço. O agente mal intencionado pode novamente mentir. Este fato pode se repetir indefinidamente, negando o direito de ingresso na rede ao host interessado (**Figura 8**).



**Figura 8.** Ataque de negação de serviço utilizando o mecanismo de detecção de endereços duplicados

No IPv4, quando o host vai enviar o pacote e tem de colocar o endereço físico do destino, é utilizado o ARP (Address Resolution Protocol) para descobrir qual é o endereço físico correspondente a um determinado endereço IP. No ARP, o pedido é enviado por meio de uma mensagem de broadcast. No IPv6, não há endereço de broadcast. Desta forma, um endereço multicast é utilizado para a descoberta do endereço físico correspondente a um determinado endereço IP. O host envia uma mensagem ICMPv6 do tipo Neighbor Solicitation com destino ao endereço multicast solicited-node, perguntando pelo endereço físico correspondente a um dado endereço IP. Um atacante pode responder esta solicitação passando seu endereço físico como resposta. A partir daí, o atacante passa a ser o *man in the middle*, interceptando os pacotes enviados pelo host que no início enviou a mensagem ICMPv6 em busca do endereço físico do destinatário de seus pacotes (**Figura 9**).

Outra novidade do IPv6 que poderia ser utilizada para ataques é o cabeçalho de extensão *Routing options*. Neste cabeçalho, utilizando o atributo "Type 0", era possível determinar nós intermediários que deveriam ser visitados até que o pacote chegasse a seu destino. O mesmo nó poderia ser incluído várias vezes na lista. Desta forma, um atacante poderia construir um fluxo de pacotes que ficaria continuamente tráfegando entre

dois roteadores até causar um congestionamento ou uma negação de serviço. O atacante também poderia construir pacotes que conseguissem desviar de firewalls ou sistemas de detecção de intrusão. A opção “Type 0”, que permitia este ataque, foi depreciada pelo RFC 5095.

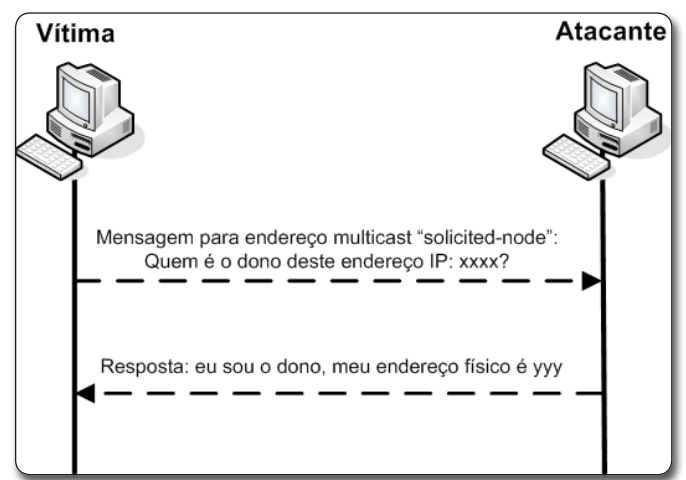


Figura 9. Ataque Man-in-the-middle

As novidades que envolvem o IPv6 trazem outra dificuldade com relação à segurança: encontrar mão de obra especializada. Com a falta de especialistas, acabam ocorrendo erros na configuração de dispositivos responsáveis pela defesa do perímetro da rede como firewalls e sistemas de detecção de intrusão. Outro problema é a presença de dispositivos IPv6 na rede sem que a administração de rede esteja ciente. Isto pode ocorrer já que há sistemas operacionais como o Windows 7, Apple OS X e algumas distribuições do Linux que já são compatíveis com IPv6 automaticamente. Desta forma, quando instalados, estes sistemas operacionais acabam inserindo as vulnerabilidades do IPv6 na rede. Os computadores com IPv6 podem servir como backdoor para ataques (ver **Nota DevMan 1**).

### Nota do DevMan 1

**Backdoor:** Backdoor é um ponto alternativo de acesso a um software ou a uma rede. Por meio de uma backdoor, é possível ingressar no ambiente alvejado sem passar pelas barreiras de segurança.

A transição também poderá trazer problemas de segurança. Uma das técnicas de transição é o tunelamento, na qual um pacote IPv6 é transportado sobre um pacote IPv4 para poder trafegar em uma rede IPv4. Se no caminho tivermos um firewall que não é ciente da presença do IPv6, ele só analisará o pacote IPv4 e o pacote IPv6 será liberado sem análise. Quando a abordagem de transição é a pilha dupla, o administrador de rede se depara com o problema de manter estruturas de segurança simultaneamente para duas redes diferentes.

Aumentando a segurança em redes IPv6

A infinita variedade de endereços oferecida pelo IPv6 involuntariamente diminuiu as chances de um tipo de ataque. Frequentemente, o primeiro passo de um ataque é a identificação de hosts ativos na rede que será o alvo. O atacante identifica uma sub rede e passa a tentar estabelecer contato com todos os endereços possíveis dentro desta sub rede. Os endereços que respondem à tentativa de contato levam à identificação dos hosts ativos. Com estas informações, o atacante pode realizar um port scan nos hosts encontrados. Com o IPv6, este tipo de ação é mais difícil, já que as faixas de endereços são muito extensas. Mesmo assim, o administrador de rede deve evitar a atribuição de endereços de forma sequencial e ordenada aos hosts, pois isso pode facilitar o trabalho do atacante que está buscando reconhecer hosts ativos para realizar o port scan.

Há algumas medidas que podem ser adotadas de forma a permitir que as novidades do IPv6 sejam utilizadas e os riscos para a segurança sejam controlados. No caso do ICMPv6, por exemplo, há alguns tipos de mensagens que podem ser bloqueados sem trazer grandes prejuízos ao funcionamento da rede. A **Tabela 1** mostra alguns tipos de mensagens que podem ser bloqueadas.

| Tipos de mensagens ICMPv6 que podem ser bloqueadas | Comentários/Justificativas                                           |
|----------------------------------------------------|----------------------------------------------------------------------|
| 137- Redirect                                      | Alertam os nós que devem enviar o tráfego para outro roteador        |
| 138- Router Renumbering                            | Permitem a reconfiguração automática de roteadores                   |
| 139- Node Information Query                        | Permitem que um agente malicioso obtenha as características de um nó |
| 140- Node Information Response                     | Permitem que um agente malicioso obtenha as características de um nó |

Tabela 1. Mensagens ICMPv6 que podem ser bloqueadas

As mensagens 133-Router solicitation, 134-Router Advertisement, 135-Neighbor solicitation e 136-Neighbor Advertisement devem ser liberadas, mesmo que tragam riscos para a rede. Estas mensagens são essenciais para o funcionamento da rede IPv6. Uma forma de proteger a rede mesmo com a utilização destas mensagens é a aplicação do SEND (Secure Neighbor Discovery Protocol - RFC 3971). Neste protocolo, os endereços dos nós são conhecidos como CGA (Cryptographically Generated Addresses). Estes endereços são gerados através da aplicação de funções hash sobre a chave pública e outros parâmetros do dono do endereço. Desta forma, é possível verificar se o endereço pertence mesmo a quem alega ser o dono dele.

O desenvolvimento de firewalls específicos para IPv6 está em seu estágio inicial. Temos algumas implementações prontas em distribuições do Linux e em equipamentos Cisco e Juniper. Seguem outras ferramentas que podem ser utilizadas em redes IPv6:

- **NDPWatch:** monitora as mensagens ICMPv6 e mantém uma base de dados que mapeia quais endereços IP estão relacionados a quais endereços físicos. Quando há mudanças no mapeamento, emite alertas;

- **NDPMon:** monitora a rede e emite alertas quando detecta mensagens de descoberta de vizinhança suspeitas. É similar ao NDPWatch;
- **Nmap:** software para realizar verificação de vulnerabilidades da rede;
- **Snort:** sistema de detecção e prevenção de intrusão bastante utilizado nas redes IPv4, que já possui versões para IPv6.

Nas situações de transição, deve-se evitar os túneis automáticos, adotando, sempre que possível, túneis de configuração manual. Quando a configuração do túnel é manual, é possível controlar o ponto de entrada e saída do túnel de forma que eles estejam cientes que existem pacotes IPv4 e IPv6 trafegando por eles. Por fim, se a abordagem de transição é a pilha dupla, é necessário se assegurar de que os firewalls foram configurados corretamente para proteger as duas redes, a IPv4 e a IPv6.

## Conclusão

O IPv6 traz muitas novidades com relação ao IPv4. Ao oferecer um volume enorme de endereços disponíveis e dar suporte a qualidade de serviço, mobilidade e segurança, o IPv6 mostra que está preparado para as demandas mais modernas que tem sido criadas para as redes IP. Porém, a utilização do IPv4 está profundamente inserida no funcionamento da Internet e abandonar esta versão do protocolo, que é madura e já foi exposta aos mais diversos desafios, tem trazido dificuldades para as equipes técnicas. Um dos pontos que tem gerado questionamento na transição entre as duas versões do IP é a segurança.

Alguns alegam que o IPv6 é mais seguro que o IPv4 por obrigatoriamente oferecer suporte ao IPSec. Porém, não podemos esquecer que o IPSec não pode ser utilizado em todas as conexões, já que demanda bastante esforço com configuração e sua utilização é complexa. Além disso, o IPv6 traz muitos elementos novos, para os quais as equipes técnicas e os equipamentos de segurança ainda não estão bem preparados, o que pode trazer vulnerabilidades para as redes que migrarem para IPv6. Entre os pontos de maior vulnerabilidade do IPv6 estão a auto configuração, os cabeçalhos de extensão e o desafio de configurar e manter equipamentos como firewalls e sistemas de detecção de intrusão para defesa de duas redes simultaneamente: a rede IPv4 e a rede IPv6.

Soluções para os problemas de segurança do IPv6 já tem sido propostas. O SEND é uma solução para diminuir os problemas de segurança originados no protocolo de descoberta de vizinhança existente nos mecanismos de auto configuração. Ferramentas

como o NDPWatch e o NDPMon podem ser utilizadas para monitorar a troca de mensagens ICMPv6 entre os hosts e emitir alarme caso comportamentos maliciosos sejam detectados. Alguns tipos de mensagens ICMPv6 também podem ser bloqueados para diminuir os riscos. Além disso, firewalls específicos para IPv6 já estão disponíveis.

Para garantir que a transição do IPv4 para o IPv6 não represente um aumento nos riscos para a segurança das redes, é necessário que ela seja realizada com critério e planejamento. Os responsáveis pela transição devem estar cientes das vulnerabilidades que envolvem a introdução do IPv6 na rede, buscando soluções para todas elas e aproveitando medidas de segurança que já eram utilizadas com o IPv4 e podem ser adaptadas para os cenários IPv6.



**Bruno Bogaz Zarpelão**

*brunozarpelao@gmail.com*

Bacharel em Ciência da Computação pela Universidade Estadual de Londrina (UEL) e doutor em Engenharia Elétrica pela Universidade Estadual de Campinas (UNICAMP). Atualmente é pesquisador no Laboratório de Redes de Comunicações da Faculdade de Engenharia Elétrica e de Computação da UNICAMP. Atua também como pesquisador associado no Next Generation Networks and Applications Group (NetGNA) da Universidade da Beira Interior em Portugal e como professor do curso de especialização em Redes de Computadores da UEL. Possui experiência de sete anos em pesquisa e desenvolvimento de soluções para gerência de redes de computadores, governo eletrônico e cidades digitais.



## Apostila "Curso IPv6 básico" do NIC.br

<http://curso.ipv6.br>

## Artigo "IPv6 Security Challenges", escrito por Carlos Caicedo, James Joshi e Summit Tuladhar

<http://dx.doi.org/10.1109/MC.2009.54>

## Livro "IPv6 Essentials", escrito por Silvia Hagen

Editora O'Reilly, 2.ª edição, 2006

## Nmap

<http://nmap.org/>

## NDPMon

<http://ndpmon.sourceforge.net/>

## NDPWatch

<http://sourceforge.net/projects/ndpwatch/>

## Snort

<http://www.snort.org/>



# Detectando e prevenindo intrusos com Snort

## Ferramentas de detecção de intrusão adicionam uma nova camada de segurança à rede

Dentro do cenário empresarial, a informação é um componente valioso e imprescindível que muitas vezes determina quem lidera o mercado. Assim, empresas de todos os setores trabalham em cima deste valor tentando aprimorá-lo, enriquecê-lo e torná-lo privado, na medida do possível, especialmente por ser um elemento construído com o mérito de todos os ativos da empresa.

Quando falamos em informação empresarial dentro de um contexto contemporâneo, nos referimos à velocidade com que a informação é tratada, graças à tecnologia atual que permite o acesso de todos os ativos em tempo real. A denominada Era Digital (ver **Nota DevMan 1**) transformou a maneira com que as organizações tratam a informação. As máquinas, por sua vez, complementam e dinamizam toda rotina do ser humano, acelerando e enriquecendo os ativos da empresa.

A Internet é um dos principais responsáveis pela evolução da Era Digital, nos proporcionando novos meios para trocar informações, permitindo a comunicação em tempo real e de forma mais econômica, por utilizar um meio de comunicação já estruturado. Alguns exemplos de novas formas de comunicação são: conexão entre diversas filiais, parceiros e fornecedores (e-business); comércio eletrônico (e-commerce), serviços públicos fornecidos pelo governo (e-governance); etc. Com todas estas possibilidades, surgem também novos métodos de expor nossa informação e, conseqüentemente, são criadas novas formas de se obter a informação de forma inescrupulosa.

Ante todos estes pontos inicialmente mencionados, a seguinte questão é levantada: Qual é o real valor da informação nesta Era Digital, onde os computadores são imprescindíveis em um ambiente corporativo? A informação “é essencial para os negócios de uma organização e conseqüentemente necessita ser adequadamente protegida” [NBR 27002, 2005]. Diante de todos os pontos levantados e desta afirmação acima, percebe-se sua importância.

### Resumo DevMan

#### *De que se trata o artigo:*

Neste artigo serão expostos os conceitos da ferramenta Snort, seu histórico e definições de IDS. Também serão mostradas suas funcionalidades, exemplos práticos e, por fim, recomendações de boas práticas de segurança no uso em produção.

#### *Em que situação o tema é útil:*

O tema é útil para facilitar a compreensão do funcionamento do Snort, bem como sua utilização em ambientes corporativos. As boas práticas de segurança descritas no final do artigo fortalecem a implementação do Snort, reduzindo falsos positivos e otimizando o desempenho da ferramenta.

#### *Detectando e prevenindo intrusos com Snort:*

A palavra “segurança”, na era atual da informática, tornou-se um requisito mínimo em ambientes corporativos. Muitas organizações dependem da segurança para expor suas negociações, transações e/ou parcerias via Internet, para manter a proteção da informação e não perder a credibilidade com o cliente. Neste contexto, os Sistemas de Detecção de Intrusão, ou IDS, podem detectar ataques contra serviços disponíveis na rede privada e Internet e até evitar que o atacante possa efetuar algum dano à rede corporativa. Assim, abordaremos neste artigo o Snort, uma ótima ferramenta IDS.

“As organizações, seus sistemas de informação e redes de computadores são expostos a diversos tipos de ameaças à segurança da informação, incluindo fraudes eletrônicas, espionagem, sabotagem, vandalismo, incêndio e inundação” [NBR 27002, 2005]. É notório que o risco envolve todos os segmentos da organização, cada um com um nível diferenciado. Este nível é a probabilidade de um incidente de segurança ocorrer em um determinado segmento, multiplicado pelo valor da informação no mesmo. Quando um incidente de segurança ocorre, seja ele acidental ou intencional, pode acarretar na perda de valores da organização. O prejuízo pode ser incalculável, principalmente

se o ativo principal da organização for perdido: a informação. A informação digital também está susceptível aos incidentes de segurança e deve ser protegida com o mesmo rigor.

Na área da segurança digital existem diversos aplicativos e sistemas que protegem os níveis de rede da organização. O IDS é um dos sistemas que possui esta função. Sua proteção abrange do nível de enlace ao nível de aplicação da pilha TCP/IP, podendo alertar sobre anomalias na rede e até prevenir certos tipos de ataque.

Neste contexto, o Snort é uma ferramenta com funções de IDS via rede que, em comparação com os concorrentes de código-aberto, possui uma grande popularidade. O motivo é o seu método de funcionamento, que não demanda grande poder de processamento.

Com base nisso, neste artigo será abordada a estrutura de funcionamento do Snort, demonstrando a função de cada segmento que compõe a ferramenta. Também será abordado um exemplo prático com detalhes, bem como boas práticas de segurança.



## Nota do DevMan 1

**Era Digital:** Também chamada de Era da Informação, esta é a era caracterizada pela habilidade de o indivíduo trocar informações livremente, além de ter acesso instantâneo à informação. Algo que poderia ser difícil ou até impossível de ser realizado no passado.

## Histórico

Lançada em 1998 por Martin Roesch, a ferramenta Snort (gratuita e de código-livre) foi uma das primeiras em seu segmento a realizar análise de tráfego em tempo real e registro dos pacotes de forma leve, utilizando recursos mínimos de processamento. À medida que o Snort foi amadurecendo, foi se tornando o padrão em IDS.

Mais tarde, em 2001, Martin Roesch fundou a Sourcefire com o intuito de atender a toda a demanda oferecida pela versão comercial do Snort. Pelo fato de as ameaças encontradas requererem uma resposta dinâmica, a Sourcefire reforça a segurança com o Time de Pesquisa de Vulnerabilidades Sourcefire (Vulnerability Research Team – VRT), que é uma equipe formada por especialistas em segurança que proativamente descobrem vulnerabilidades, malwares, atividades hackers e outras atividades maliciosas na rede. Em consequência, são criadas regras para o Snort identificar os pacotes suspeitos.

## Snort

O Snort é uma ferramenta de código aberto que atua como IDS via rede (também chamado de NIDS – Network Intrusion Detection System), combinando inspeções em protocolos e assinaturas e, ainda, podendo identificar o ataque através do comportamento anormal do tráfego da rede. A forma de detecção dos ataques é baseada em regras (também chamadas de assinaturas), podendo ser adquiridas em [1]. O registro para obtê-las é obrigatório.

A eficiência da ferramenta se deve à equipe VRT, que atua ativamente para detectar as vulnerabilidades e criar as regras para

proteger a rede. As regras são lançadas mensalmente. Na versão gratuita é possível obter as regras com um mês de atraso com relação às regras disponíveis na versão comercial.

Nas primeiras versões, o Snort possuía apenas a função de sniffer (ver **Nota DevMan 2**). Com os aprimoramentos, passou a gerar logs dos pacotes coletados para uma melhor compreensão e, com suas funções descentralizadas, obteve a função de IDS.



## Nota do DevMan 2

**Sniffer:** Sniffer é um analisador de pacotes que possui a característica de coletar pacotes que trafegam na interface de rede do computador para fins de depuração.

## Componentes

Assim que os pacotes adentram a interface de rede, o Snort trata os pacotes de acordo com a **Figura 1**. Cada bloco possui uma função específica, auxiliando o Snort a realizar suas tarefas com agilidade. As definições de cada bloco são comentadas a seguir:

- **Biblioteca de Captura de Pacotes:** é um software a parte que insere os pacotes advindos da camada de enlace para o Snort. No Linux é o libpcap que faz esta tarefa, e no Windows é o WinPcap;
- **Decodificador de Pacotes Snort:** o decodificador recebe os dados, separa os quadros da camada de enlace da pilha TCP/IP, os pacotes IP, TCP e UDP, e classifica-os para uma análise posterior. Outros tipos de pacotes como IPX e AppleTalk não são reconhecidos;
- **Pré-processadores:** são plug-ins que operam em cima dos pacotes decodificados, desempenhando uma série de transformações, o que torna o acesso à leitura dos dados mais rápido para o Snort. Pré-processadores podem classificar, habilitar alertas ou descartar pacotes antes que sejam enviados à *Engine*. Ademais, podem ser habilitados ou desabilitados, a gosto do analista;
- **Deteção (Engine):** este é o núcleo do Snort. Ele coleta as informações do pré-processador e atua nas camadas de transporte e aplicação da pilha TCP/IP, comparando o conteúdo das informações dos pacotes com os *plug-ins de Detecção*. Nesses plug-ins estão as regras dos ataques conhecidos;
- **Saída (plug-ins):** quando um pré-processador ou regra é acionado, é gerado um alerta (ou uma resposta ao ataque) e um registro no log. O Snort suporta diversas formas de registro de log.

## Coletando os pacotes

A coleta dos pacotes pode ser feita de três formas:

- **Modo bridge:** permite que o fluxo de dados trafegue pelo Snort de forma transparente, fazendo uma espécie de “grampo” no segmento selecionado. Para tal, são necessárias três placas de rede, sendo duas para a bridge e uma para análise ou transmissão dos alertas/logs;
- **Modo espelhamento:** switches gerenciáveis possuem um recurso chamado de *port monitoring* ou *port mirroring*, com a função de

obter todos os pacotes que passam pelas portas do equipamento e copiá-los para a porta configurada com a característica de espelhamento. A desvantagem de utilizar este recurso é a sobrecarga da CPU do switch, que acaba tornando a rede mais lenta. Em alguns casos, se a sobrecarga da CPU for alta, o *port mirroring* é desabilitado automaticamente, prejudicando a coleta do tráfego espelhado. Neste caso, duas placas de rede serão necessárias, sendo uma para coletar os pacotes e outra para análise ou transmissão dos alertas/logs;

• **Modo TAP:** neste modo um equipamento é instalado entre o switch e o servidor Snort, com a finalidade de copiar o tráfego ou até fazer a função de bridge, dependendo do modelo adquirido. São necessárias três placas de rede, duas para o TAP e uma para análise ou transmissão dos alertas/logs;

A escolha da forma de coleta depende da necessidade do administrador. Se desejar apenas levantar uma estatística ou realizar alguns testes, a forma de espelhamento é mais fácil de implementar. Caso contrário, o modo bridge ou o modo TAP são mais indicados, principalmente para aqueles que desejam utilizar o método de prevenção dos pacotes suspeitos (*Intrusion Prevention System – IPS*).

## IDS e IPS

Há muita confusão quanto às diferenças entre IDS e IPS. O IDS age de forma passiva, apenas detectando anomalias e pacotes maliciosos, enquanto IPS age de forma ativa, atuando diretamente nas regras de firewall e/ou regras de outros serviços de segurança (TCPWrappers, por exemplo). Muitos entendem que IPS é separado do

IDS, quando na verdade é uma função extra que o IDS pode proporcionar na detecção do ataque. O Snort pode atuar das duas formas.

Existem aplicativos no mercado que possuem apenas a função de IPS, porém só são funcionais se analisam os alertas de aplicativos IDS. Um exemplo é o Guardian, que age de acordo com os alertas que o Snort gera.

O IPS é vantajoso quando utilizado na borda da rede corporativa, pois ajuda a diminuir ataques de categoria zero day (ver **Nota DevMan 3**), vírus, worms e outros, impedindo a ação do atacante e evitando que a rede seja comprometida. Em contrapartida não é recomendável utilizá-lo na rede privada da organização, pois caso ocorra um falso positivo, a conexão entre servidores e clientes confiáveis ficará comprometida.



## Nota do DevMan 3

**Zero day:** Ameaças que ainda não foram descobertas pelos desenvolvedores e/ou pelas empresas de segurança.

## Escolha do hardware

A escolha do hardware apropriado para suportar o Snort depende muito do fluxo da rede da organização. Quanto maior o fluxo de pacotes do segmento, mais potente deve ser o servidor. A seguir são analisadas algumas dicas para realizar a escolha mais adequada ao ambiente:

• **CPU:** os sensores de tráfego do Snort consomem bastante processamento. Portanto, quanto mais pré-processadores ativados, mais recursos da CPU serão uti-

lizados. O indicado, neste caso, é escolher processadores que contenham mais de um núcleo;

• **Memória RAM:** a quantidade de memória RAM é analisada de acordo com o desempenho do Snort. Quando todo o espaço disponível da memória RAM é ocupado, o Snort perde desempenho e pacotes são descartados, por não conseguir analisá-los a tempo. Ao adicionar uma quantidade maior de memória RAM pode-se eliminar o problema;

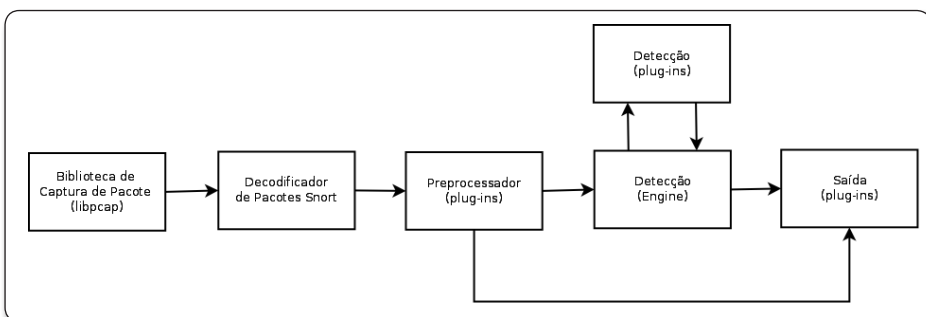
• **Disco Rígido:** a recomendação é utilizar discos rígidos eficientes. A eficiência de um disco rígido consiste na Rotação Por Minuto (RPM) do mesmo, na transferência de dados por segundo suportada e outros detalhes como buffer de disco. Atualmente, discos com a tecnologia SAS (*Serial Attached SCSI*) são os mais indicados, em se tratando de servidores. Discos SATA (*Serial ATA*) de 3 Gb/s de transferência são uma alternativa de baixo custo;

• **Placa de rede:** a taxa de transferência das placas de rede deve ser compatível com a taxa de transferência da rede da organização. Placas com taxa de transferência inferior a ativos da rede (switch, por exemplo) podem congestionar o tráfego, interferindo na otimização e desempenho da rede. A escolha de uma placa de rede própria para servidores é mais indicada, pois foram projetadas especialmente para aplicativos de alto desempenho que requerem um grande tráfego de dados (como o Snort), além de outros recursos como criptografia via hardware e SNMP.

## Localização dos servidores IDS

A **Figura 2** representa uma típica rede corporativa, somando os pontos estratégicos de cada IDS. Analisando-os, é necessário que cada um possua um comportamento diferente, se adequando a cada situação. Abaixo é explicada a situação de cada IDS:

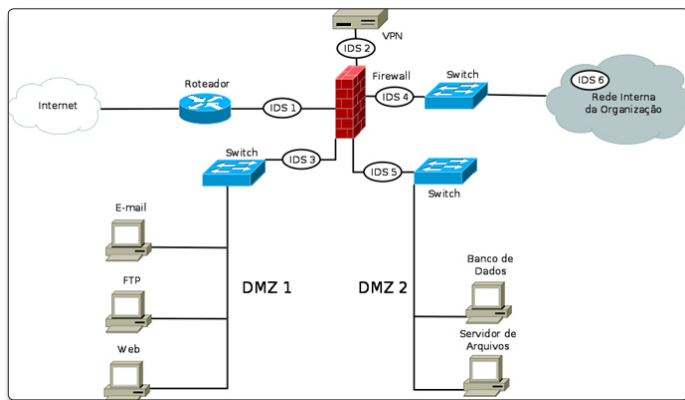
• **IDS 1:** conexão entre o roteador e o firewall, ou seja, o perímetro da rede. O reforço deve ser maior neste segmento, pois esta parte é a que se expõe à rede pública. O ideal é restringir ao máximo pacotes mal formados e ataques em camada de aplicação, como Buffer Overflow (ver



**Figura 1.** Componentes internos da ferramenta Snort

**Nota DevMan 4)**, malware e outros. A restrição é realizada com o Snort operando em modo IPS;

- **IDS 2:** conexão entre a rede privada da organização e parceiros/fornecedores/filiais (VPN). A união entre redes privadas via Internet requer um nível de segurança maior, podendo fornecer encapsulamento, criptografia e restrição de acesso, porém pode haver comportamento indevido de algum funcionário/parceiro insatisfeito com a sua organização. Deste modo, um IDS alertando pacotes que possam comprometer a rede pode ser o suficiente;
- **IDS 3 e IDS 5:** conexão entre o firewall e as DMZs. Ataques específicos a certos tipos de serviço neste segmento devem ser impedidos ou evitados. Estes IDSs devem agir como IPS em certas situações (de proteção aos serviços expostos à rede) e registrar os alertas restantes. Nota-se que já existe a proteção do IDS 1, então os IDSs 3 e 5 devem ser mais restritos em relação a certos tipos de ataque;
- **IDS 4:** conexão entre o firewall e o switch da rede privada. Neste segmento podem estar ligados diversos servidores privados (Intranet, por exemplo). Assim, é necessário adicionar um IDS para alertar sobre possíveis ataques internos. Sendo mais cauteloso, pode-se habilitar o modo IPS nas regras que comprometam a rede, como as que detectam o ARP Spoofing (**Nota DevMan 5**). Apesar dos switches gerenciáveis poderem impedir este tipo de ataque, nada melhor do que garantir a segurança com uma camada a mais;
- **IDS 5:** conexão entre o switch e as estações de trabalho. Ataques advindos da Internet neste segmento são raros, mas existem as ameaças internas, como ataques no mesmo segmento ou propagação de malwares. A decisão por implementar um IDS neste segmento vai depender do administrador e da situação da rede corporativa.



**Figura 2.** Simulação de uma rede corporativa para demonstrar os locais possíveis de instalação de um IDS

## Instalação

O Snort possui suporte a diversos sistemas operacionais (Windows, Linux, Mac OS, FreeBSD etc.), mas este artigo abordará apenas a instalação no Linux, pois há um número maior de documentação referente a esta plataforma.

Antes de começar a instalação do Snort, é necessário instalar as dependências a seguir:



## Nota do DevMan 4

**Buffer Overflow:** É um ataque que explora o mau funcionamento da aplicação ou protocolo no momento da escrita dos dados no buffer, permitindo a execução de código malicioso na memória adjacente.



## Nota do DevMan 5

**ARP Spoofing:** Técnica que o atacante utiliza para criar mensagens falsas do protocolo ARP dentro da rede local. O protocolo ARP não é orientado a conexão, permitindo que a associação entre um endereço MAC e um endereço IP seja armazenado na tabela ARP do equipamento, independente de quem enviou o pacote ARP. Esta vulnerabilidade permite a manipulação da tabela ARP de cada equipamento, possibilitando que o atacante finja ser qualquer equipamento pertencente à rede local.

- **Libpcap:** a Biblioteca de Captura de Pacotes (*Library Packet Capture*), como demonstrado anteriormente, faz a captura dos pacotes que adentram na placa de rede do servidor. Ela consta nos repositórios da maioria das distribuições. O download do código-fonte pode ser realizado em [2], caso seja necessário compilar manualmente;
- **PCRE (Perl-Compatible Regular Expressions):** as Expressões Regulares Compatíveis com Perl (em tradução livre) facilitam a consulta nos pacotes, otimizando a aplicação. Este recurso consta nos repositórios da maioria das distribuições. O download do código-fonte pode ser realizado em [3], caso seja necessário compilar manualmente;
- **Libdnet:** biblioteca que provê suporte aos protocolos das camadas 2 e 3 da pilha TCP/IP. Normalmente não consta nos repositórios das distribuições, portanto, é necessário fazer o download em [4] e compilá-lo com o comando `./configure && make && make install`;
- **DAQ (Data Acquisition):** biblioteca que substitui as chamadas diretas das funções do PCAP (ver **Nota DevMan 6**) e inclui uma camada de abstração que facilita a operação em diversas interfaces de hardware e software, permitindo que o Snort tenha compatibilidade a elas sem ser modificado. É necessário compilá-la, pois não consta nos repositórios das distribuições. Disponível em [3];



## Nota do DevMan 6

**PCAP:** Abreviação de Packet Capture, utiliza a biblioteca libpcap para capturar os pacotes e pode, através do plug-in de output, converter os dados no formato PCAP para leitura. Este plug-in de output é a opção padrão do Snort.

- **Barneyard:** interpretador para o Snort que consegue ler a saída do tipo unified2, que é um formato binário, colhido da forma pura, sem tratamento. Permite que o Snort escreva no disco de forma eficiente e traduz a saída em vários formatos para análise

posterior dos dados, em processos separados. Isto impedirá a perda de tráfego da rede. Não é obrigatório seu uso, porém pode melhorar o desempenho do Snort, principalmente se o tráfego de dados for muito grande.

Após as instalações das dependências, é hora de instalar o Snort. Disponível em [4], será instalada a versão em código-fonte ao invés de pacotes compilados, pois o procedimento é válido em qualquer distribuição Linux, bastando apenas instalar as dependências necessárias para compilação (de acordo com a distribuição escolhida). A versão utilizada é a 2.9.2.1. Assim, execute os seguintes comandos:

```
# tar xzvf snort-2.9.2.1.tar.gz
# cd snort-2.9.2.1
# ./configure
# make && make install
```

Um parâmetro interessante que pode ser inserido no momento da configuração da instalação do Snort (`./configure`) é o `--prefix=</dir>`. A função deste parâmetro é informar ao script de configuração o local de instalação do Snort, mas fica a critério do administrador do sistema modificar o caminho. Apesar desta opção, este parâmetro não será utilizado, sendo o Snort instalado no diretório padrão (`/usr/local`).

Concluída a instalação, é necessário instalar os plug-ins e as regras, contidas em [1], lançadas periodicamente pelo VRT. A versão do lançamento das regras sempre coincide com a versão do Snort. Portanto, se a versão utilizada do Snort é a 2.9.2.1, consequentemente a versão das regras a ser baixada é a 2921 (também disponível em [1]). A **Listagem 1** demonstra o procedimento de instalação das regras e plug-ins.

**Listagem 1.** Procedimento de instalação das regras e plug-ins, disponibilizados pela VRT.

```
# mkdir /etc/snort
# tar xzvf snortrules-snapshot-2921.tar.gz -C /etc/snort
# cd /etc/snort
# mv etc/* .
# rmdir etc
# ls -l
-rw-r--r--. 1 1210 1210 3621 Jan 12 15:17 classification.config
-rw-r--r--. 1 1210 1210 28624 Jan 12 15:17 gen-msg.map
-rw-r--r--. 1 1210 1210 1112 Jan 19 2010 open-test.conf
drwxr-xr-x. 2 1210 1210 4096 Feb 22 23:51 preproc_rules
-rw-r--r--. 1 1210 1210 666 Jan 12 15:17 reference.config
drwxr-xr-x. 2 1210 1210 4096 Feb 22 21:17 rules
-rw-r--r--. 1 1210 1210 2728300 Jan 12 15:19 sid-msg.map
-rw-r--r--. 1 1210 1210 24116 Jan 12 15:17 snort.conf
drwxr-xr-x. 4 1210 1210 4096 Feb 22 23:52 so_rules
-rw-r--r--. 1 1210 1210 2556 Jan 12 15:17 threshold.conf
-rw-r--r--. 1 1210 1210 53841 Jan 12 15:17 unicode.map
```

O conteúdo do diretório `/etc/snort` (também apresentado na **Listagem 1**) mostra diversos arquivos e subdiretórios. Cada um possui uma função que auxilia o Snort a classificar, priorizar, registrar e alertar contra diversos tipos de ameaças. O significado de cada arquivo e subdiretório está discriminado abaixo:

- **classification.config:** inclui informações para priorizar as regras, permitindo que os alertas sejam classificados e priorizados. É possível especificar que prioridade cada classificação possui;
- **gen-msg.map:** é usado para traduzir os IDs de todos os geradores de alertas no Snort, exceto os de *Deteção (Engine)*. São necessários para o Barnyard;
- **open-test.conf:** arquivo de exemplo para carregar regras padrão. Serve para testar a ferramenta de uma forma rápida;
- **preproc\_rules:** pasta que contém as regras de alertas relativas aos pré-processadores e decodificadores;
- **reference.config:** configurações de referências de sites de segurança. Todos os links que são adicionados neste arquivo são acrescidos aos alertas gerados, de acordo com cada regra de alerta;
- **rules:** pasta que contém as regras de alertas de saída;
- **sid-msg.map:** é usado para traduzir o ID encontrado em uma regra do Snort para uma variável. São necessários para o Barnyard;
- **snort.conf:** arquivo de configuração que carrega todas as regras, plug-ins, variáveis, base de detecção etc.;
- **so\_rules:** aparentemente, as regras que constam nesta pasta não se diferem muito das regras da pasta *rules*. A diferença está na forma em que são lançadas, pois são regras que possuem acordos com terceiros para não serem divulgadas. Por padrão, não são carregadas;
- **threshold.conf:** serve para suprimir os alertas que se repetem com frequência, evitando que os arquivos de log se encham rapidamente;
- **unicode.map:** padrões de codificação dos logs cadastrados.

## Configuração

A configuração do Snort pode ser realizada de duas formas: via parâmetros (ex.: `snort -A full -i eth0 --daq-mode pcap`) ou através do arquivo `snort.conf`. Neste arquivo é definido as regras, em quais IPs/redes o Snort deve observar, lista branca/negra, pré-processadores e outros detalhes. O arquivo original, obtido junto com as regras da equipe VRT, é totalmente intuitivo. Ele possui um roteiro de configuração que explica o que significa cada parâmetro, instruindo o administrador a configurá-lo da forma desejada.

Antes de configurar o Snort, é necessário ter em mente que ele possui três formas de funcionamento:

- **Modo sniffer:** apenas captura os pacotes no formato puro e mostra na tela. Seu comportamento é semelhante ao da ferramenta tcpdump. Exemplo de comando: `snort -vde`;
- **Modo registro de pacotes:** além de capturar os pacotes, possui a função de registrar todos os pacotes que entraram na interface de rede. Exemplo de comando: `snort -vde -l /var/log/snort`;
- **Modo detecção/prevenção de intrusos na rede:** modo que detecta e/ou previne intrusos na rede. É mais complexo, pois envolve várias etapas, conforme foi descrito na **Figura 1**. O `snort.conf` facilitará a configuração do sistema. Exemplo de comando: `snort -c /etc/snort/snort.conf`.

O arquivo *snort.conf* possui várias configurações, portanto, serão comentadas aqui apenas as principais opções que podem modificar o comportamento do Snort. A equipe VRT organizou os parâmetros do *snort.conf* em seções para facilitar a configuração e compreensão das opções. As seções estão definidas da seguinte forma:

- **Configurar as variáveis:** esta seção especifica as variáveis que serão utilizadas por todo o arquivo, inclusive pelas regras. Há variáveis de portas, serviços, rede etc.;

- **Configurar o decodificador:** são opções que alertam o sistema contra pacotes fora do padrão. O manual possui uma lista dos códigos de cada alerta que ele classifica;

- **Configurar a base de detecção:** define as opções gerais da base de detecção e os plug-ins disponíveis, como método de detecção, fila de eventos a serem registrados pelo mesmo pacote ou stream, ordem dos pacotes na fila e outras opções;

- **Configurar as bibliotecas dinâmicas a serem carregadas:** como o próprio nome diz, esta configuração informa ao Snort o caminho correto para as bibliotecas dinâmicas a serem carregadas;

- **Configurar os pré-processadores:** carrega os plug-ins de cada pré-processador e suas opções;

- **Configurar os plug-ins de saída:** configura a forma que o Snort irá reportar os alertas. O padrão é dentro de */var/log/snort*;

- **Personalizar a lista de regras de pré-processadores e decodificadores:** possibilita escolher quais regras devem constar no momento da análise dos pacotes. Também permite criar regras para detectar um comportamento suspeito, além de habilitar regras extras;

- **Personalizar sua lista de regras de objetos compartilhados (shared objects):** possibilita escolher quais regras devem constar na análise dos pacotes. Esta lista é diferente da anterior, pois a equipe VRT não pode divulgar vulnerabilidades descobertas sem autorização. Outra diferença está na criação das regras: a lista comum de regras é criada na linguagem própria do Snort, enquanto a linguagem desta lista é criada na linguagem C, para facilitar a distribuição para outras plataformas.

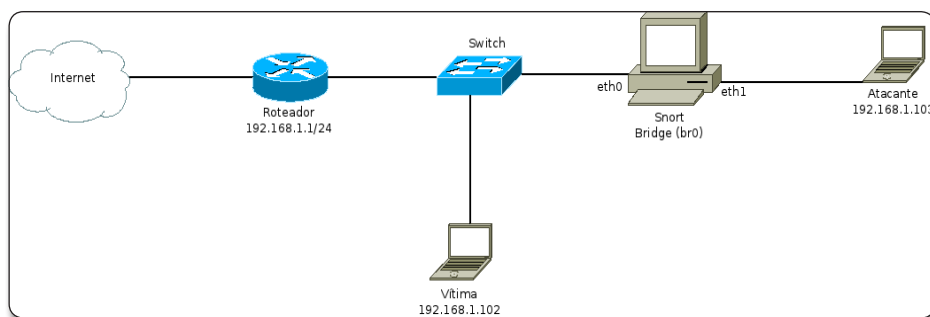


Figura 3. Ambiente de rede para realizar testes com o Snort

Os parâmetros a seguir se referem àqueles que foram modificados para os testes:

- **ipvar HOME\_NET 192.168.1.0/24** – rede (ou host) a ser protegida. O parâmetro *any* significa qualquer IP;

- **var RULE\_PATH rules** – informa a localização do diretório das regras gerais. Neste caso, no mesmo diretório do arquivo de configuração;

- **var SO\_RULE\_PATH so\_rules** – informa a localização do diretório das regras de objetos compartilhados;

- **var PREPROC\_RULE\_PATH preproc\_rules** – informa a localização do diretório das regras de pré-processamento;

- **var WHITE\_LIST\_PATH rules** – informa a localização da “lista branca” do Snort. É necessário criar um arquivo chamado *white\_list.rules*;

- **var BLACK\_LIST\_PATH rules** – informa a localização da “lista negra” do Snort. É necessário criar um arquivo chamado *black\_list.rules*;

- **config daq: nfq** – determina o módulo DAQ para a coleta dos pacotes. O módulo NFQ (*Netfilter Queue*) possui um novo método de manipulação dos pacotes em conjunto com o iptables e vai ser utilizado nos testes;

- **config daq\_dir** – */usr/local/lib/daq* – indica o caminho para os módulos DAQ;

- **include \$PREPROC\_RULE\_PATH/preprocessor.rules** – inclui as regras dos pré-processadores necessárias para o teste;

- **include \$PREPROC\_RULE\_PATH/decoder.rules** – inclui as regras dos decodificadores necessárias para o teste.

Não há a necessidade de alterar os outros parâmetros, pois serão suficientes para rodar o Snort sem falhas.

O teste a seguir será baseado em bloqueio de varreduras de portas, com o Snort em modo IPS (no modo inline) através do módulo DAQ chamado NFQ. Este módulo controla os pacotes que estão na fila das regras do iptables, decidindo o destino de cada pacote analisado. Quando a varredura do aplicativo malicioso é realizada, o Snort irá detectar esses pacotes maliciosos e não permitirá a passagem destes, impedindo o atacante de obter informações.

A Figura 3 demonstra um exemplo de como o Snort pode se estabelecer na rede. Neste caso, o Snort está em modo bridge (ponte), ou seja, todos os pacotes que adentram na interface de rede eth0 passam pela interface eth1 (e vice-versa) de forma transparente, agindo como um switch. A conexão de uma bridge depende do pacote *bridge-utils*, que utiliza o comando *brctl* para criar a conexão. A Listagem 2 mostra como criar uma bridge e adicionar as duas interfaces de rede.

Nota-se que as interfaces eth0 e eth1 não podem, em hipótese alguma, obter endereço de rede. Os comandos *ifconfig eth0 0* e *ifconfig eth1 0* certificam que estas interfaces não possam ter endereço de rede configurado.

Os pacotes devem estar em fila (queue) para a análise do módulo DAQ NFQ, sendo assim, a Listagem 3 os põe nestas condições.

Depois de tudo configurado, o Snort pode ser iniciado com o comando *snort -c /etc/snort/snort.com -Q*, onde o parâmetro *-Q* informa para o Snort atuar em modo inline (para permitir o modo IPS). A Listagem 4 mostra o comando executado e sua saída.

## Listagem 2. Criação da bridge e adição das duas interfaces de rede.

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
ifconfig eth0 0
ifconfig eth1 0
ifconfig br0 up
```

## Listagem 3. Regras de iptables para colocar os pacotes em fila.

```
iptables -A INPUT -j QUEUE
iptables -A OUTPUT -j QUEUE
iptables -A FORWARD -j QUEUE
```

## Listagem 4. Execução do Snort, incluindo sua saída.

```
snort -c /etc/snort/snort.conf -Q
Enabling inline operation
Running in IDS mode
--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 591 593 901 1220 1414 1830 2301 2381
2809 3128 3702 4343 5250 7001 7145 7510 7777 7779 8000 8008 8014 8028 8080
8088 8118 8123 8180:8181 8243 8280 8800 8888 8899 9080 9090:9091 9443 9999
11371 55555 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]
(...)
+++++
Initializing rule chains...
2873 Snort rules read
2478 detection rules
142 decoder rules
253 preprocessor rules
2873 Option Chains linked into 171 Chain Headers
0 Dynamic rules
+++++
[ Number of patterns truncated to 20 bytes: 511 ]
nfq DAQ configured to inline.
The DAQ version does not support reload.
Reload thread starting...
Reload thread started, thread 0x7f8c45e03700 (7492)
--== Initialization Complete ==--
"" -> Snort! <*-
o" ~ Version 2.9.2.1 IPv6 GRE (Build 107)
"" By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
Copyright (C) 1998-2012 Sourcefire, Inc., et al.
Using libpcap version 1.1.1
Using PCRE version: 8.12 2011-01-15
Using ZLIB version: 1.2.5
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.15 <Build 18>
Preprocessor Object: SF_SDF (IPv6) Version 1.1 <Build 1>
Preprocessor Object: SF_SSLPP (IPv6) Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 (IPv6) Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS (IPv6) Version 1.1 <Build 1>
Preprocessor Object: SF_DNS (IPv6) Version 1.1 <Build 4>
Preprocessor Object: SF_DCERPC2 (IPv6) Version 1.0 <Build 3>
Preprocessor Object: SF_REPUTATION (IPv6) Version 1.1 <Build 1>
Preprocessor Object: SF_POP (IPv6) Version 1.0 <Build 1>
Preprocessor Object: SF_SIP (IPv6) Version 1.1 <Build 1>
Preprocessor Object: SF_GTP (IPv6) Version 1.1 <Build 1>
Preprocessor Object: SF_SSH (IPv6) Version 1.1 <Build 3>
Preprocessor Object: SF_FTPTELNET (IPv6) Version 1.2 <Build 13>
Preprocessor Object: SF_SMTP (IPv6) Version 1.1 <Build 9>
Preprocessor Object: SF_IMAP (IPv6) Version 1.0 <Build 1>
Commencing packet processing (pid=7492)
Decoding Raw IP4
```

As informações geradas informam, passo a passo, as configurações e regras carregadas. O destaque em negrito é importante para sabermos se o Snort e o módulo DAQ NFQ estão sendo executados em modo inline.

A simulação de varredura de portas será realizada com o aplicativo chamado hping3, que possui a capacidade de gerar diversos tipos de pacotes da pilha TCP/IP personalizados. Neste caso, a técnica utilizada é a varredura de TCP com a flag SYN ativada e com o payload de zero byte, forçando a vítima a responder a requisição como se fosse uma solicitação de conexão TCP válida.

Supondo que o Snort não esteja em execução, a vítima irá responder às requisições feitas pelo hping3. Veja o comportamento deste aplicativo na **Listagem 5**.

Com o Snort em execução, a saída do comando hping3 será diferente, conforme a **Listagem 6**.

Analisando o log de alertas do Snort em tempo real (**tail -f /var/log/snort/alert**), surgem os registros apresentados na **Listagem 7**.

## Listagem 5. Comportamento do aplicativo hping3 quando o Snort não está sendo executado.

```
hping3 --syn 192.168.1.102
HPING 192.168.1.102 (eth0 192.168.1.102): S set. 40 headers + 0 data bytes
len=46 ip=192.168.1.102 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=5.8 ms
len=46 ip=192.168.1.102 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=7.4 ms
len=46 ip=192.168.1.102 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=20.4 ms
(...)
```

## Listagem 6. Comportamento do aplicativo hping3 quando o Snort está sendo executado.

```
hping3 --syn 192.168.1.102
HPING 192.168.1.102 (eth0 192.168.1.102): S set. 40 headers + 0 data bytes
```

## Listagem 7. Log de alertas do Snort no momento do bloqueio do tráfego.

```
[**] [129:1:1] Syn on established session [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/25-12:22:59.297441 192.168.1.103:2884 -> 192.168.1.102:0
TCP TTL:64 TOS:0x0 ID:21260 IpLen:20 DgmLen:40
*****S* Seq: 0x1867B5E Ack: 0x5C027F24 Win: 0x200 TcpLen: 80

[**] [129:1:1] Syn on established session [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/25-12:23:00.289089 192.168.1.103:2885 -> 192.168.1.102:0
TCP TTL:64 TOS:0x0 ID:3267 IpLen:20 DgmLen:40
*****S* Seq: 0x31A38B47 Ack: 0x2159E6A5 Win: 0x200 TcpLen: 20

[**] [129:1:1] Syn on established session [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/25-12:23:01.289362 192.168.1.103:2886 -> 192.168.1.102:0
TCP TTL:64 TOS:0x0 ID:29667 IpLen:20 DgmLen:40
*****S* Seq: 0x1BC7846F Ack: 0x5506B5A3 Win: 0x200 TcpLen: 20

[**] [129:1:1] Syn on established session [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/25-12:23:02.289530 192.168.1.103:2887 -> 192.168.1.102:0
TCP TTL:64 TOS:0x0 ID:65354 IpLen:20 DgmLen:40
*****S* Seq: 0x2B1C05AC Ack: 0x3DAAFC2 Win: 0x200 TcpLen: 20
(...)
```

Analisando a saída do alarme, a primeira linha informa o tipo de ataque e um campo (ID) com três números, separados por dois pontos. O primeiro ID informa qual componente do Snort gerou o alerta. Consultando o guia *gen-msg.map* (dentro de */etc/snort*), é informado que o decodificador que gerou o alerta foi o stream5. O segundo ID é a identificação do alerta gerado de acordo com a regra criada. O terceiro ID, por fim, indica a revisão do alerta, sendo que neste exemplo não sofreu nenhuma alteração (o que é explicitado pelo número 1).

A segunda linha faz referência à classificação que o alerta recebeu e sua prioridade. Quanto menor o número, mais crítico é o alerta.

Da terceira linha em diante são apresentadas informações do pacote capturado (TTL, flag TCP, sequência, tamanho do cabeçalho, origem e destino e outras características do cabeçalho).

Concluindo o exemplo, percebe-se que o Snort conseguiu bloquear este tipo de ataque, informando nos registros de alerta qual decodificador agiu contra o ataque, dentre outras informações. Vale lembrar que os decodificadores atuam em um nível mais baixo (desde a camada de enlace até a camada de transporte da pilha TCP/IP) para detectar o ataque, porém há pré-processadores que auxiliam na busca de ataques na camada de aplicação.

## Ajustes finos

O pleno funcionamento do Snort depende muito de um ajuste fino na ferramenta. A seguir, listamos algumas dicas para evitar falsos positivos (alarmes falsos), falsos negativos (alarmes que são ignorados, mas são legítimos), melhorias de desempenho e algumas dicas de segurança:

- **Inclusão/exclusão de regras:** dependendo do local físico onde o servidor IDS é instalado, certas regras não têm necessidade de serem inseridas. Ex.: se no segmento em que o Snort se encontra não há servidor web, não é preciso carregar regras para proteger servidores web. Isto diminuiria a carga da ferramenta Snort e eliminaria falsos positivos;

- **Testes de bloqueio:** a eficiência da ferramenta depende de testes no segmento em que o servidor IDS está localizado. A análise dos alertas em um determinado período vai informar ao administrador a rotina dos pacotes que adentram neste segmento. Em modo IPS, esta análise é de suma importância para bloquear os pacotes específicos. Para simular o bloqueio de pacotes pode-se utilizar o parâmetro `--enable-inline-test` ou inserir a linha `config policy mode:inline_test` no arquivo *snort.conf*;

- **Definindo e configurando pré-processadores e decodificadores:** de acordo com a análise da rotina dos pacotes que trafegam no segmento, é possível perceber que nem todos os pré-processadores e decodificadores serão necessários. O ajuste das opções dos pré-processadores e decodificadores é importante, pois cada um possui uma particularidade e fornece diversas opções de verificação. O manual de cada pré-processador/decodificador encontra-se no código-fonte do Snort, no diretório *doc*;

- **Escolha do hardware:** assunto abordado anteriormente, a escolha do hardware é importante para o desempenho da ferramenta

Snort. Se o servidor não conseguir suportar a exigência da ferramenta, a análise dos pacotes pode ficar comprometida, havendo a perda de pacotes;

- **Faça suas próprias regras:** a ferramenta Snort permite a criação de suas próprias regras. Em seu manual é explicado com detalhes como criar um alerta personalizado, seja para uma nova descoberta de vulnerabilidade, seja para se adequar à situação da rede organizacional;

- **Criptografia do tráfego de alertas:** os logs armazenados, por padrão, ficam no formato PCAP. Apesar de ser ilegível utilizando um editor de texto comum, aplicativos que conseguem ler arquivos neste formato (tcpdump e Wireshark, por exemplo) podem ler os logs do Snort. Se os logs são registrados em outro servidor, estes dados devem ser criptografados, evitando interceptação. A criação de túneis criptografados é a solução para suprir a falta de segurança. VPNs (IPSec, OpenVPN, L2TP) ou aplicativos mais simples, como o stunnel (via SSH), podem criptografar os dados trafegados;

- **Não execute o Snort como root:** a execução de serviços com privilégios de super usuário não é recomendada. Ataques de Buffer Overflow podem usurpar a aplicação e obter os mesmos direitos de execução, podendo ter total acesso ao servidor. O recomendado é criar um novo usuário e um novo grupo com privilégios restritos. Obs.: alguns módulos DAQ só aceitam serem executados com privilégios de super usuário;

- **Utilize o Barnyard:** à medida que o fluxo de pacotes aumenta o Snort não consegue mais tratá-los, tornando-o mais lento e ocasionando o descarte de novos pacotes. O modo output chamado unified2 permite um número maior de pacotes por segundo a ser analisado, pois guarda estes valores em binário e não em modo texto. O Barnyard lê estes logs gerados e converte-os para o formato desejado (PCAP, syslog etc.).

## Aplicativos e dicas que facilitam o gerenciamento do Snort

O gerenciamento e monitoramento da ferramenta Snort não é uma tarefa simples, mas existem ferramentas que podem facilitar o trabalho do administrador de redes. Aqui serão mencionadas as mais utilizadas apenas.

- **Oinkmaster:** é uma ferramenta escrita em Perl que atualiza automaticamente as regras através da base de dados da equipe VRT, substituindo as regras existentes. Esta ferramenta manterá sempre o seu Snort apto a detectar as últimas ameaças descobertas pela equipe VRT. O ponto fraco está no método de atualização, que substitui as regras anteriores pelas novas adquiridas pelo aplicativo. Uma forma de evitar isto é separar as regras modificadas/criadas em outro diretório e adicioná-las no *snort.conf*. Assim, é aconselhável realizar uma cópia de segurança do arquivo *snort.conf* para o caso de falha na execução do Snort. A ferramenta Oinkmaster está disponível em [5];

- **BASE (Basic Analysis and Security Engine):** esta ferramenta, em conjunto com diversas dependências (MySQL, PHP, Apache, ADODB), facilita a visualização dos alertas gerados pelo Snort, através de uma interface web. O Snort registra seus alertas no ban-

co de dados do MySQL e o BASE realiza a consulta, informando na interface web os tipos de protocolos capturados, com detalhes de cada um. Os logs podem ser registrados no banco através da opção **output database: alert, <tipo\_banco>, dbname=<banco>, user=<usuário>, pass=<senha>**. Observação: a ferramenta em questão é baseada no antigo ACID (Analysis Console for Intrusion Database), que está descontinuado. Disponível em [6];

- **SnortAlog:** semelhante ao BASE, porém mais enxuto. Tem a capacidade de ler todos os tipos de outputs disponíveis no Snort e gera gráficos em formatos JPEG, PNG e GIF. Disponível em [7];

- **SnortSam:** aplicativo que substitui o método tradicional de IPS do Snort, bloqueando o tráfego manipulando as regras do firewall do servidor. Sua estrutura é separada em duas partes: a primeira fornece uma nova opção de output para o Snort, chamada **alert\_fwscan**; e a segunda é o agente instalado no servidor, que manipula o firewall do sistema (o agente funciona em Linux, Windows e até em firewall Cisco). O ponto fraco da ferramenta é a configuração das regras, porque é necessário acrescentar o parâmetro **fwscan** em cada regra desejada. Disponível em [8];

- **Guardian:** substitui o método tradicional de IPS do Snort, semelhante ao SnortSam, porém não utiliza de agentes nem output específico. O bloqueio do tráfego é realizado, também, manipulando as regras do firewall. O Guardian é mais simples de configurar, porém suas opções são limitadas. Disponível em [9];

- **IDScenter:** um front-end gráfico que gerencia alertas, regras e opções do Snort. Bastante intuitivo, adapta aqueles administradores que não estão acostumados com a famosa tela preta. Funciona apenas em Windows. Disponível em [10].

## Conclusão

O Snort é o IDS open source mais utilizado no mundo, por sua robustez e precisão na geração de alertas de pacotes suspeitos. Sua estrutura descentralizada favorece o desempenho e escalabilidade, podendo adicionar novos recursos sem modificar seu núcleo. A equipe VRT mantém as regras atualizadas constantemente, melhorando sempre a precisão do Snort em encontrar pacotes maliciosos, e a possibilidade de criar regras personalizadas amplia a detecção de novos ataques, permitindo a colaboração para a comunidade Snort.

O gerenciamento e o monitoramento são complexos, o que é uma desvantagem da ferramenta. Porém, existem diversos aplicativos que auxiliam na administração do IDS, além de acrescentar funcionalidades extras ao Snort.

Além das opções demonstradas neste artigo, existem muitas outras que permitem ao Snort se adaptar ao ambiente desejado, exigindo do administrador um alto conhecimento sobre elas. Por isso, é sempre recomendado contar com a ajuda do manual disponível no código-fonte para utilizar as melhores opções.

Mesmo assim, a eficácia do Snort na rede organizacional depende de um estudo para saber quais são os pontos mais vulneráveis a ataques, com análises de tráfego de cada segmento em um determinado período. Esta análise é importante para determinar quais regras serão utilizadas no Snort, bem como os pré-processadores/decodificadores ideais.

Por fim, vale ressaltar que configurar o Snort é um processo árduo e demorado. Depende de diversos ajustes finos, análises de desempenho, testes, redução de falsos positivos e negativos. No entanto, quando se chega a um ponto estável, faz todo o esforço valer à pena. Afinal, uma camada de segurança a mais é sempre bem-vinda.



**Renato Carneiro Pacheco**

*renatocarneirop@gmail.com*

Atua como Analista de Redes e Segurança há 2 anos, trabalhando com sistemas Linux, Windows e FreeBSD. Atua, também, como professor em instituição de graduação e pós-graduação. Pós-Graduado em Segurança de Redes e bacharel em Tecnologia de Redes de Comunicação. Possui certificado LPIC-1. Palestrante em eventos de Goiânia como FLISOL e FGSL.



## Download das regras do Snort [1]

<http://www.snort.org/snort-rules/>

## Site oficial do tcpdump (sniffer) [2]

<http://www.tcpdump.org>

## Download do Snort e dependências [3]

<http://www.snort.org/snort-downloads>

## Site oficial do Libdnet [4]

<http://libdnet.sourceforge.net/>

## Site oficial do Oinkmaster [5]

<http://oinkmaster.sourceforge.net/>

## Site oficial da suíte BASE [6]

<http://base.secureideas.net>

## Site oficial do SnortAlog [7]

<http://jeremy.chartier.free.fr/snortalog/>

## Site oficial do SnortSam [8]

<http://www.snortsam.net>

## Site oficial do Guardian [9]

<http://www.chaotic.org/guardian/>

## Site oficial do IDScenter [10]

<http://www.engagesecurity.com/products/idscenter/>

## Manual do Snort

[http://www.snort.org/assets/166/snort\\_manual.pdf](http://www.snort.org/assets/166/snort_manual.pdf)

## Passo-a-passo para instalar o Snort com o BASE no Fedora

<http://www.rootninja.com/snort-ids-basic-analysis-security-engine-base-fedora/>

## Livros

**Snort for Dummies** por Charlie Scott, Paul Wolfe e Bert Hayes

## Normas

**Tecnologia da Informação – Técnicas de Segurança – Código de Prática para a gestão de segurança da informação pela ABNT NBR ISO/IEC 27002:2005**



# Salve seus dados com Bacula – Parte 2

## Configuração e instalação



ESTE ARTIGO FAZ PARTE DE UM CURSO

No Brasil, o Bacula tem despertado o interesse de diversos profissionais para seu uso. Estamos na era conhecida como a “era da informação”. Mas qual a preocupação que se deve ter com a informação? O que é importante, quanto tempo deve ser armazenado e o que pode ser descartado? Será que dependemos de todas e quaisquer tipos de informação? Nos dias atuais, o conhecimento passou a ser chave-mestra, no qual a informação é utilizada para gerir estratégias do negócio, seja para uma pequena empresa até multinacionais preocupadas em equilibrar seus gastos.

Um importante diferencial ligado ao crescimento e a competitividade está relacionado diretamente às informações contidas na organização como, segredos de negócios, históricos operacionais, pesquisas e análise de mercado.

A segurança da informação está relacionada à garantia de que as informações, em qualquer formato (seja mídias eletrônicas, papel, conversações pessoais) estejam protegidas contra o acesso por pessoas não autorizadas, bem como estejam sempre disponíveis quando necessárias, e que sejam confiáveis.

À medida que o Bacula mostra-se um sistema de gestão de backup seguro, livre e econômico, de fácil aplicabilidade nas empresas de pequeno, médio e grande porte, é portátil para os principais sistemas operacionais, e possui interface web para fácil administração, ele passa a ser um forte candidato ao escolhermos uma ferramenta para automação do processo de backup.

Vale ressaltar que o Bacula encontra-se em constante desenvolvimento a cerca de suas versões e atualizações, daí a importância de estudos aprofundados deste software.

### Resumo DevMan

#### *De que se trata o artigo:*

Este artigo tem como objetivo dar sequência ao estudo sobre a ferramenta Bacula, demonstrando suas formas de instalação e configuração. Também serão apresentados os modos de instalação e configuração com o MySQL e apresentadas as variantes para instalação do PostgreSQL e SQLite como bases de dados para gerenciar as rotinas de backup.

#### *Em que situação o tema é útil:*

O Bacula é uma solução de backup multiplataforma desenvolvida sob a licença GPL. É robusta, cheia de recursos e modular. O Bacula se adapta às redes de qualquer tamanho e em qualquer topologia. Com essa solução não há mais a necessidade de altos investimentos com licenças de software para backup.

#### *Salve seus dados com Bacula – Parte 2:*

Este artigo irá contemplar as fases de implantação e configuração do Bacula, extraindo a essência dos conceitos e sua aplicabilidade corporativa. Não foi possível demonstrar neste artigo todas as possibilidades de implementações do software devido a sua extensa variedade de configurações. Contudo, todas as definições aplicadas no artigo são portáteis a outros clientes de backup.

O primeiro passo a ser executado quando se planeja uma instalação do Bacula é selecionar e instalar uma base de dados, pois conforme mencionado na primeira parte do artigo, o Bacula usa uma base de dados para armazenar autorizações, índices e atributos que ficam gravados no *Catalog SQL*. Na versão 5 do Bacula são suportados os seguintes SGBDs: MySQL, PostgreSQL e SQLite.

Cada um destes possui características e limitações distintas, que muito provavelmente não serão perceptíveis ao usuário do Bacula. Os procedimentos de instalação também variam de acordo com a escolha do banco a ser utilizado. Existem diversos manuais na Internet, inclusive em Português, com todos os passos a serem seguidos para a instalação dos bancos. O caminho mais rápido para instalar o banco de dados consiste na utilização de um gerenciador de pacotes (programas que auxiliam a instalação

de aplicativos) como o *apt*, *aptitude* em distribuições baseadas em Debian (ler **Nota DevMan 1**), *yum* naquelas baseadas em RedHat e *yast* nas baseadas em Suse.

Dito isto, vamos iniciar agora a instalação e configuração do Bacula.



## Nota do DevMan 1

**Aptitude:** Aptitude é um front-end para o Advanced Packaging Tool (APT). Ele exibe uma lista de pacotes de software e permite ao usuário escolher interativamente pacotes para instalar ou remover.

### Instalação da base de dados

Nesse artigo será utilizado como exemplo de configuração o MySQL. Para a instalação do PostgreSQL ou SQLite basta alterar o nome do pacote a ser instalado (com o aplicativo *apt*, pode-se instalar o SGBD de forma fácil). Observe na **Listagem 1** as formas de instalação de cada um dos bancos em uma distribuição baseada em GNU/Linux Debian.

Depois de terminada a instalação, verifique se o serviço de banco de dados está rodando de forma correta. Pode-se confirmar sua execução utilizando o comando *ps*, conforme a **Listagem 2** (no exemplo desta listagem, considerando o MySQL).

#### Listagem 1. Formas de instalação do SGBD no Debian.

Para instalação do MySQL use:  
`# apt-get install mysql-server`

Para instalação do PostgreSQL use:  
`# apt-get install postgresql`

Para instalação do SQLite use:  
`# apt-get install sqlite`

#### Listagem 2. Verificando o status do serviço.

`# ps aux | grep mysql`

### Instalando o Bacula em um Servidor GNU/Linux

Partiremos agora para a instalação do Bacula em um servidor GNU/Linux. De agora em diante serão apresentadas com maiores detalhes duas formas básicas de instalação dos pacotes do Bacula: utilizando um gerenciador de pacotes (*apt-get*, *aptitude*, *yum*) e por compilação (via pacotes binários).

Mas antes de iniciar a instalação do servidor, inicialmente execute o comando (`#apt-cache search bacula`) para obter a lista de pacotes disponíveis. Observe na **Listagem 3** os pacotes disponibilizados em um sistema GNU/Linux Debian. Como podemos notar, a saída do comando mostra o pacote disponível e uma breve descrição de sua funcionalidade. Por exemplo, da lista de pacotes disponíveis, o pacote *bacula-console-qt* se refere a uma interface gráfica que traz uma grande facilidade para a administração do Bacula, como acompanhamento de agendamentos.

No site oficial do Bacula, [www.bacula.org](http://www.bacula.org), é possível encontrar os pacotes compilados para as mais diferentes distribuições GNU/Linux. Note que, neste caso, os pacotes já estão traduzidos para o idioma Português. Então, será necessário saber quais pacotes devem ser instalados, de acordo com a finalidade da instalação, se cliente ou servidor.

Bem, conhecidos os pacotes disponíveis, partiremos agora para a instalação dos pacotes propriamente ditos.

### O servidor de backup

Se ao iniciar a instalação do servidor Bacula, apenas for selecionado o pacote Bacula conforme o comando (`#apt-get install bacula`), o gerenciador de pacotes (*apt-get*) irá automaticamente escolher uma série de pacotes que possibilitam a instalação completa do serviço de backup. Nesta lista, um pacote referente ao banco de dados também será selecionado. Na **Listagem 4** pode-se observar os pacotes extras que o *apt-get* irá instalar.

#### Listagem 3. Pacotes disponibilizados nas distribuições baseadas no Debian.

*bacula-director-common* - verificação, recuperação e "backup" via rede - arquivos comuns do Director  
*bacula-server* - verificação, recuperação e "backup" via rede - meta pacote de servidor  
*bacula-console* - verificação, recuperação e "backup" de rede - console de texto  
*bacula-sd-pgsql* - verificação, recuperação e "backup" via rede - ferramentas PostgreSQL SD  
*bacula-traymonitor* - verificação, recuperação e "backup" de rede monitor de bandeja  
*bacula-director-mysql* - verificação, recuperação e "backup" via rede - armazenamento MySQL p/ o Director  
*bacula-sd-mysql* - verificação, recuperação e "backup" via rede - ferramentas MySQL SD  
*bacula-director-pgsql* - verificação, recuperação e "backup" via rede - armazenamento PostgreSQL p/ o Director  
*bacula-sd-sqlite3* - verificação, recuperação e "backup" via rede - ferramentas SQLite 3 SD  
*bacula-console-wx* - verificação, recuperação e "backup" de rede - console WxWindows  
*bacula-director-sqlite3* - verificação, recuperação e "backup" via rede - armazenamento SQLite 3 p/ o Director  
*bacula* - verificação, recuperação e "backup" via rede - meta pacote  
*bacula-sd* - verificação, recuperação e "backup" de rede - daemon de armazenamento  
*bacula-doc* - documentação para o Bacula  
*bacula-director-sqlite* - verificação, recuperação e "backup" via rede - armazenamento SQLite p/ o Director  
*bacula-console-qt* - Ferramenta de Administração Bacula  
*bacula-sd-sqlite* - verificação, recuperação e "backup" via rede - ferramentas SQLite SD  
*bacula-common* - verificação, recuperação e "backup" via rede - arquivos de suporte comum  
*bacula-fd* - verificação, recuperação e "backup" de rede - daemon de arquivo  
*bacula-client* - verificação, recuperação e "backup" via rede - meta pacote cliente

#### Listagem 4. Pacotes extras a serem instalados.

*bacula-client* *bacula-common* *bacula-console* *bacula-director-common* *bacula-director-sqlite3* *bacula-fd* *bacula-sd* *bacula-sd-mysql* *bacula-server* *bacula-traymonitor* *mt-st* *mtx* *sqlite3*

Pode-se observar que nesse exemplo está sendo utilizado o SQLite. Este pacote deverá ser alterado caso o administrador de-seje utilizar outro banco como o MySQL ou o PostgreSQL. Assim,

para instalar um servidor Bacula com o MySQL, basta substituir os pacotes que contêm a expressão `sqlite` por `mysql`. A mesma orientação pode ser seguida para uso com o `postgresql`.

Neste momento, vale uma atenção especial aos pacotes *mtx* e *mailx*. Estes trazem funcionalidades importantes para a administração e manipulação de fitas magnéticas e mensagens internas do serviço.

A **Tabela 1** apresenta os pacotes mínimos necessários para instalação e configuração de um servidor Bacula.

| Pacote                          | Função                                                                                                   |
|---------------------------------|----------------------------------------------------------------------------------------------------------|
| bacula-fd                       | Para fazer backup do próprio servidor de backup                                                          |
| bacula-common                   | Arquivos comuns do Bacula                                                                                |
| bacula-console                  | Caso deseje administrar o Bacula através do próprio servidor                                             |
| bacula-director-[nome_do_banco] | O director é o servidor Bacula propriamente dito                                                         |
| bacula-sd-[nome_do_banco]       | storage daemon – ou seja, manipula os dispositivos de armazenamento. O [nome_do_banco] pode ser opcional |
| bacula-traymonitor              | Apenas se o servidor possuir interface gráfica, e você desejar usá-la para administrar o Bacula.         |
| bacula-doc                      | Documentação do Bacula                                                                                   |

**Tabela 1.** Pacotes mínimos a serem instalados

Nas versões 3.x do Bacula, existe mais uma dependência: o pacote *libssl*, utilizado para criptografia. Durante a instalação, o *apt* irá perguntar se foi configurada alguma senha no banco de dados. Caso positivo, tal senha deverá ser informada.

Após a instalação, já se pode iniciar os *daemons* do Bacula (**Nota DevMan 2**), sempre seguindo a ordem indicada na **Listagem 5**.



## Nota do DevMan 2

**Daemon:** Daemons são processos que rodam em segundo plano aguardando por requisições.

**Listagem 5.** Ordem de inicialização dos daemons.

```
/etc/init.d/bacula-fd
/etc/init.d/bacula-sd
/etc/init.d/bacula-director
```

Qualquer erro que venha a ocorrer nas configurações dos arquivos (localizados em */etc/bacula*) será imediatamente demonstrado. Caso nenhum erro seja reportado, a instalação estará completa.

Para ter acesso ao *director*, que é o serviço responsável pela administração de todos os processos de *backup*, digite *bconsole*. Entretanto, caso o *director* não retorne nenhum erro, mas tenha sua execução terminada, provavelmente poderemos ter um problema relacionado à comunicação com o banco de dados. Caso isso venha a ocorrer, execute os scripts da **Tabela 2**.

| Script                              | Descrição                                                                |
|-------------------------------------|--------------------------------------------------------------------------|
| /etc/bacula/create_bacula_database  | Cria o banco de dados Bacula – no caso do MySQL, há um script específico |
| /etc/bacula/make_bacula_tables      | Cria as tabelas do banco                                                 |
| /etc/bacula/grant_bacula_privileges | Cria o usuário bacula e garante os privilégios no banco de mesmo nome.   |

**Tabela 2.** Scripts disponíveis no Bacula

Caso o erro persista, deve-se verificar a configuração de usuário e a senha do banco. Para isso, verifique o arquivo (*/etc/bacula/bacula-dir.conf*). Se mesmo assim o *daemon* do *bacula-director* vier a apresentar problemas, consulte os arquivos de *log* do Bacula para mais detalhes sobre o erro. Geralmente, quando a instalação é feita por compilação, essa irá apresentar melhores resultados quanto à ocorrência destes incidentes.

## Cliente de backup

Obviamente, se a máquina for apenas um cliente de *backup* Bacula, não será necessário instalar um banco de dados para o catálogo. A instalação do cliente pode ser feita através do pacote de nome *bacula-client*. Em algumas distribuições o pacote será *bacula-fd*.

Nessa parte do artigo será apresentado como efetuar a instalação por meio de compilação, sendo que o exemplo apresentado contemplará o servidor de *backup* (inclui o cliente e o *storage daemons*). A sequência exibida a seguir demonstra o processo de instalação:

- 1) Efetue o download do pacote no endereço: *www.bacula.com.br*;
- 2) Descompacte o *.tar.gz* (*tar -xzf bacula.tar.gz -C /opt*). A opção “*xzf*” descompacta arquivos antes compactados com o *gzip*, e o parâmetro “*-C*” indica o diretório onde o mesmo será descompactado; Após descompactar, acesse o diretório */opt*. Antes de compilar, instale os compiladores e demais pacotes necessários. Para isso, em distribuições baseadas em GNU/Linux Debian, execute *#apt-get install build-essential*;
- 3) Se estiver instalando a versão 3 ou superior do Bacula, instale o pacote *libssl-dev* (*#apt-get install libssl-dev*) a fim de tratar questões de criptografia;
- 4) Execute o comando *./configure* a fim de checar possíveis dependências ainda não satisfeitas. Observe as possíveis opções:

```
./configure
--sbindir=$HOME/bacula/bin
--sysconfdir=$HOME/bacula/bin
--with-pid-dir=$HOME/bacula/bin/working
--with-sysdir=$HOME/bacula/bin/working
--enable-smartalloc
--with-mysql
--with-working-dir=$HOME/bacula/bin/working
--with-dump-email=reis.falexandre@gmail.com
--with-job-email=reis.falexandre@gmail.com
--with-smtp-host=localhost
```

- 5) Caso seja utilizado o MySQL, há uma dependência específica, portanto deve-se instalar o pacote *libmysql++-dev* (se optar por PostgreSQL, instale o *libpq-dev*);
- 6) Pode-se executar: *./configure* (apenas optando pelo banco MySQL *--with-mysql* ou PostgreSQL *--with-postgresql*);

- 7) Execute o comando `make` (`#make`) para gerar o instalador. Mas atenção, o autostart faz com que o Bacula seja inicializado junto com o sistema operacional;
- 8) Agora execute o comando `make` com a opção `install` para iniciar a instalação do Bacula (`#make install`);
- 9) Execute o comando `#cd $HOME/bacula/bin` e acesse o diretório do Bacula onde se encontra os seus binários;
- 10) Execute o comando `#./bacula start`. Esse script é responsável por iniciar o Bacula;
- 11) Execute o comando `#./etc/bacula/create_bacula_database`. Ele irá criar o banco de dados Bacula. No caso do MySQL, há um “script” específico;
- 12) Execute o script `#/etc/bacula/create_bacula_tables`. Este irá criar as tabelas do banco;
- 13) Por fim, execute o script `#/etc/bacula/grant_bacula_privileges`. Este irá criar o usuário bacula e garantir os privilégios no banco de mesmo nome (ler **Nota DevMan 3**).



### Nota do DevMan 3

**Usuário e senha do banco:** Para todos os scripts mencionados, é possível passar o usuário e senha do banco de dados como parâmetro.

Exemplo: `/etc/bacula/create_bacula_database -u "usuario" -p "senha"`

Depois de instalado o servidor, instale o cliente Bacula nas estações. A sequência de comandos a seguir apresenta a forma de instalação por compilação:

- 1) Descompacte o arquivo `.tar.gz` e, no diretório criado, digite:

```
#./configure --enable-client-only
#make
#make install autostart
```

- 2) Configure o `bacula-fd`. Compilando apenas um dos “daemons” do Bacula, quando a instalação será em um cliente, não há necessidade de instalar todas as opções do Bacula. Assim, podemos compilar apenas os pacotes necessários. As opções de compilação para seleção dos daemons a serem instalados são as apresentadas a seguir. Perceba que neste caso estamos definindo que serão instalados apenas alguns daemons:

```
#./configure
--enable-client-only (apenas compila o File Daemon)
--enable-build-dird (também compilará o Director)
--enable-build-stored (também compilará o Storage Daemon)
```

### Configurando o Bacula

O arquivo criado automaticamente na instalação do Bacula irá funcionar apenas como um ambiente para testes. É indispensável que você faça algumas modificações para que se torne realmente um sistema útil.

Um dos erros mais comuns que podem acontecer na instalação do Bacula consiste na falha de autenticação com o banco de dados

(isso pode ser verificado através do log do Bacula, conforme já visto anteriormente). Quando isso acontece, não é possível se conectar ao *director* (`bacula-dir`) e o *daemon* deixa de estar em execução. Para corrigi-lo, tenha certeza de que rodou o script `/etc/bacula/grant_bacula_privileges` e que o login e senha de acesso ao *catalog* estejam corretos (observe o arquivo `/etc/bacula/bacula-dir.conf`).

Com o Bacula acessível através do *bconsole*, podem-se efetuar algumas configurações básicas para sua customização. Os tópicos seguintes irão guiar o leitor nessas alterações.

O nome do *Director* e do *Monitor* deverá ser personalizado. Essa informação é inserida durante a instalação do Bacula e é utilizado o *hostname* como nome do *director*. Caso o administrador deseje modificar esta informação, esse é o melhor momento de fazê-lo, pois esse ainda é o início da configuração e ainda não há outros clientes configurados. Esta alteração precisa ser feita em todos os arquivos *.conf* do servidor (*bacula-dir*, *bacula-fd*, *bacula-sd* e *bconsole.conf*). Como exemplo, pode-se inserir o novo nome como *devmedia-dir* (*director*) e *devmedia-mon* (*monitor*). As respectivas senhas são randômicas e deverão estar corretamente configuradas, portanto, não é necessária nenhuma alteração nesse momento.

Agora altere o nome do seu único *Job* de *backup*, até então configurado para corresponder ao novo nome do seu *director*. Em nosso caso, renomearíamos de *default-job* para *devmedia-job*. Lembre-se de reiniciar os *Daemons* (ler **Nota DevMan 4**).



### Nota do DevMan 4

**Reiniciar daemons:** A cada passo da instalação, reinicie os daemons para que as mudanças sejam aplicadas. Dessa forma, pode-se verificar possíveis erros na configuração.

O nome do *JobDefs* (definições de job) deve ser alterado, por exemplo, para *debian*. Observe que *JobDefs* são definições gerais para os jobs de *backup* que facilitam a configuração de novos *Jobs*, tornando desnecessária a repetição de informações. Ainda há as opções constantes nos *Jobs* que irão prevalecer sobre as do *JobDefs*. O administrador precisará alterar o nome do *JobDefs* também no Recurso *Job* (*bacula-dir.conf*). No exemplo em questão, também seria para *debian*. Feita esta alteração, reinicie os *Daemons*.

Para trabalhar com a estratégia GFS (ler **Nota DevMan 5**), crie três novas *pools* no Recurso *Pool* (diário, semanal e mensal). Depois, apague as duas *pools* existentes. O nome da *pool* padrão precisa ser alterado no *JobDefs*. Feita esta alteração, reinicie os *Daemons*.



### Nota do DevMan 5

**Esquema GFS:** O esquema GFS (Grandfather-father-son backup) se refere ao mais comum esquema de rotação de mídias para as cópias de segurança. Originalmente concebido para o backup em fitas, funciona bem para qualquer estrutura de backup hierárquico. O método básico consiste em criar três conjuntos, sendo um diário, um semanal e outro mensal. Para melhor entendimento podem ser associados da seguinte forma. Os diários ou filhos serão rotacionados a cada dia com um semanal (ou pai) a cada semana; o semanal é rotacionado, em bases semanais, com um mensal (ou avô).

Agora modifique o agendamento (*WeeklyCycle*), pois o original são apenas exemplos de como configurá-lo. É aconselhável utilizar apenas um agendamento para ambos os *jobs* (*backup* e *backup Catalog*) para que sempre sejam feitos nas mesmas *pools*. Obviamente, o nome *WeeklyCycle* pode ser alterado para um nome mais simpático, desde que as mudanças sejam feitas também no *JobDefs* e em outras partes do *Director*. Feita esta alteração, reinicie os Daemons.

Agora, configure um *storage* (em *bacula-sd* e depois no *bacula-dir*). O *storage* configurado originalmente pelo Bacula (*File, Device = /tmp*) obviamente trata-se apenas de um laboratório de testes. Entretanto, no próprio *bacula-sd* há vários exemplos de configuração para outros dispositivos. Feita esta alteração, também reinicie os Daemons.

O *FileSet* original do *bacula-dir.conf* vem configurado para “backupear” apenas um diretório. Como exemplo de uso inicial, é aconselhável mudar para que seja “backupeado” todo o sistema (“/”, no caso de um servidor GNU/Linux). Altere o nome dele para *devmediafileset*. Essa alteração precisa ser replicada em outras partes do *bacula-dir.conf*. Assim, crie outros *FileSets* para os novos servidores. Entretanto, há vários servidores com perfis semelhantes, que podem usufruir do mesmo *FileSet* – exemplo: servidores GNU/Linux nos quais tudo será “backupeado”, ou seja, todo o sistema (“/”).

Neste momento, o sistema de *backup* básico já está quase pronto para ser utilizado. Obviamente, novos clientes serão adicionados e algumas alterações de customizações mais avançadas precisarão ser feitas.

Na próxima seção serão apresentados os arquivos de configuração com alguns comentários mais importantes. No entanto, não serão explicadas todas as possibilidades visto que o material ficaria muito extenso.

## bacula-dir.conf

O *Bacula Director* (ou diretor Bacula) é o arquivo principal do sistema de backup. Para que tudo funcione, é necessário que, pelo menos, exista um *Director* configurado. Nele estão as principais configurações de backup: clientes, *storages*, *pools*, *file sets*, retenções, agendamentos, etc. Enquanto mais de 99% das alterações na configuração são feitas no diretor, os demais arquivos (*bacula-fd*, *sd* e *bconsole.conf*) são raramente alterados. Em compensação, trata-se do maior e mais complexo arquivo de configuração do Bacula. Na **Listagem 6** é apresentado um arquivo customizado, com alguns comentários sobre as principais opções do Bacula. A ordem entre os “recursos” pode variar sem que haja problema na configuração.

**Listagem 6.** Conteúdo do arquivo *bacula-dir.conf*.

```
Director {
  Name = devmedia-dir
  DIRport = 9101
  QueryFile = "/etc/bacula/scripts/query.sql"
  WorkingDirectory = "/var/lib/bacula"
  PidDirectory = "/var/run/bacula"
  Maximum Concurrent Jobs = 1
  Password = "Ui3dLcjKWIr6QlyTu+PjPK6wHlzl7fZISiCgiZZ3jq"
  Messages = Daemon
  DirAddress = 127.0.0.1
}

JobDefs {
  Name = "DefaultJob"
  Type = Backup
  Level = Incremental
  Client = devmedia-fd
  FileSet = "Full Set"
  Schedule = "WeeklyCycle"
  Storage = File
  Messages = Standard
  Pool = File
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%c.bsr"
}

Job {
  Name = "BackupClient1"
  JobDefs = "DefaultJob"
}

Job {
  Name = "BackupCatalog"
  JobDefs = "DefaultJob"
  Level = Full
  FileSet = "Catalog"

  Schedule = "WeeklyCycleAfterBackup"
  # This creates an ASCII copy of the catalog
  # Arguments to make_catalog_backup.pl are:
  # make_catalog_backup.pl <catalog-name>
  RunBeforeJob = "/etc/bacula/scripts/make_catalog_backup.pl MyCatalog"
  # This deletes the copy of the catalog
  RunAfterJob = "/etc/bacula/scripts/delete_catalog_backup"
  Write Bootstrap = "/var/lib/bacula/%n.bsr"
  Priority = 11
}

Job {
  Name = "RestoreFiles"
  Type = Restore
  Client = devmedia-fd
  FileSet = "Full Set"
  Storage = File
  Pool = Default
  Messages = Standard
  Where = /nonexistent/path/to/file/archive/dir/bacula-restores
}

FileSet {
  Name = "Full Set"
  Include {
    Options {
      signature = MD5
    }
    File = /usr/sbin
  }

  Exclude {
    File = /var/lib/bacula
    File = /nonexistent/path/to/file/archive/dir
    File = /proc
    File = /tmp
  }
}
```

**Continuação: Listagem 6.** Conteúdo do arquivo bacula-dir.conf.

```
File = /.journal
File = /.fsck
}
}

Schedule {
  Name = "WeeklyCycle"
  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sat at 23:05
}

Schedule {
  Name = "WeeklyCycleAfterBackup"
  Run = Full sun-sat at 23:10
}

FileSet {
  Name = "Catalog"
  Include {
    Options {
      signature = MD5
    }
    File = "/var/lib/bacula/bacula.sql"
  }
}

Client {
  Name = devmedia-fd
  Address = localhost
  FdPort = 9102
  Catalog = MyCatalog
  Password = "4pVPQ7Rca5ydDivqsHJWX9IFRz4gOgU5z"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}

Storage {
  Name = File
  Address = localhost
  SDPort = 9103
  Password = "L7NeeJ6XlsN3sJFjoiRHFuEJTTncT4DGG"
  Device = FileStorage
  Media Type = File
}

Catalog {
  Name = MyCatalog
  dbname = bacula; DB Address = ""; dbuser = "bacula"; dbpassword = "s3curity"
}

Messages {
  Name = Standard

  mailcommand = "/usr/lib/bacula/bsmtp -h localhost -f \"\\(Bacula) \\<%r\\>\" -s \"Bacula: %t %e of %c %l\" %r"
  operatorcommand = "/usr/lib/bacula/bsmtp -h localhost -f \"\\(Bacula) \\<%r\\>\" -s \"Bacula: Intervention needed for %j\" %r"
  mail = root@localhost = all, !skipped
  operator = root@localhost = mount
  console = all, !skipped, !saved
  append = "/var/lib/bacula/log" = all, !skipped
  catalog = all
}

Messages {
  Name = Daemon
  mailcommand = "/usr/lib/bacula/bsmtp -h localhost -f \"\\(Bacula) \\<%r\\>\" -s \"Bacula daemon message\" %r"
  mail = root@localhost = all, !skipped
  console = all, !skipped, !saved
  append = "/var/lib/bacula/log" = all, !skipped
}

Pool {
  Name = Default
  Pool Type = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 365 days
}

Pool {
  Name = File
  Pool Type = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 365 days
  Maximum Volume Bytes = 50G
  Maximum Volumes = 100
}

Pool {
  Name = Scratch
  Pool Type = Backup
}

Console {
  Name = devmedia-mon
  Password = "ID-1Z-7tIOFkLymasyvks7HhyUtKkfxCd"
  CommandACL = status, .status
}
```

Dentro de *Jobs* pode-se alterar as opções de *level*. A seguir são apresentadas algumas delas:

- **InitCatalog:** Faz o *scan* do *fileset* especificado e armazena os atributos dos arquivos no catálogo. Basicamente, permite que seja verificado se houve codificações em arquivos, deleções ou criações. Normalmente é utilizado pela primeira vez que o sistema é “backupeado”, e depois de uma mudança no sistema (exemplo: upgrade);
- **Catalog:** Compara o estado atual de arquivos contra as informações auferidas pelo *InitCatalog*. Qualquer diferença é reportada. Tipicamente este comando é executado diariamente para verificação de mudanças nos arquivos do sistema;

- **VolumeToCatalog:** Este nível faz com que o Bacula leia o atributo do arquivo escrito no Volume e compare com as informações do catálogo;

- **DiskToCatalog:** Esta opção compara os arquivos no disco (sistema) com os copiados no último backup. É útil quando são percebidos problemas no disco, e é desejável saber se o “restore” se faz necessário.

Já em *Pools* podem-se alterar os limitadores de uso dos volumes (necessários para a reciclagem). A seguir podemos observar algumas opções:

- **Use Volume Once = yes:** Com esta opção habilitada (o valor default é *no*) o Bacula só usará o volume uma vez e, após, irá encerrá-lo;
- **Volume Use Duration = ttt:** Período de tempo pelo qual o volume pode ser gravado. Este é iniciado a partir do primeiro “job” para ele submetido. Após este tempo, o volume é automaticamente encerrado;
- **Maximum Volume Jobs = nnn:** Quando atingido o número máximo de “jobs” do volume, o mesmo é encerrado;
- **Maximum Volume Bytes = mmm:** Número máximo de “bytes por volume”. Quando atingido, de igual sorte, o volume se encerra.

Esses limitadores de uso dos volumes podem ser ignorados com o uso da linha *Purge Oldest Volume = Yes*. Se utilizada, todos os limitadores de reciclagem de volumes mencionados serão ignorados (exceto o de tamanho máximo do volume). Trata-se de um segundo método de reciclagem de volumes, mas que não respeita tempos de retenção.

### bacula-sd.conf

O *bacula-sd* (ou *Bacula Storage Daemon*) é o “estoquista” do sistema de *backup*. Trata-se do responsável por armazenar todos os dados, independente de quais dispositivos sejam utilizados. Um *Bacula Director* pode controlar vários *Storage Daemons* em máquinas diferentes. Na **Listagem 7** pode-se observar o exemplo padrão de um arquivo *bacula-dir.conf*.

**Listagem 7.** Conteúdo do arquivo *bacula-dir.conf*.

```
Storage {
    Name = devmedia-sd
    SDPort = 9103
    WorkingDirectory = "/var/lib/bacula"
    Pid Directory = "/var/run/bacula"
    Maximum Concurrent Jobs = 20
    SDAddress = 127.0.0.1
}

Director {
    Name = devmedia-dir
    Password = "L7NeeJ6XIsN3sJFjoIRHFuEJTncT4DGG"
}

Director {
    Name = devmedia-mon
    Password = "3YbvH1lgpDjYTHa3NN3imJ4A5Z0z2bbYd"
    Monitor = yes
}

Device {
    Name = FileStorage
    Media Type = File
    Archive Device = /nonexistant/path/to/file/archive/dir
    LabelMedia = yes;
    Random Access = Yes;
    AutomaticMount = yes;
    RemovableMedia = no;
    AlwaysOpen = no;
}

Messages {
    Name = Standard
    director = devmedia-dir = all
}
```

### bacula-fd.conf

O *bacula-fd* (ou *Bacula File Daemon*) é o “vampiro” do sistema de *backup*. Trata-se do responsável por “sugar” as informações dos computadores (arquivos) e enviá-las para o *Storage Daemon*. Os sistemas de *backup* Bacula geralmente possuem inúmeros *file daemon* instalados em diferentes servidores (ou até estações de trabalho), todos ligados a um mesmo “director”. Na **Listagem 8** pode-se observar um exemplo padrão de um arquivo *bacula-fd.conf*.

**Listagem 8.** Conteúdo do arquivo *bacula-dir.conf*.

```
Director {
    Name = devmedia-dir
    Password = "4pVPQ7Rca5ydDivqsHJWX9lFRz4gOgU5z"
}

Director {
    Name = devmedia-mon
    Password = "DRkBz_tyQ6QZ5GfKMrLIU443exw0Im-EK"
    Monitor = yes
}

FileDaemon {
    Name = devmedia-fd
    FDport = 9102
    WorkingDirectory = /var/lib/bacula
    Pid Directory = /var/run/bacula
    Maximum Concurrent Jobs = 20
    FDAddress = 127.0.0.1
}

Messages {
    Name = Standard
    director = devmedia-dir = all, !skipped, !restored
}
```

### bconsole.conf

O último de nossos módulos do Bacula é o *bconsole*. Trata-se da ferramenta que abre as portas para acesso e administração do sistema de *backups*. Para os administradores e operadores, é interessante que se tenha o *bconsole* instalado em seus equipamentos, pois se trata de um acesso mais seguro que o *SSH* na medida em que só permite o acesso ao diretor Bacula em si.

Na **Listagem 9** é apresentada a forma mais simples de configuração do *bconsole.conf*. Nesta configuração definimos o agente do Bacula que está sendo executado em localhost (máquina local). Pode-se observar também o nome do host (localhost-dir), sua porta de atuação (DirPort = 9101), o endereço (localhost, neste caso) podendo ser um endereço IP e a senha definida pelo Bacula.

Além disso, existe a possibilidade de criar usuários que só terão acesso a determinados elementos do sistema de *backup* (um *job* específico, um cliente, etc.).

### Utilizando Include para organizar os arquivos de configuração do Bacula

Ao invés de declarar todos os recursos dentro de um mesmo arquivo de configuração (ex.: *Jobs* dentro do *bacula-dir.conf*), é possível fazê-lo em arquivos distintos apenas citando os caminhos

para os mesmos conforme o exemplo na **Listagem 10**, onde são inseridas várias referências a configurações externas como jobs, storages, schedules, dentre outras.

**Listagem 9.** Conteúdo do arquivo bacula-dir.conf.

```
#
# Bacula User Agent (or Console) Configuration File
#

Director {
  Name = localhost-dir
  Dirport = 9101
  address = localhost
  Password = "Ui3dLcjKWIR6QlyTu+PjPK6wHlzl7fZISiCgiZZ3jq"
}
```

**Listagem 10.** Exemplo de bacula-dir.conf.

```
@/etc/bacula/bacula-dir-filesystems.conf
@/etc/bacula/bacula-dir-jobs.conf
@/etc/bacula/bacula-dir-jobdefs.conf
@/etc/bacula/bacula-dir-clients.conf
@/etc/bacula/bacula-dir-storage.conf
@/etc/bacula/bacula-dir-pools.conf
@/etc/bacula/bacula-dir-schedules.conf
```

### Ciclo de vida dos volumes

Quando inserido um novo dispositivo de hardware (ex.: uma nova fita ou HD), o primeiro passo para sua utilização junto ao Bacula é o etiquetamento lógico (comando *label*). Assim que “etiquetado” (ou seja, batizado), o novo volume está pronto para gravação (ou seja, status *append*).

Se nada for configurado no *bacula-dir.conf*, este volume será gravado, diariamente, enquanto houver espaço na fita (seu volume ficará sempre *append*, podendo ser gravado de onde se parou, até o momento que se torne *full*). Ou seja, o comportamento natural do Bacula é a gravação contínua dos volumes. Dessa forma, com a configuração original, sempre serão necessários novos volumes, indefinidamente (a não ser que haja intervenção manual), pois não há uma rotação dos dados. Em outras palavras, você não programou o Bacula para automaticamente reciclar os volumes (por exemplo, após determinado período de tempo). Neste sentido, pode ser necessário adotar uma das estratégias de reciclagem apresentadas a seguir.

### Reciclagem bruta

Neste método, o Bacula sempre reciclará o volume mais velho (ou seja, com mais tempo desde a sua última gravação). Desta maneira, sempre que não houver volumes disponíveis para a gravação dentro de determinado grupo de fitas (*pool*), o Bacula irá permitir que os dados do volume mais antigo sejam sobrescritos. A opção que habilita este comportamento localiza-se no *bacula-dir.conf*, dentro do recurso *Pool*: *Purge Oldest Volume = Yes*

A grande desvantagem deste método consiste na necessidade de uma administração proativa para evitar que os dados dos *backups* cresçam demasiadamente, e o tempo de retenção dos dados se torne muito pequeno – já que não há nenhuma espécie de controle sobre a reciclagem.

### Reciclagem por tempos de retenção

É possível estabelecer um tempo de retenção para os volumes de determinado grupo (*pool*) dentro do arquivo *bacula-dir.conf*. Para isso, utilizamos o parâmetro *VolumeRetention*. Um exemplo seria *VolumeRetention = 6 Days*, onde definimos que após seis dias os dados do volume poderão ser sobrescritos. Contudo, para que este tempo de retenção comece a ser contabilizado, é necessário encerrar o volume. Para isso, a opção abaixo pode ser utilizada (também no *bacula-dir.conf*, recurso *Pool*): *Use Volume Once = yes*

Com esta opção habilitada (o valor *default* é *no*), o Bacula só usará o volume uma vez e, após isso, irá encerrá-lo.

Podemos também definir o período de tempo pelo qual o volume pode ser gravado, sendo contabilizado do início do primeiro *job* para ele submetido. Após este tempo, o volume é automaticamente encerrado. Geralmente, este valor é um pouco menor do que sua janela de backups. Para esta definição, o parâmetro abaixo pode ser utilizado: *Volume Use Duration = ttt*

Podemos também definir que quando for atingido o número máximo de *jobs* do volume, o mesmo seja encerrado. Para esta definição, o parâmetro abaixo pode ser utilizado: *Maximum Volume Jobs = nnnn*

Por fim, também podemos definir o número máximo de *bytes* por volume. Quando atingido, o volume se encerra. Para esta definição, o parâmetro abaixo pode ser utilizado: *Maximum Volume Bytes = mmm*

Este parâmetro é indispensável para a utilização de volumes em discos rígidos para possibilitar a gravação de vários volumes e não apenas um para o disco inteiro (o mínimo recomendado são quatro volumes por disco). Para as fitas, essa opção nunca deve ser utilizada, pois a capacidade de gravação das mesmas é variável. No momento que uma destas opções atingir seu limite, o volume deixa de ter status *append* e passa a ser *full*, passando a ocorrer o *Volume Retention*.

Como exemplo, suponha que se deseja obter uma retenção de 7 dias (volume gravado numa segunda-feira seja sobrescrito na próxima segunda, considerando a realização de *backups* diários). Neste caso, as configurações serão:

```
VolumeRetention = 6 Days
Volume Use Duration = 23 hours
```

Assim, o tempo de retenção real dos volumes contidos nesta *pool* seria de 6 dias + 23 horas. Observe que esta retenção deve ser sempre um pouco menor do que o intervalo entre os *backups* (que no caso seria de 7 dias), de maneira a evitar a sobreposição do tempo de retenção com a necessidade de gravação de volume, na próxima segunda-feira.

### Exemplo de GFS no Bacula

A implementação da estratégia GFS clássica no Bacula se dá através da definição de ao menos 03 (três) *pools* distintas (no *bacula-dir.conf*). *Pools* diária, semanal e mensal. Obviamente, você pode chamar de outra maneira (ex.: *daily*, *weekly*, *monthly*), mas a

função delas deverá ser a mesma: hospedar os *backups* para cada hierarquia (diferenciais ou incrementais diários; *full* semanais e *full* mensais). Atente às observações a seguir:

- Quando se deseja que o Bacula sempre utilize a mesma fita para determinado dia da semana (ex.: VOL1 sempre ser gravado às segundas-feiras), deve-se criar *pools* específicas para cada dia da semana (ex.: *diario\_seg*, *diario\_ter*, etc.);
- Pode-se criar uma *pool* para fitas que estão fora do seu robô de fitas para evitar que o Bacula as procure durante a operação de *backup* – e para melhor organizá-las. Para criar uma nova *pool*, basta duplicar as configurações de uma *pool* qualquer (inicialmente a “default”), e alterar seu nome. Não se esqueça de configurá-la (tempo de retenção, tempo de uso do volume, reciclagem – “yes/no”, etc.), tudo isso no *bacula-dir.conf*.

## Agendamento

O *schedule* ou agendamento também é configurado no *bacula-dir.conf*. Você deve associar um *Job* criado neste arquivo a um agendamento. Portanto, aconselhamos criar um novo *schedule* (ex.: *agenda\_gfs*) e ir associando os *Jobs* conforme o exemplo da **Listagem 11**.

### Listagem 11. Exemplo de agendamento.

```
Schedule {
  Name = agenda_gfs
  Run = Level=Differential
  Run = Level=Full
  Pool=Diaria Monday-Thursday at 19:00
  Pool=Semanal 2nd 3rd 4th 5th
  Friday at 19:00
  Run = Level=Full Pool=Mensal 1st Friday at 19:00
}
```

No exemplo, teremos backups diários de segunda a quinta, semanais na segunda, terça, quarta e quinta, e mensais na primeira sexta-feira do mês.

## Retenções do Bacula

Retenção trata-se de um conceito muito importante para quem precisa trabalhar com *backups*. Esta expressão se encontra presente em quase toda ferramenta do gênero. Retenção consiste no período de tempo no qual os dados gravados em um *backup* não podem ser apagados, nem pelo sistema nem por intervenções humanas convencionais.

A única forma para apagar as informações seria explicitamente indicando que a ferramenta o fizesse (no caso do Bacula, com o comando *purge*). Esta proteção visa ainda evitar erros de um operador (ex.: colocar a fita errada no drive), ou do administrador através do console. Em linhas gerais: a retenção é a segurança dos seus dados. Depois de terminado o tempo de retenção, se diz que ele expirou e, dessa forma, o volume poderá ser sobrescrito em um próximo trabalho de *backup*.

No sistema GFS clássico, a retenção para os *backups* diários deve ser de aproximadamente 7 dias. No semanal, de aproximadamente um mês. E no mensal, de aproximadamente um ano.

Geralmente pode ser um pouco menos, para evitar a sobreposição do tempo de retenção com a data na qual o volume deve ser sobrescrito, como vimos anteriormente. Lembre-se que o tempo de retenção do volume só começa a ser contado quando o mesmo é encerrado, ou seja, quando ele enche (status *full*) ou quando previamente foi configurado um dos limitadores.

A Retenção do Volume não se confunde com as retenções de informação no catálogo (*jobs* e *file retention*), que servem como limitadores do tamanho do banco e podem ser apagadas automaticamente se a opção *auto prune* estiver ativada.

## File e Job Retention no Bacula

As duas retenções em negrito na **Listagem 12** servem apenas para preservar informações do catálogo do Bacula (banco de dados), especificamente para este cliente. Se o *Auto Prune* estiver ativo, após este tempo, as informações de *file* e *jobs* serão automaticamente apagadas. Ou seja, essas retenções servem para limitar o tamanho do Catálogo do “Bacula”.

### Listagem 12. Exemplo de File e Job retention.

```
Client {
  Name = devmedia-fd
  Password = "senhadadevmedia"
  Address = x.x.x.x
  FDPort = 9102
  Catalog = MyCatalog
  AutoPrune = yes
  File Retention = 30 days
  Job Retention = 6 months
}
```

## File Retention

O *file* são as informações sobre os arquivos gravados em cada volume do *backup*. É um verdadeiro índice que permite a restauração parcial de arquivos de um determinado *job*. Se esta informação for expirada, não é mais possível selecionar alguns arquivos de um *job* para restauração, mas apenas o *job* inteiro.

## Job Retention

A informação do *job* permite que ele seja restaurado pelo Bacula. Sem esta informação, só é possível a restauração através do *bextract*, ou se o *bscan* for utilizado no volume para restaurar as informações do catálogo.

Cuidado com essas duas opções. Se há um bom espaço em disco para o banco de dados Bacula, deve-se sempre aumentar estes parâmetros, principalmente a retenção do *job*. Se estas duas retenções forem maiores do que o tempo de reciclagem (ou retenção) do volume, não há problema, pois a reciclagem do volume também irá apagar estas informações do catálogo para aquele volume específico.

## Novos clientes Bacula

Nesta seção será apresentado como configurar novos clientes para *backup* no Bacula. Sempre que novos hosts forem planejados

para fazerem parte do projeto de backup, alguns passos devem ser respeitados para que o processo seja concluído com sucesso. O primeiro arquivo a ser configurado é o *bacula-dir.conf*. Observe a sequência de configurações para que o cliente seja inserido:

1. Definir um novo *job* para o cliente a ser criado;
2. Criar um novo recurso *Client*. A senha (*password*) será a mesma que consta no *bacula-fd.conf* do cliente correspondente;
3. Criar um novo *FileSet*, caso os arquivos a serem “backupeados” sejam diferentes do *FileSet* que já existe. Obs.: no caso do Windows, lembrar que deve ser utilizado / (barra) ao invés de \ (barra invertida);
4. No recurso *storage*, certificar-se de que o endereço utilizado seja um IP ou de que o nome da máquina esteja num servidor DNS;
5. Reiniciar os serviços do Bacula.

O segundo arquivo a ser modificado é o *bacula-fd.conf*. Veja a seguir a sequência de configurações:

1. Colocar o nome do director;
2. Modificar a senha que o director irá utilizar para se conectar ao cliente;
3. Reiniciar o daemon ou serviço (bacula-fd).

### Conclusão

Este artigo apresentou como configurar o Bacula para efetuar cópias de segurança em uma organização. Como pôde ser observado, o assunto é bastante extenso para que possa ser trabalhado por completo nesse artigo. Assim, apresentamos aqui algumas informações importantes para quem deseja iniciar os estudos e implementações com essa aplicação.



#### Flávio Alexandre dos Reis

[reis.falexandre@gmail.com](mailto:reis.falexandre@gmail.com) – [www.reisfa.eti.br](http://www.reisfa.eti.br)  
Especialista em Segurança da Informação – FMG/JF, Pós Graduando em Redes de Computadores – CES/JF, Certificado LPIC-1, CLA. Atualmente professor em Instituições de Ensino Superior da Região de Juiz de Fora em disciplinas ligadas a Redes de Computadores e Sistemas Operacionais.



#### Eduardo Pagani Julio

[epagani@gmail.com](mailto:epagani@gmail.com)  
Doutorando e Mestre em Computação pela Universidade Federal Fluminense. Atualmente professor do curso de Ciência da Computação da Universidade Federal de Juiz de Fora em disciplinas ligadas a redes.



#### Source Forge

<http://sourceforge.net/projects/bacula/files/#files>

#### Source Force - Download

<http://sourceforge.net/projects/bacula/files/bacula/5.2.3/bacula-5.2.3.tar.gz/download>

#### Software Livre

<http://softwarelivre.org/bacula>

#### Dicas-L

<http://www.dicas-l.com.br>

#### Bacula

<http://www.bacula.org/en/>

#### Bacula-br

<http://www.bacula.com.br/?p=367>

#### Viva o Linux

<http://www.vivaolinux.com.br>

#### Referências

FARIA, M.H. *Bacula – Ferramenta Livre de Backup*. Editora Brasport.



# Acesso Remoto e Seguro também nas Pequenas e Médias Empresas

## A importância das soluções de VPN

O conceito prático de uma VPN (*Virtual Private Network* – Rede Privada Virtual) é criar um túnel seguro virtual que permita a comunicação entre dois pontos distantes, utilizando uma infraestrutura de rede pública e não confiável. A segurança na comunicação dos dados que trafegam por esse túnel é realizada por meio do uso de criptografia entre a origem e o destino, eliminando assim os riscos de uma eventual interceptação das informações durante sua transmissão. Usando uma analogia muito empregada para explicar o conceito de um túnel VPN, é o exemplo do Eurotúnel. Imagine que você quer ir da França para a Inglaterra e para isso você utiliza um barco e navega pelo Canal da Mancha. Neste trajeto, qualquer um pode ver o seu barco, ver as pessoas que estão com você e até a roupa que você está vestindo. Não há privacidade. Agora imagine que você esteja percorrendo o mesmo trajeto, só que passando pelo Eurotúnel (uma VPN). Pelo Eurotúnel só há uma entrada e uma saída, não permitindo a interceptação de outras pessoas no meio do caminho, como também ninguém fora do túnel saberá sequer que você está passando por ali. Isso lhe dá segurança e privacidade durante todo o trajeto.

No mundo virtual, para que isso aconteça, a comunicação via VPN deve prover os seguintes requisitos básicos de segurança:

- **Autenticidade:** a comunicação entre os pontos de uma rede devem ser reconhecidos como autênticos por ambos os lados através da troca de chaves de criptografia, autorizando assim o recebimento e o envio de informações entre as partes. Sem a autenticidade dos dois pontos, não há garantia que o tráfego esteja sendo enviado para o destino correto;
- **Confidencialidade:** este requisito é visto como o principal item de uma VPN, embora todos sejam necessários para a sua completa realização. Garantir que as informações trafegadas entre os dois pontos pelo

### Resumo DevMan

#### *De que se trata o artigo:*

O artigo apresenta as principais razões para a adoção massiva do acesso remoto seguro nas Pequenas e Médias Empresas por meio das soluções de VPN, as características técnicas dos diferentes tipos de acessos e exemplos de produtos acessíveis a este mercado.

#### *Em que situação o tema é útil:*

Hoje a palavra de ordem é mobilidade, sejam nos negócios, nos estudos ou mesmo nos momentos de lazer. Prover acesso remoto e principalmente seguro aos diversos conteúdos disponíveis é o grande desafio para as equipes envolvidas nas áreas de Tecnologia, Comunicação e Segurança da Informação. Garantir ao mesmo tempo proteção e praticidade neste tipo de acesso pode ser um diferencial de negócio. Recursos como VPN criam a possibilidade de garantir isso através de diferentes métodos de conexão e, o que é mais interessante, a um custo acessível, inclusive para o mercado das Pequenas e Médias Empresas (PME's).

#### *Acesso Remoto e Seguro também nas Pequenas e Médias Empresas:*

Diante deste novo cenário onde a necessidade de mobilidade atinge a todos, especialmente às organizações que atuam nos mais diversificados tipos de negócio, sendo que, ao mesmo tempo, novas ameaças proliferam na rede mundial, é fundamental que a comunicação entre os diferentes pontos de acesso sejam realizados com segurança e por que não, de forma simples. É neste ambiente caótico que soluções de VPN ganham destaque, agora também nas Pequenas e Médias Empresas (PME's), devido ao baixo custo de sua implementação somado às atuais necessidades do mercado.

túnel de uma Rede Virtual Privada estejam realmente protegidas contra uma possível captura de pacotes no meio da transmissão é fundamental para a segurança das organizações, uma vez que estas informações passam através de uma rede pública sem proteção;

- **Integridade:** garantir que as informações transmitidas entre os dois pontos não sejam corrompidas e nem alteradas. Isso é muito importante, pois se as informações puderem ser capturadas durante a transmissão, além de comprometer a sua confidencialidade, elas podem também ser alteradas e enviadas erroneamente ao receptor;
- **Disponibilidade:** e por fim, depois de garantir estes três requisitos principais, uma VPN funcional só terá sentido se ela estiver disponível para os usuários autorizados (Autenticidade), permitindo a troca de informações segura (Confidencialidade) e validadas por ambas as partes (Integridade).

## Elementos que compõem uma VPN

Para que uma VPN seja criada, além destes requisitos básicos de segurança, são necessários alguns elementos para compor fisicamente e virtualmente esta rede privada:

- **Servidor VPN:** para que uma conexão VPN seja criada, deve existir um servidor que autentique, criptografe e transmita as informações para o cliente. Na maioria dos casos, em redes menores, o próprio servidor de *firewall* é o responsável por realizar esta função. Em redes mais complexas, deve-se usar um servidor dedicado ou uma solução *appliance* somente para esta função;
- **Cliente VPN:** o cliente é aquele host que solicita uma autorização ao servidor para acessar as informações da rede. O cliente pode ser desde um simples computador de um usuário solicitando acesso a partir de qualquer ponto da Internet, como também um roteador configurado entre duas redes fisicamente identificadas;
- **Túnel:** é o caminho lógico por onde os dados trafegam de forma protegida em uma rede pública. Seria uma analogia para representar que este caminho só possui uma entrada e uma saída, acessíveis somente aos dois pontos previamente configurados para se comunicarem;
- **Protocolos de Tunelamento:** são estes os responsáveis pelo encapsulamento e o gerenciamento das conexões ativas nos túneis das VPNs;
- **Rede Pública:** é a Internet em si, através da infraestrutura disponibilizada pelos provedores de serviços de comunicação por meio do fornecimento dos *links* de comunicação.

Mesmo que a recomendação seja restringir o máximo possível a possibilidade de que a informação seja acessada por terceiros não autorizados, existem opções em que a transmissão dos dados não é criptografada por padrão. A seguir, a descrição de cada opção:

- **Modo Transmissão:** somente os dados são criptografados e os pacotes não sofrem alterações de tamanho;
- **Modo Transporte:** somente os dados são criptografados, mas pode haver mudança no tamanho dos pacotes;
- **Modo Túnel Criptografado:** os dados e os pacotes são criptografados e encapsulados para serem enviados no túnel estabelecido entre os dois pontos da VPN;
- **Modo Túnel não Criptografado:** os dados e os pacotes são encapsulados e enviados no túnel estabelecido entre os dois pontos da VPN, porém sem alterar as informações, tanto do pacote como dos dados.

A criptografia é a chave para a proteção contra o acesso não autorizado aos dados transmitidos entre os dois pontos da VPN, graças à combinação de chaves pública e privadas. A criptografia pode ser realizada de duas formas. Uma utilizando chave simétrica e outra utilizando chave assimétrica:

- **Criptografia por Chave Simétrica:** Utiliza uma única chave para criptografar e descriptografar, o que torna a tecnologia limitada e não tanto segura, pois, de alguma forma, a chave deve ser trocada entre o emissor e o receptor e, neste momento, pode ocorrer uma interceptação;
- **Criptografia por Chave Assimétrica:** Utiliza duas chaves, sendo uma pública e outra privada. Dessa forma, o emissor e o receptor possuem duas chaves cada. A pública, que pode ser divulgada, e a privada, que fica somente com o proprietário. Por exemplo, se Paulo precisa enviar uma mensagem segura para João, ele deverá realizar os seguintes passos:

1. Paulo escreve uma mensagem no seu gerenciador de e-mails e a criptografa utilizando a chave pública disponibilizada por João. A chave pública de João está disponível para qualquer pessoa;
2. Paulo então envia a mensagem através da Internet para o endereço eletrônico de João;
3. João recebe a mensagem e executa então o processo de descriptografia utilizando sua chave privada, que somente ele possui;
4. João consegue assim ler a mensagem. Para responder a mensagem de Paulo com o mesmo nível de segurança, ele utiliza então a chave pública que Paulo disponibilizou. Ao chegar a mensagem, Paulo utilizará a sua chave privada que somente ele possui.

Dessa forma, ninguém mais, senão o proprietário da chave privada, possui a condição de realizar a descriptografia da mensagem recebida. Além disso, não há troca de chaves privadas, o que impede a sua interceptação.

Os principais algoritmos utilizados para a criptografia são:

- DES (*Data Encryption Standard*);
- Triple-DES;
- RSA (*Rivest Shamir Adleman*);
- Diffie-Hellman.

Também são utilizados alguns algoritmos para garantir integridade, a saber:

- SHA-1 (*Secure Hash Algorithm One*);
- MD5 (*Message Digest Algorithm 5*).

A criptografia utilizada nas conexões VPN normalmente utiliza um dos seguintes protocolos:

- **PPP**, ou Protocolo Ponto a Ponto, é o responsável por transportar pacotes entre dois dispositivos de rede em uma conexão física ponto a ponto, na camada de enlace do modelo OSI (camada 2);
- **PPTP**, ou Protocolo de Tunelamento Ponto a Ponto, foi desenvolvido pela Microsoft em conjunto com outras empresas de tecnologia. É uma variação do protocolo PPP, muito difundida para o uso de conexões remotas. Foi um dos primeiros protocolos utilizados para a realização de conexões VPN. O PPTP encapsula

pacotes PPP utilizando-se de uma versão modificada do GRE, o que torna o PPTP capaz de lidar com outros tipos de pacotes além do IP, como o IPX (*Internet Packet Exchange*) e o NetBEUI (*Network Basic Input/Output System Extended User Interface*), também baseado na camada 2 do modelo OSI (enlace);

- **L2TP**, ou Protocolo de Tunelamento da Camada 2, também utiliza uma variação do protocolo PPP com o extinto padrão utilizado pela CISCO, L2F (*Layer 2 Forwarding*). As mensagens do protocolo L2TP são de dois tipos: Mensagens de Controle, utilizadas para estabelecer e manter as conexões, e Mensagens de Dados, utilizadas para transportar informações. O L2TP realiza o encapsulamento de pacotes PPP para permitir o uso dos recursos de autenticação. Suporta também autenticação do túnel, permitindo que a autenticação seja realizada nas duas pontas;

- **AH**, ou Cabeçalho de Autenticação, é um protocolo utilizado para garantir a autenticidade e a integridade no pacote IP, mas sem confidencialidade durante a transmissão. O protocolo AH acrescenta ao cabeçalho do pacote IP dados de autenticação para checagem do receptor, através de uma função *hash* MD5 e SHA-1;

- **ESP**, ou na tradução mais próxima da sigla, Encapsulamento Seguro de Dados, fornece a criptografia e a autenticidade dos dados do pacote IP. O protocolo ESP trabalha tanto no modo transporte como no modo túnel. Na continuidade do artigo serão detalhados cada um destes modos;

- **IPSec**, ou Protocolo de Segurança da Internet, é o protocolo responsável por prover a segurança nas conexões via Internet entre diferentes *hosts* de uma rede privada, utilizando o mesmo padrão de comunicação. Dessa forma, dois pontos distantes podem utilizar a infraestrutura de uma rede pública (como a Internet) para transmitir de forma segura informações que devem ser acessíveis apenas para os *hosts* localizados em cada uma das extremidades. O IPSec trabalha na camada 3 do modelo OSI e pode utilizar em conjunto os protocolos AH (*Authentication Header*) e ESP (*Encapsulated Security Payload*). Com isso garante a autenticidade, a confidencialidade e a integridade dos dados enviados nos pacotes IP;

- **SSL / TLS** são abreviações em inglês que significam Camada de Conexão Segura / Segurança de Camada de Transporte. Ambos são protocolos de criptografia muito utilizados nos serviços de e-mail e navegação, permitindo que as informações enviadas pelo gerenciador de mensagens ou o browser sejam transmitidas de forma segura. O protocolo SSL / TLS oferece criptografia e autenticação no servidor através de uma sessão remota, garantindo segurança entre a comunicação do cliente com o servidor. O protocolo TLS seria a versão atualizada do protocolo SSL;

- **SSH**, abreviação de *Secure Shell*, ou na tradução aproximada do termo, pode-se entender como Sessão Segura, é ao mesmo tempo um programa como também um protocolo usado para acesso remoto protegido, que substitui o famoso Telnet, por permitir uma comunicação criptografada.

## Tipos de Conexões VPN

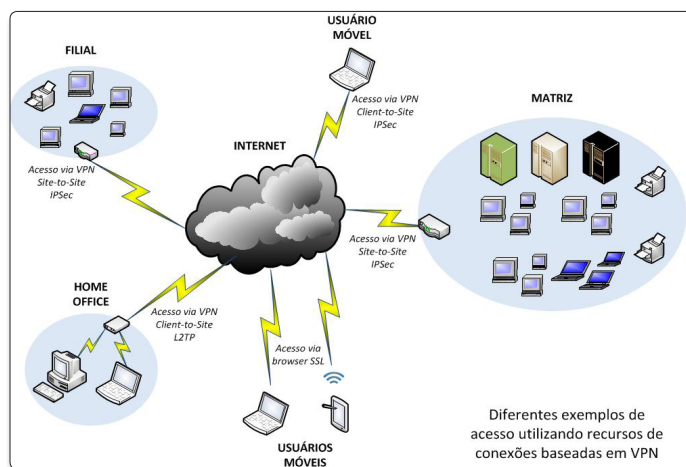
Depois destes detalhes técnicos, pode-se identificar a finalidade da criação da VPN. Ela pode ser destinada para três finalidades:

- **VPN para Intranet:** também chamada de VPN *Lan-to-Lan* ou *Site-to-Site*, é utilizada para interligar duas unidades de uma mesma organização, facilitando assim a troca de informações seguras entre departamentos de uma mesma empresa fisicamente distantes. Também permite uma melhor centralização das informações, uma vez que todas as unidades filiais podem se conectar à base de dados da matriz;

- **VPN para Acesso Remoto:** também conhecida como VPN *Client-to-Lan* ou *Client-to-Site*, é a comunicação que permite que dispositivos móveis como *notebooks* ou *smartphones* ou mesmo *desktops* comuns fechem um canal de comunicação direto com a rede. Nesse caso, deve haver um gerenciamento da rede controlando os acessos simultâneos dos usuários, uma vez que a tendência deste tipo de comunicação está aumentando significativamente nas organizações devido à expansão do conceito de *home-office*;

- **VPN para Extranet:** este tipo de VPN tem como objetivo permitir a troca de informações entre diferentes organizações, como por exemplo, fornecedores, clientes ou empresas pertencentes a um mesmo grupo de sócios. Isso agiliza o processo de troca de informações entre as partes. Em muitos casos esse tipo de VPN acaba também sendo considerada uma VPN *Lan-to-Lan* ou *Site-to-Site*.

A **Figura 1** apresenta alguns exemplos das diferentes formas de VPN citadas.



**Figura 1.** Diagrama dos diferentes tipos de VPN

## Aplicação nas Pequenas e Médias Empresas

No contexto da aplicação de VPN nas Pequenas e Médias Empresas, pode-se imaginar as seguintes situações nos respectivos cenários:

- **Cenário I** – A direção da empresa em que você trabalha, uma pequena metalúrgica do grande polo industrial da região da Grande São Paulo, com cerca de 200 funcionários, sendo que destes, aproximadamente 60 são usuários dos sistemas de informação, acaba de lhe informar sobre a aquisição de duas novas empresas menores no Nordeste do país, como parte de seu planejamento de expansão dos negócios. Estas empresas, cada uma com cerca de 20 usuários de sistemas, precisarão utilizar o mesmo sistema de gestão ERP utiliza-

do em sua unidade, agora definida como a matriz da empresa. Com certeza, é inviável lançar mão de uma infraestrutura própria ou contratar um provedor de serviço para interconectar todas as três unidades, ainda mais quando estamos tratando do âmbito de uma Pequena Empresa. Então, qual seria a melhor solução?

• Cenário II - Você, gerente de TI de uma pequena empresa de logística, responsável pela gestão de cerca de 120 estações e 4 servidores, que mesmo sendo pequena tem como um de seus principais requisitos manter 24 horas no ar os seus servidores responsáveis pelo Sistema de Frotas, Monitoramento dos Caminhões, Armazenamento dos Dados, além dos serviços de Internet e e-mails. Para manter essa infraestrutura disponível a todos os usuários da empresa, você conta com um grupo de diferentes parceiros técnicos contratados que prestam serviços de suporte e manutenção em todos estes servidores. Como a empresa de logística possui acordos de confidencialidade devido à troca de informações estratégicas com seus principais clientes, é fundamental que ela garanta segurança, inclusive durante os acessos dos parceiros técnicos aos seus servidores. É essencial, por exemplo, que o DBA responsável pela manutenção do banco visualize apenas o sistema de gestão do banco de dados e nada mais. Qual a forma mais simples e eficaz para este caso, sem exceder o baixo orçamento disponível?

• Cenário III – Um escritório de advocacia e cobrança que presta serviço para a rede bancária, especializado em processos de negociação de aquisição e apreensão de veículos para pessoas físicas e jurídicas, necessita disponibilizar aos seus advogados e cobradores acesso remoto aos principais sistemas da empresa: Sistema de Cobrança, Consulta de Processos e o Serviço de FTP. Como se tratam de informações altamente confidenciais pertencentes aos bancos, estes profissionais externos que ficam alocados em diferentes escritórios e alguns em seus *home-offices* espalhados em todo território nacional, devem ter, além de dispositivos móveis criptografados, a garantia de acesso remoto seguro como também simplificado, uma vez que boa parte dos usuários possui pouca intimidade com esses recursos tecnológicos. O que você sugeriria?

## Soluções VPN para PME's

Bem, por mais diferente que se pareçam os cenários, existem duas coisas em comum aqui. A primeira é que todas estas organizações são PME's, o que significa que não possuem um grande *budget* de investimento voltado para gastos com TI, como também não possuem grandes equipes dedicadas para estarem em diferentes pontos realizando as configurações e manutenções destas conexões de rede, bem como treinar os usuários sobre os diferentes métodos de acessos que cada um precisará ter.

A segunda coisa em comum é que, para todas as situações, pode-se fazer o uso de soluções de VPN (*Virtual Private Network*), voltadas especificamente para o mercado das PME's, a custos acessíveis, de simples implementação e, principalmente, de fácil utilização para todos os usuários, onde quer que eles estejam.

É pensando nisso que alguns fabricantes internacionais de soluções de *Firewall* e Acesso Remoto como a SonicWALL e a Fortinet, que antes destinavam parte de seus produtos para grandes organizações, começaram a voltar os seus olhos para este imenso mercado, assim como empresas nacionais como a BRConnection e Aker Security Solutions, que já nasceram focadas em desenvolverem soluções de acesso seguro aos principais serviços de Internet e e-mails para as PME's.

Segundo os últimos dados do SEBRAE, as PME's representam 99,1% das empresas brasileiras. São empresas que possuem até 500 funcionários e a maioria delas depende de recursos tecnológicos. Isso graças à expansão da Internet e à queda acentuada dos preços de produtos como computadores e celulares.

Mas como estas soluções de VPN podem realmente ajudar estas organizações? A resposta é "SIMPLES". Isso mesmo, SIMPLES. As conexões realizadas por meio dos diferentes tipos de VPN atualmente permitem que as configurações de implantação sejam executadas de forma simples, utilizando assistentes de instalação passo a passo. Os acessos realizados pelos usuários são feitos da forma mais simples possível, através do fornecimento do nome de usuário e senha no próprio browser. Permitem também o simples gerenciamento e monitoramento por parte dos administradores da solução, mesmo sem conhecimentos avançados em redes e, por fim, permitem a redução de custo com planos de licenciamento e de suporte simplificados para atender à realidade destas pequenas e médias empresas.

Tem-se, como exemplo, as soluções de acesso remoto SRA 1200 da SonicWALL (Figura 2) e o OMNE officeANYWHERE da BRConnection. Ambas desenvolvidas para o mercado das PME's, com excelente relação

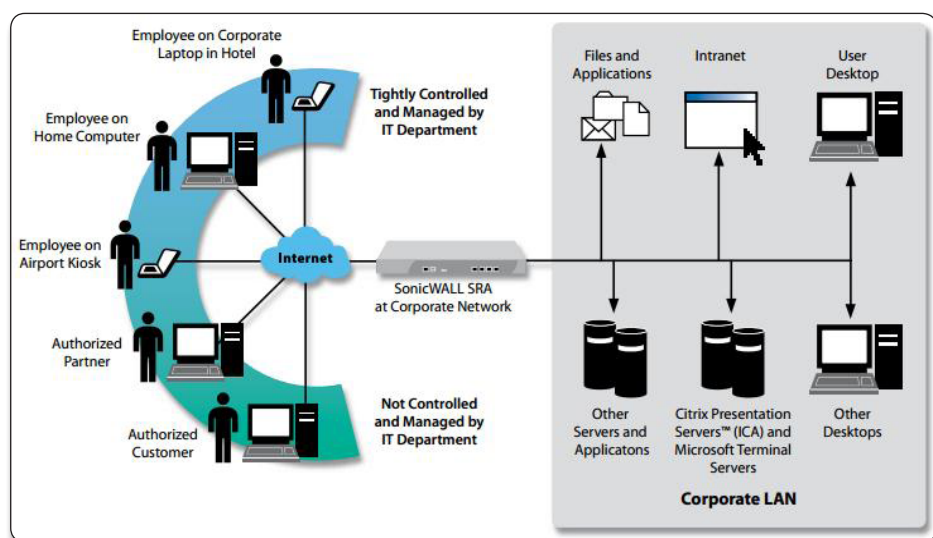


Figura 2. Diagrama da solução SonicWALL SRA

de custo benefício e de fácil utilização, especialmente para os usuários finais.

Estas soluções fornecem opções de conexão *Lan-to-Lan* e *Client-to-Lan*, seja entre produtos do mesmo fabricante, como também permitem a interconexão entre equipamentos distintos, devido à padronização dos protocolos de criptografia e comunicação. O recomendado é que as unidades pertencentes a uma mesma organização padronizem os seus dispositivos de conexão para VPN, facilitando assim a administração da unidade central responsável e favorecendo a redução de custos.

Mas o principal atrativo destas atuais soluções de VPN é a facilidade de conexão, que permite aos usuários finais uma agradável experiência na realização de acesso remoto, o que até pouco tempo atrás era sinônimo de complexidade para o usuário, como também para a equipe de suporte técnico. Hoje, com recursos como o Virtual Office, que permite aos usuários remotos acesso aos recursos de forma personalizada através de um portal acessado pelo browser, com toda a segurança de uma VPN tradicional, tanto a configuração da ferramenta como o acesso para o usuário final é simplificado ao extremo, garantindo assim a praticidade sem abrir mão da segurança (Figura 3).

Por mais que o profissional da área de TI goste de tecnologia, com todas as suas siglas e suas inúmeras especificações técnicas, esses detalhes devem ser transparentes aos usuários comuns. Eles não querem saber como é o processo de criptografia ou muito menos como devem configurar um cliente para conexão VPN em seus computadores. O usuário comum quer apenas entrar na Internet, clicar num botão e ter acesso ao seu recurso remotamente, como no exemplo da Figura 4, onde através de um portal o usuário precisa apenas clicar no módulo que deseja trabalhar e pronto, a tela de acesso é disponibilizada sem que ele tenha conhecimento profundo de como aquele acesso chegou até ele.

Acessar um computador remotamente, um sistema de ERP ou mesmo enviar arquivos em uma área de FTP da empresa, tudo isso, para quem não é da área de tecnologia, pode mais confundir do que

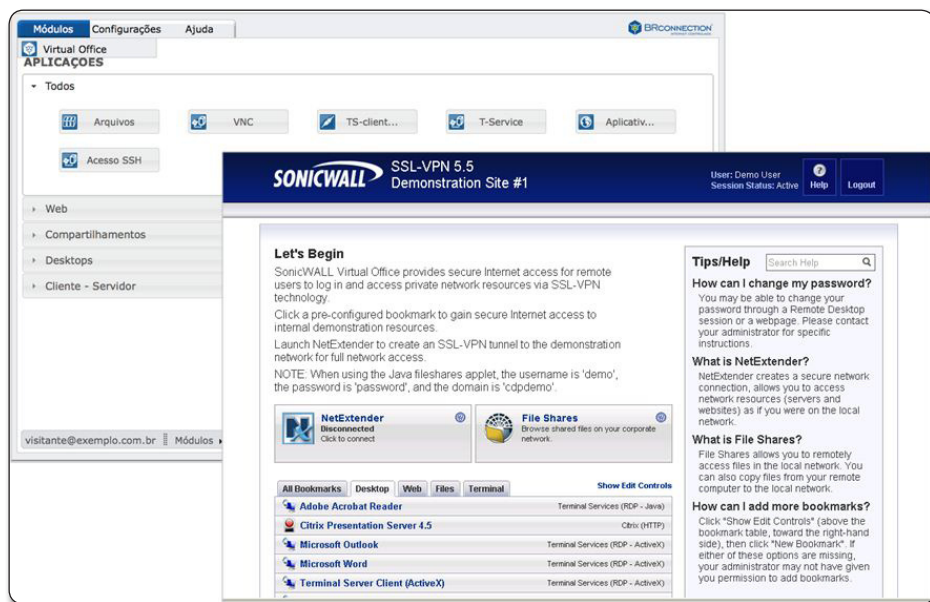


Figura 3. Portais do recurso Virtual Office das soluções BRConnection e SonicWall

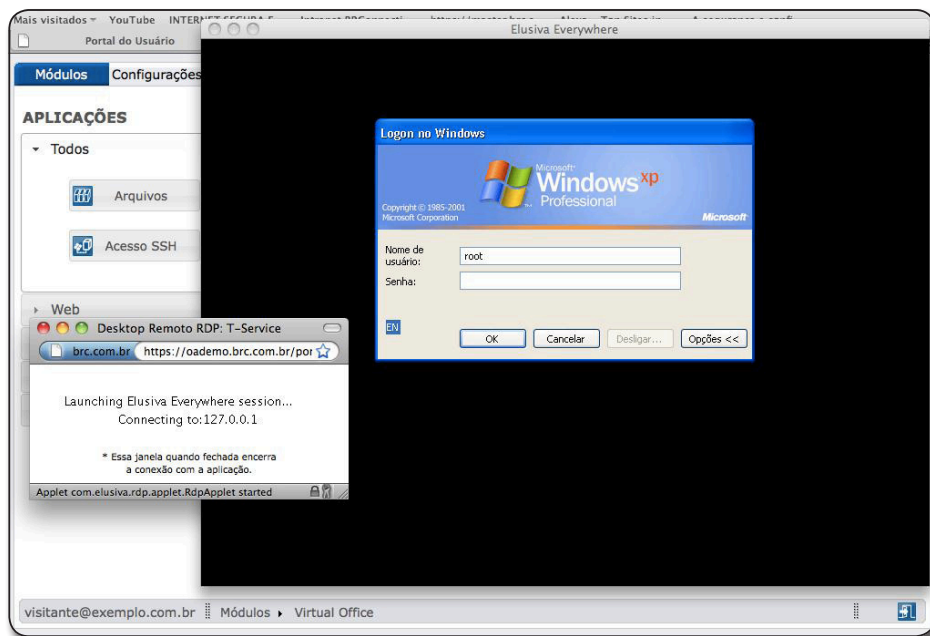


Figura 4. Acesso ao Desktop Remoto com o OMNE officeANYWHERE

facilitar o trabalho. Por isso estes novos recursos podem ser o melhor argumento para um *case* de sucesso.

Além disso, ao escolher uma solução de VPN, deve-se lembrar de ter o fornecedor como parceiro, pois por mais simples que pareça, qualquer tecnologia pode apresentar falhas eventuais e são nestes momentos que uma sólida parceria pode livrar o profissional de TI de possíveis problemas dentro de uma empresa.

## Conclusão

Como descrito no artigo, as soluções atuais para criação de uma VPN estão disponíveis a todos os tipos e tamanhos de empresas e instituições que necessitem estabelecer uma comunicação entre suas redes fisicamente separadas a longas distâncias. Por usarem a Internet como meio de transmissão, as soluções de VPN podem chegar aos pontos mais remotos, desde que possuam ao menos

uma conexão de banda larga, via ADSL, cabo ou rádio. Baixo valor de investimento e simplicidade na sua utilização são os diferenciais que podem trazer inúmeros benefícios tanto para as equipes de suporte técnico responsáveis pela sua implantação como também para os usuários finais, que são as principais razões para a sua adoção nas Pequenas e Médias Empresas, uma vez que existe uma grande expansão do uso de *home-office* como forma de reduzir os custos e trazer melhor qualidade de vida aos colaboradores.

Por fim, foram citados alguns exemplos de soluções de VPN, mas, mais importante do que conhecer a tecnologia utilizada neste tipo de solução, é fundamental que o profissional de TI saiba argumentar com a direção sobre os reais benefícios que tal investimento pode trazer ao negócio como um todo e não somente ao departamento de TI. Com isso, todos ganham, inclusive os usuários.



**Roberto Henrique de Sousa**

*roberto@abctec.com.br*

É Analista de Segurança da Informação na ABCTec, com 14 anos de experiência na área de TI (Suporte, Gestão, Consultoria), especializado em análise de vulnerabilidades e no tratamento de incidentes de segurança da informação. Formado em Análise e Desenvolvimento de Sistemas, possui as certificações ISFS® ISO/IEC 27002 Certified, F-Secure Certified Technical Professional - FSCTP, D-Link DBC, Microsoft MCP/MCDST. Escreve artigos para sites e revistas especializadas em Tecnologia e Segurança da Informação. Membro do Comitê Técnico ABNT/CB21:CE27 sobre Segurança da Informação e membro do Núcleo de TI do CIESP - São Bernardo do Campo.



## Rede Nacional de Ensino e Pesquisa

<http://www.rnp.br/newsgen/9811/vpn.html>

## Solução de Acesso Remoto – PME's - SonicWALL

<http://www.sonicwall.com/br/446.html>

## ABUSAR.org

<http://www.abusar.org.br/vpn/vpnport.htm>

## Blogs do autor

<http://abctec.blogspot.com/>



# DÊ UM SALTO EM CONHECIMENTO!



Acesse o maior portal para desenvolvedores da América Latina!

**17**  
mil  
posts

**280**  
mil  
cadastrados

**5**  
milhões de  
page-views  
por mês

# Supercomputação e virtualização

## A utilização de processadores com múltiplos núcleos no suporte à virtualização

Quando os primeiros computadores pessoais entraram no cotidiano das pessoas, o número de aplicativos era bastante limitado. O que normalmente se usava era um editor de textos e, talvez, uma planilha eletrônica. Isso funcionava com o suporte de um sistema operacional (SO) que gerenciava os recursos da máquina – processador, memória e dispositivos de entrada e saída.

Atualmente, o próprio usuário se tornou multitarefa. Em outras palavras, independente de estar usando um notebook, um *tablet* ou mesmo um celular, o usuário deseja manter uma série de aplicações em execução ao mesmo tempo.

Aliás, com o crescimento dos serviços disponíveis na *World Wide Web* nos últimos anos – o que se convencionou chamar de *cloud computing* ou computação em nuvem – os navegadores acabaram assumindo a função de aplicativos multifuncionais, embutindo outras aplicações dentro de si. Com essa quantidade de programas exigindo processamento, somado a um sistema operacional que precisa gerenciar vários recursos, um único processador ficaria sobrecarregado, por mais rápido que seja.

Do ponto de vista dos serviços disponíveis na nuvem, a situação é ainda mais complexa. Nesse caso, são necessários equipamentos muito mais robustos, capazes de realizar um processamento de alto desempenho e assim prover serviços de qualidade.

Contudo, a tecnologia para criar processadores mais rápidos para atender essa demanda de processamento está próxima de um limite imposto por vários fatores. Entre eles o consumo de energia para atingir velocidades de processamento mais altas e a dissipação térmica decorrente desse consumo. Sendo assim, ao invés de aumentar a velocidade de um processador, por que não aumentar o número de núcleos dos processadores?

Os processadores com dois, quatro, oito ou mais núcleos são chamados *multicore* (ou multinúcleo). Com esse acréscimo no número de núcleos, é possível ter mais programas sendo executados simultaneamente ou ter programas que utilizam dois ou mais núcleos ao mesmo tempo, tornando-os mais rápidos, sem aumentar muito o consumo ou a dissipação térmica. Isso se chama de supercomputação ou processamento de alto desempenho.

### Supercomputação como suporte à virtualização

A virtualização é uma estratégia que permite a criação de ambientes virtuais de processamento. Esses ambientes podem conter um SO específico, um dispositivo de armazenamento virtual e um espaço de endereçamento (memória RAM) próprio, além de poder compartilhar sistemas de entrada e saída como recursos de redes de computadores, por exemplo. Geralmente um ambiente virtual contém uma ou mais aplicações ou serviços que dependem de uma configuração específica. Cada um desses ambientes virtuais é emulado por uma máquina virtual que, por sua vez, é vista pelo SO como um processo comum. A **Figura 1** ilustra uma modalidade de relação entre o hardware, o sistema operacional, as aplicações e as máquinas virtuais.

Outra modalidade possível é ter um hypervisor – ou gerenciador de máquinas virtuais – que possui uma relação direta com o hardware, sem interferência de nenhum sistema operacional. Nesse caso, os diversos ambientes virtuais são gerenciados pelo hypervisor e não mais por um sistema operacional específico.

Os sistemas de virtualização muito se beneficiam do hardware com processadores *multicore*. A razão para isso é que os processadores modernos com esta tecnologia permitem o particionamento dos diversos recursos computacionais – especialmente memória e núcleos de processamento – entre os diferentes processos. No entanto, para obter esses benefícios, é importante que o sistema operacional hospedeiro (*host*) – aquele sobre o qual as várias máquinas virtuais serão executadas – providencie o suporte

adequado. Esse suporte deve incluir balanceamento de carga de processamento eficiente entre os diversos núcleos, mecanismos de particionamento e proteção de segmentos de memória, políticas de memória virtual compatíveis com as necessidades de cada máquina virtual, e compartilhamento e gerenciamento dos dispositivos de entrada e saída.

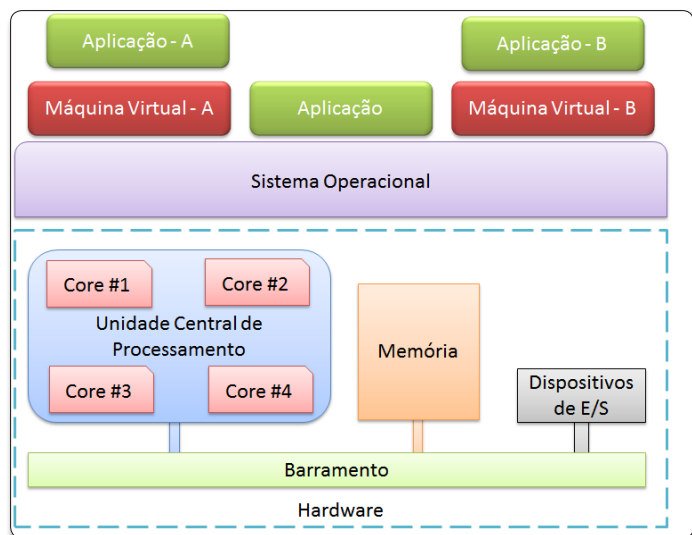


Figura 1. Relação entre os diversos componentes em um sistema com virtualização

## Prós e contras da virtualização com supercomputação

A virtualização tem muitas vantagens a seu favor, mas também há desvantagens a considerar. Entre as vantagens da virtualização, estão os aspectos a seguir:

- Redução de custos na aquisição de hardware, pois se evita adquirir equipamentos diferentes para implementar os diversos servidores. Nesse caso, um único hardware mais robusto pode substituir vários outros com menor robustez;
- Aproveitamento eficiente do hardware, uma vez que haverá mais serviços implementados nas diversas máquinas virtuais executadas sobre um único equipamento;
- Redução do consumo de energia e da necessidade de refrigeração, visto que serão necessários menos equipamentos para executar os diversos servidores embutidos nas máquinas virtuais;

- Melhor utilização do espaço físico, pois em um único hardware pode-se ter várias máquinas virtuais em funcionamento;
- Segurança na recuperação de falhas e acidentes, pois é possível criar “imagens” das máquinas virtuais em dispositivos de armazenamento secundário (CDs ou DVDs). As “imagens” são cópias das máquinas virtuais com suas respectivas configurações. Essas cópias permitem a rápida recuperação de uma máquina virtual.

No entanto, há contrapontos a considerar. São eles:

- Ponto único de falha, visto que quando há falhas no hardware sobre o qual as diversas máquinas virtuais estão em execução, todas essas máquinas falham ao mesmo tempo. O sistema operacional hospedeiro também pode falhar, prejudicando as diversas máquinas ao mesmo tempo;
- Demanda por máquinas muito potentes, pois é importante observar que o poder de processamento do equipamento real vai ser dividido entre as diversas máquinas virtuais;
- Nem todas as aplicações podem ser executadas em máquinas virtuais, pois demandam acesso direto a instruções específicas do processador, algo que não pode ser emulado pelas máquinas virtuais;
- Há questões relativas aos custos das licenças de software, pois, em alguns casos, o modelo de licença varia em função do número de núcleos de processamento do sistema como um todo. Portanto, apesar da aplicação usar uma fração do número de núcleos, o custo da licença pode ser calculado com base no número total de *cores* do processador.

## Alguns dos sistemas de virtualização mais comuns

São muitos os sistemas de virtualização. Citar todos esses sistemas seria inviável. No entanto, a **Tabela 1** mostra uma compilação dos sistemas de virtualização mais comuns baseados em arquiteturas x86-64 bits. É importante destacar que alguns sistemas que aparecem na **Tabela 1** usam licença GPL (*GNU General Public License*) para software livre.

## Conclusão

A virtualização como técnica para melhor aproveitamento do poder computacional disponível não é recente. Começou em meados da década de 1960 com o IBM CP-40. Anos depois, na

| Sistema                            | Fabricante | SO Hospedeiro               | SO da máquina virtual                                     | Licença      |
|------------------------------------|------------|-----------------------------|-----------------------------------------------------------|--------------|
| Hyper-V                            | Microsoft  | Windows 2008                | Linux, Windows e suas diferentes versões                  | Proprietário |
| KVM                                | Qumranet   | Linux.                      | Linux                                                     | GPL versão 2 |
| Virtual Box                        | Oracle     | Windows, Linux, Mac OSX x86 | Linux, Mac OSX, Solaris, Windows                          | GPL versão 2 |
| VMware e seus diversos subsistemas | VMware     | Windows, Linux              | Linux, Mac OSX, Solaris, Windows                          | Proprietário |
| Xen                                | XenSource  | NetBSD, Linux, Solaris      | FreeBSD, NetBSD, Linux, Solaris, Windows XP e 2003 Server | GPL          |

Tabela 1. Sistemas de virtualização mais comuns na arquitetura x86 - 64 bits

década de 1990, a IBM desenvolveu o conceito de particionamento lógico (LPAR), no qual o hardware do computador pode ser dividido entre vários processos, onde conjuntos redundantes de dispositivos (núcleos e memória cache, por exemplo) são distribuídos entre processos distintos.

No entanto, é a partir do desenvolvimento da tecnologia *multicore* que a virtualização pode atingir o seu melhor momento. Com o suporte de um equipamento produzido para multiplicar o poder computacional, apesar de fatores limitantes como consumo de energia, dissipação térmica e compartilhamento de recursos, sistemas virtualizados podem atingir desempenho além do satisfatório.



**André Leon S. Gradvohl**

[gradvohl@ft.unicamp.br](mailto:gradvohl@ft.unicamp.br)

É doutor em Telecomunicações, mestre e bacharel em Computação.

Atua como professor e pesquisador da Faculdade de Tecnologia

(FT) da Universidade de Estadual Campinas (Unicamp). Também é

membro do Laboratório de Simulação e Computação de Alto Desem-

penho (LaSCADo) e do Grupo de Engenharia da Informação e Conhecimento (GEICon),

ambos sediados na FT/Unicamp. É responsável pelo projeto - Metalinguagem para

processamento de alto desempenho em arquiteturas híbridas - financiado pela Fundação

de Amparo à Pesquisa do Estado de São Paulo (FAPESP - 2011/00861-0).



**Artigo "Virtualization on Multicore for Industrial Real-Time Operating Systems", escrito por Chris Main.**

<http://ieeexplore.ieee.org>

**Artigo "Virtualization Shootout: VMware Server vs. VirtualBox vs. KVM", escrito por Bill Childers.**

[www.linuxjournal.com/magazine/virtualization-shootout-vmware-server-vs-virtualbox-vs-kvm](http://www.linuxjournal.com/magazine/virtualization-shootout-vmware-server-vs-virtualbox-vs-kvm)

**Site do sistema Microsoft Hyper-V Server.**

<http://www.microsoft.com/brasil/hyper-v-server>

**Site do sistema KVM – Kernel-based Virtual Machine.**

<http://www.linux-kvm.org>

**Site do sistema Virtual Box.**

<http://www.virtualbox.org>

**Site da Empresa VMware.**

<http://www.vmware.com>

**Site do sistema Xen.**

<http://xen.org>



# CURSOS ONLINE



A Revista .net Magazine oferece aos seus assinantes uma série de Cursos Online de alto padrão de qualidade.



## CONHEÇA OS CURSOS MAIS RECENTES:

- Curso .net Entity Framework - Loja Virtual
- Novidades do Visual Studio 2010
- Componentes DevExpress
- Crie um chat utilizando Asp.Net 3.5 e Linq
- Novidades do C# (csharp) 3.0
- POO com C#
- Microsoft Visual Studio Tools for Office (VSTO)
- Computação na Nuvem com a Plataforma Microsoft Windows Azure

Para mais informações :

[www.devmedia.com.br/curso/netmagazine](http://www.devmedia.com.br/curso/netmagazine)

(21) 3382-5038