# UNIDADE 4 – REDES PRIVADAS VIRTUAIS, AUDITORIA DE SEGURANÇA DA INFORMAÇÃO E CONFIGURAÇÕES DE SERVIDORES
## MÓDULO 1 – VPN PPTP E L2TP

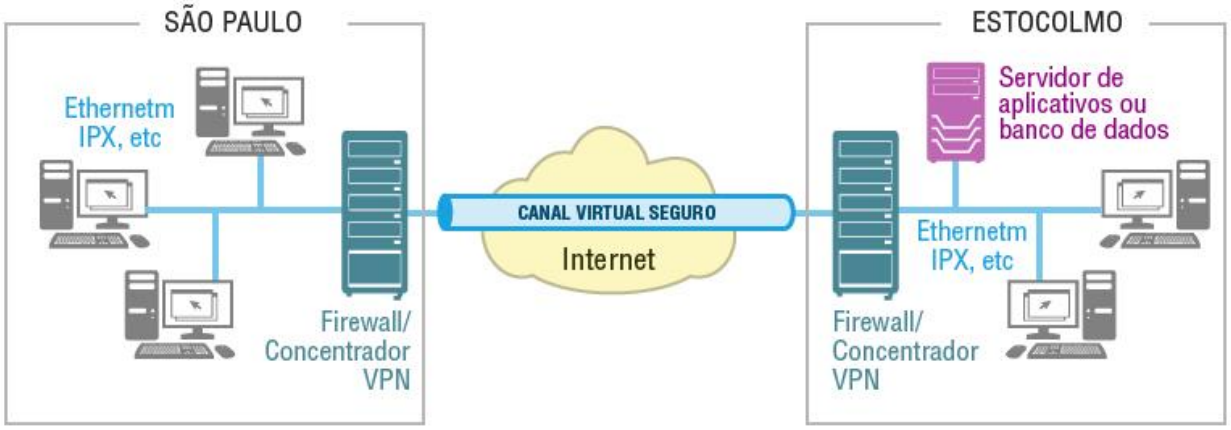
01

## 1 – CONCEITOS BÁSICOS DE VPN

Vimos anteriormente que a criptografia é um assunto extenso e muito importante para a segurança da informação. Veremos agora os aspectos teóricos do uso de VPN, IPSec e VPN SSL, além de abordarmos aspectos práticos dessas aplicações.

### 1.1 - VPN

VPN (Virtual Private Network) é muito utilizada atualmente. A possibilidade de uso de uma rede pública como a internet para interligar escritórios comerciais e grandes empresas tem permitido a redução de custos e viabiliza negócios que têm como premissa requisitos de comunicação eficiente. Dessa forma, gestores de empresas vêm buscando mecanismos para manter as equipes sempre em comunicação, visando diminuir os investimentos em infraestrutura de TI ou até mesmo buscando melhor uso do parque tecnológico já instalado.




**Visão geral de uma VPN.**
**Fonte: Internet, 2015.**

02

É importante citar algumas **possibilidades da VPN** (Virtual Private Network):

a) utilizar uma rede pública para interligar escritórios comerciais, com custos reduzidos;
b) viabilizar negócios que têm como premissa requisitos de comunicação eficiente e transportar dados de modo seguro;
**c)** usar para transferir informações sigilosas usando um canal compartilhado para interligar duas redes privadas protegidas.

Premissas de uma VPN:

a) Confidencialidade dos dados;
b) Integridade dos dados;
c) Não repúdio do emissor;
d) Autenticação da mensagem e
e) Analogia ao modelo OSI.

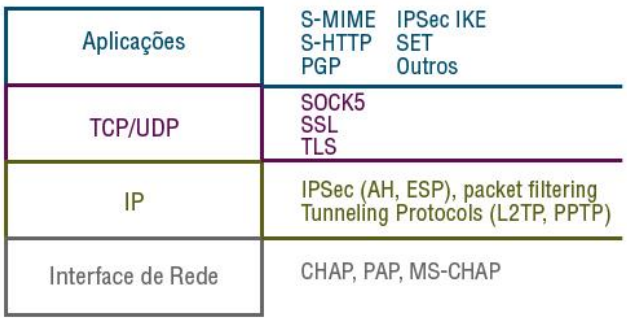

Uma solução efetiva de VPN visa transportar os dados de modo seguro e sigiloso, usando um canal compartilhado para interligar duas redes privadas protegidas.

03

Para que ocorra o transporte de dados nas condições descritas, precisamos alcançar **quatro objetivos** importantes:

| a) Confidencialidade dos dados | b) Integridade dos dados | c) Não repúdio do emissor | d) Autenticação da mensagem |
|---|---|---|---|
| • garantia de que a mensagem não poderá ser interpretada por origens não autorizadas; | • garantia de que o conteúdo da mensagem não foi alterado durante a transmissão entre o emissor e o receptor; | • o emissor não poderá repudiar o envio da mensagem, ou seja, dizer que ele não enviou a mensagem questionada, com embasamento legal; | • garantia de que a mensagem foi enviada por uma fonte autêntica e será entregue a um destino autêntico. |

Vamos recordar o modelo TCP/IP, com 4 camadas para classificar as tecnologias de VPN. Na ilustração, as camadas do modelo TCP/IP e as respectivas aplicações de tecnologias de VPN.




**Modelo TCP/IP versus Tecnologia VPN.**

**Fonte: Internet, 2015.**

Ao se fazer o estudo do Modelo OSI em comparação com a tecnologia VPN, as camadas de Apresentação e de Sessão do modelo OSI não possuem correspondência com a tecnologia VPN. No final, recaímos no modelo TP/IP com pequena diferença na camada de enlace e física.

04

## 2 – OBJETIVOS DE UMA VPN

Vale recordar mais uma vez os objetivos de uma VPN:

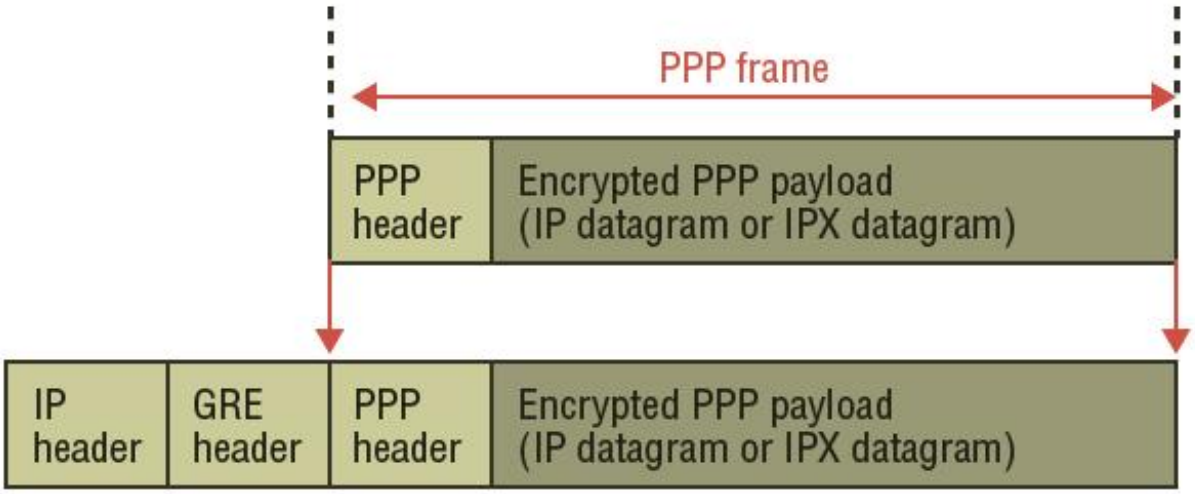
a) Confidencialidade dos dados.
b) Integridade dos dados.
c) Não repúdio do emissor.
d) Autenticação da mensagem.

### 2.1 - Algumas implementações específicas de aplicações VPN

#### 2.2.1 - VPN PPTP (Virtual Private Network Point-to-Point Tunneling Protocol)

Ao analisar o modelo TCP/IP, a VPN PPTP encontra-se na camada de enlace, por ser uma derivação do protocolo Point to Point Protocol (PPP), que consiste em um protocolo de comunicação ponto a ponto, muito utilizado no passado em linhas telefônicas.

O PPTP foi desenvolvido pela Microsoft com o objetivo de incrementar recursos ao PPP. Ele utiliza a autenticação do PPP com um recurso de túnel que pode ser criptografado utilizando criptografia de 40 ou 128 bits.




**Estrutura do Protocolo do Túnel PPTP**

**Fonte: Internet, 2015.**

A Figura acima apresenta uma estrutura básica de um pacote PPTP, contido dentro de um pacote IP.

05

O PPTP é um protocolo orientado à conexão que exige uma estrutura cliente/servidor, logo trafega, por padrão, pela porta TCP 1723. Para estabelecer o túnel PPTP em redes com firewall, é necessário liberar essa porta TCP e utilizar NAT.

Será necessário NAT de um-para-um ou algum protocolo especial para permitir o uso de PPTP (muitas vezes chamado de **VPN passthru**, uma configuração dos roteadores que permite estabelecer uma conexão VPN segura entre dois computadores. Nem todos os roteadores suportam essa característica, portanto, consulte o site do fabricante ou o manual do roteador para verificar a configuração do *hardware*).

Há várias **formas de autenticação do PPP**, sendo as usuais:

| a) **Protocolo de Autenticação de Palavras (PAP)** | b) **Challenge Handshake Authentication Protocol (CHAP)** | c) **MS-CHAP** |
|---|---|---|
| • protocolo de autenticação simples em que o cliente do túnel PPP enviará o usuário e a senha para o servidor em texto claro. | • protocolo de autenticação em que o cliente e o servidor responderão a um desafio, através de senha criptografada com o algoritmo HASH MD5, que, trocada entre o cliente e o servidor, evita que a senha seja transmitida em texto claro. | • protocolo proprietário da Microsoft criado para autenticar estações de trabalho remotas baseadas no Windows, ou seja, um processo de autenticação mútua, com senha unidirecional e criptografada. Tal como o CHAP, o MS-CHAP utiliza um mecanismo de contestação-resposta para autenticar ligações sem enviar a palavra-chave em texto claro. Saiba+ |

Pode-se verificar, assim, que o CHAP e o MS-CHAP são preferíveis ao PAP, pois não trafegam a senha em texto claro.

**Saiba+**

O MS-CHAP utiliza o algoritmo Hash MD4 (Message Digest 4) e o algoritmo de encriptação Data Encryption Standard (DES) para gerar a autenticação challenge/response. O MS-CHAP também fornece mecanismos para comunicar erros de ligação e para alterar a palavra-passe do utilizador.

06

**2.2.2 - L2TP**

O L2TP é um protocolo aberto, especificado na RFC 2661 e foi desenvolvido por um grupo de empresas incluindo Cisco e Microsoft. Utiliza a estrutura cliente/servidor e é orientado a pacotes, usando UDP como protocolo de transporte.

Por utilizar UDP como protocolo de transporte, alguns problemas de desempenho do PPTP foram contornados.

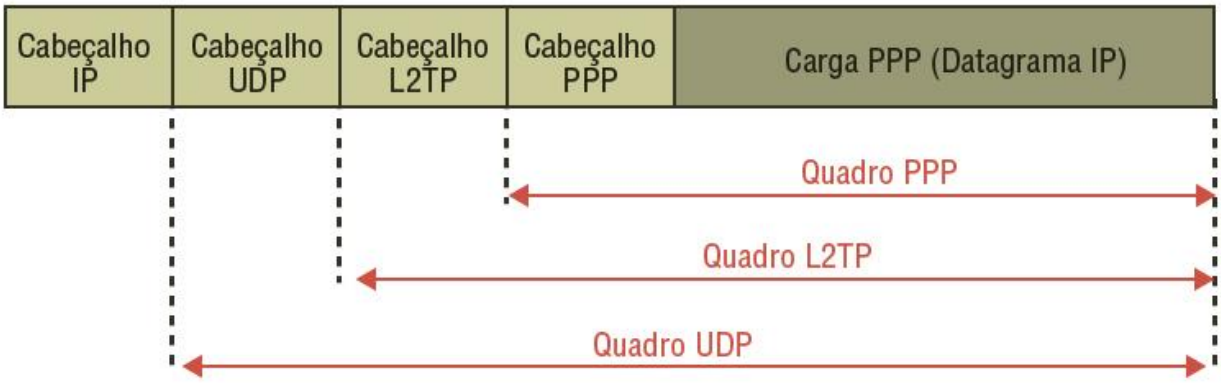



**Figura: Estrutura do Protocolo do L2TP.**
**Fonte: Internet, 2015.**

Para implementar o túnel L2TP em redes com firewall padrão, é necessário liberar a porta UDP 1701. Como é um protocolo orientado a pacotes, o NAT pode ser implementado no modelo um-para-muitos ou um-para-um. Por ser mais leve e prover melhor desempenho, recomenda-se usar o L2TP no lugar do PPTP, sempre que possível.

Deve-se ainda evitar o PPTP, especialmente as versões com chaves de 40 bits, pois diversas vulnerabilidades já foram descobertas nesse protocolo.

07

## 3– IPSEC

IPSec é um conjunto de protocolos, também conhecido como suíte de segurança IP. A segurança de IP (IPSec) é a capacidade que pode ser acrescentada a qualquer versão atual do protocolo Internet (IPv4 e IPv6) por meio de cabeçalhos adicionais.

Os protocolos inclusos na suíte de segurança IP estão focados em:

a) Entrega da mensagem autêntica.
b) Integridade dos dados.
c) Confidencialidade dos dados.
d) Não repúdio do emissor.

O IPSec atua na camada de rede do modelo OSI, por criptografar o conteúdo (*payload*) do pacote IP. Como o IPSec não é um protocolo único, mas sim um conjunto de protocolos, cada qual com um objetivo específico, pode-se chamar o IPSec de suíte de segurança IP. Os protocolos inclusos na suíte de segurança IP estão focados na entrega da mensagem autêntica, com integridade dos dados, confidencialidade dos dados e não repúdio do emissor.

A especificação do IPSec está em várias RFCs, sendo as mais importantes delas emitidas em 1998:

a) RFC 2401;
b) RFC 2402;
c) RFC 2406;
d) RFC 2408.

| |
|---|
| **RFC 2401**<br><br>Descrição da visão geral de uma arquitetura de segurança. |
| **RFC 2402**<br><br>Descrição de uma extensão de autenticação de pacotes para IPv4 e IPv6. |
| **RFC 2406**<br><br>Descrição de uma extensão de criptografia de pacote para IPv4 e IPv6. |
| **RFC 2408**<br><br>Especificação das capacidades de gerenciamento de chaves. |

08

Além das quatro RFCs, diversos rascunhos foram publicados pelos grupos de trabalho do IP Security Procotol. Os documentos estão descritos na RFC 2401, divididos em sete grupos, conforme a figura:

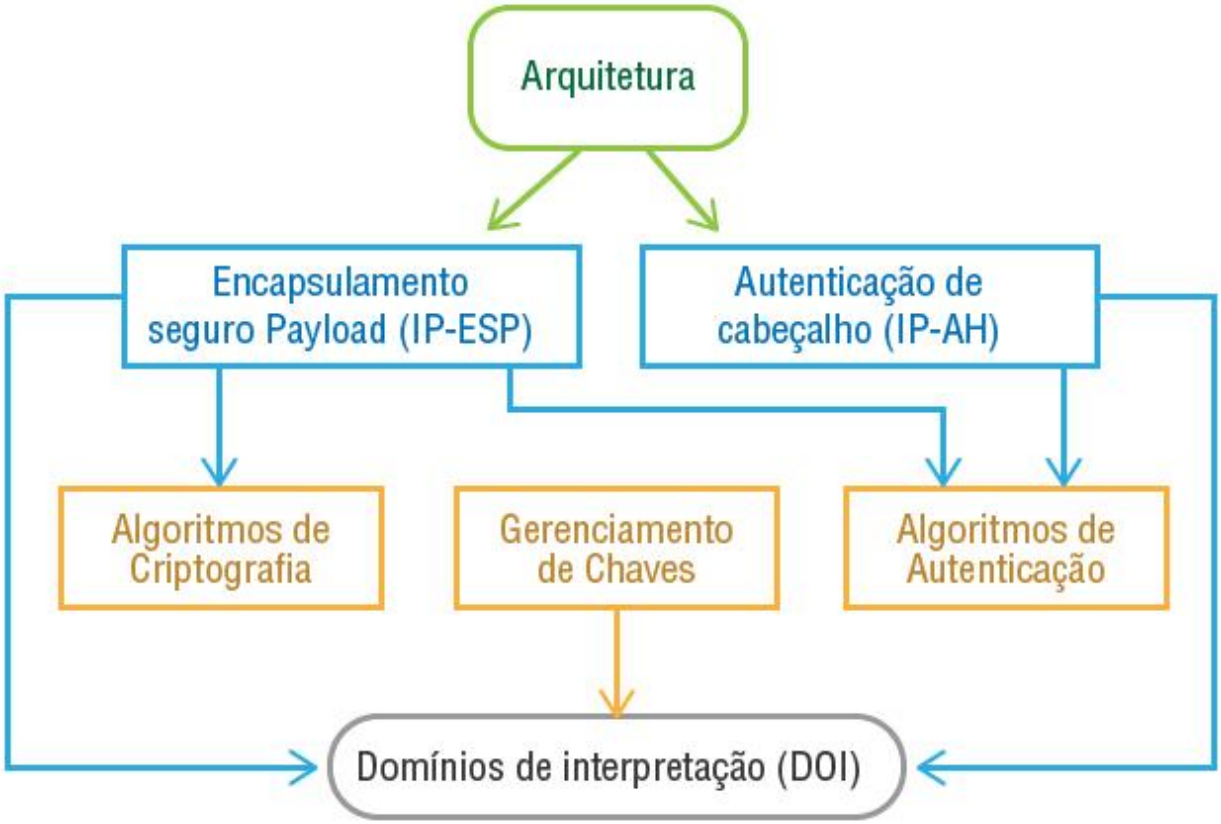



**Figura: Visão do protocolo IPSec (RFC-2401).**
**Fonte: Internet, 2015.**

| |
|---|
| **Arquitetura**<br><br>Abrange os conceitos gerais, os requisitos de segurança, definições e mecanismos, definindo a tecnologia IPSec. |
| **Protocolo ESP (Encapsulating Security Payload)**<br><br>Abrange o formato de pacote e questões gerais relacionadas ao uso de ESP para criptografia de pacote e, opcionalmente, autenticação. |
| **Protocolo AH (Authentication Header)**<br><br>Abrange o formato de pacote e questões gerais relacionadas ao uso do AH para autenticação de pacotes. |
| **Algoritmo de criptografia**<br><br>Um conjunto de documentos que descrevem como diversos algoritmos de criptografia são usados para ESP. |
| **Algoritmos de autenticação**<br><br>Um conjunto de documentos que descrevem como vários algoritmos de autenticação são usados para AH e para a opção de autenticação do ESP. |

| **Gerenciamento de chaves**<br><br>Documentos que descrevem esquemas de gerenciamento de chaves. Exemplo: ISAKMP. |
|---|
| **Domínio de interpretação**<br><br>São valores para os outros documentos se relacionarem entre si. Incluem identificadores para algoritmos aprovados de criptografia e autenticação, além de parâmetros operacionais, como tempo de vida da chave. |

09

**3.1 - Modos de operação do IPSec**

**3.1.1- Cifragem de blocos**: divide os dados em conjuntos de tamanho fixo (chamados de **blocos**):

a) Electronic Codebook (ECB);
b) Cipher-Block Chaining (CBC);
c) Propagating Cipher-Block Chaining (PCBC);
d) Cipher Feedback (CFB);
e) Output Feedback (OFB);
f) Counter (CTR).

**3.1.2- Cifragem stream** – realiza a cifragem de bits. Não há a necessidade de aguardar a formação de um bloco:

a) RC4;
b) A5/1 (usado em redes GSM de telefonia celular).

**3.1.3- Modo de Transporte**: oferece proteção principalmente para os protocolos das camadas superiores. Esse modo de operação do IPSec criptografa todo o payload do pacote IP. Compatível com protocolos IP, UDP, TCP e ICMP.

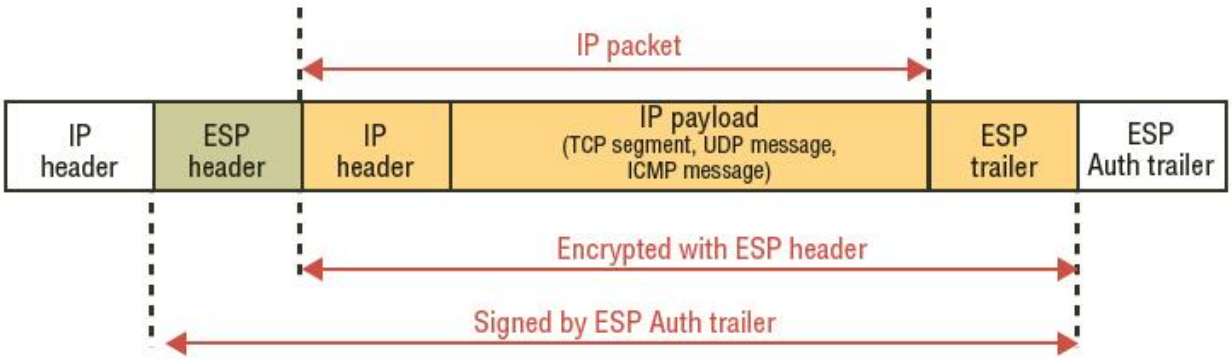



**Figura: Pacote encapsulado ESP no modo transporte.**
**Fonte: Internet, 2015.**

Na figura acima o cabeçalho IP (IP header) e a autenticação ESP (ESP Auth trailer) não são protegidos.

10

**3.1.4- Modo de Túnel**: esse modo de operação oferece proteção a todo pacote IP. Todo o pacote original viaja por um "túnel" de um ponto de uma rede IP para outro e nenhum roteador ao longo do caminho é capaz de examinar o cabeçalho IP interno.

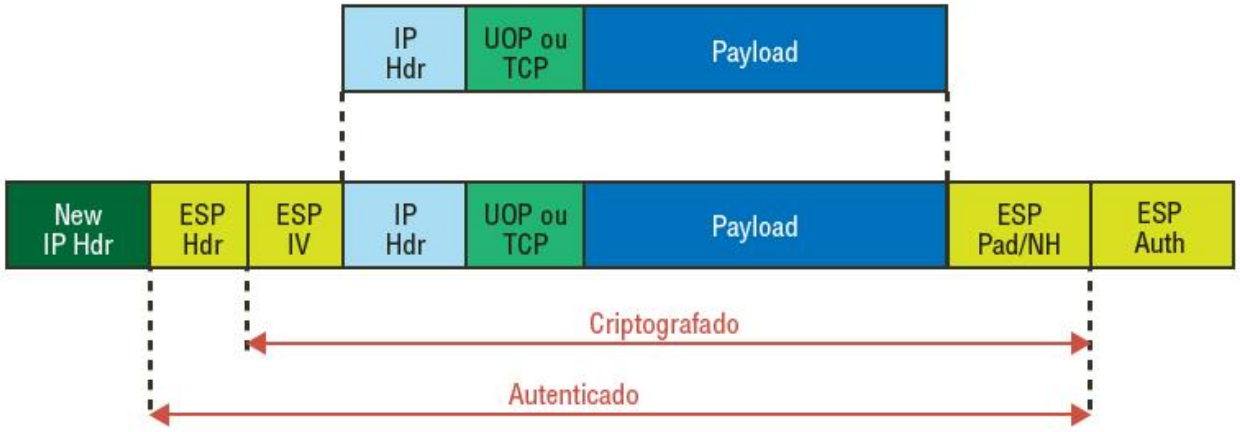



**Figura: Pacote encapsulado ESP no modo túnel.**
**Fonte: Internet 2015.**

Na figura acima o novo cabeçalho IP (New IP Hdr) e a autenticação ESP (ESP Auth) não são protegidos.

11

**3.1.5- Protocolos IPSec**

O IPSec oferece serviços de segurança na camada de IP. Permite que um sistema selecione protocolos de segurança exigidos e determine os algoritmos necessários para os serviços, ao colocar as chaves criptográficas exigidas para oferecer os serviços solicitados. Dois protocolos podem ser usados para oferecer segurança:

- **Autenticação do cabeçalho** ou **Authetication Header (AH) e**
- **Encapsulamento de Segurança do *Payload*** ou **Encapsulating Security Payload (ESP**), um protocolo combinado de criptografia e autenticação, designado pelo formato de pacote para esse protocolo.

Os serviços e suporte de cada protocolo IPSec estão listados na tabela a seguir.

**Tabela 1: Serviços de cada protocolo IPSec**

| | AH | ESP | ESP + AH |
|---|---|---|---|
| Controle de Acesso | Sim | Sim | Sim |
| Integridade sem conexão | Sim | x | Sim |
| Autenticação da origem | Sim | x | Sim |
| Rejeição de pacotes repetidos | sim | Sim | Sim |
| Confidencialidade | x | Sim | Sim |

**Fonte: Peixinho, 2013.**

O IPSec pode ser utilizado tanto para comunicação segura entre computadores (geralmente no modo transporte), quanto para o estabelecimento de VPN (geralmente no modo túnel).


Sistemas operacionais MS, a partir do Windows 2000, já possuem suporte nativo a IPSec, de modo que é possível que todo o tráfego entre servidores seja criptografado. Normalmente para utilizar IPSec, os roteadores presentes na rede devem suportar e entender o protocolo, para poderem encaminhar corretamente os dados.

**Autenticação do cabeçalho**

Estabelece mecanismos de verificação da autenticidade e integridade de pacotes IP. Normalmente, na verificação da autenticidade de pacotes, é calculado o Hash de HMAC (Hash Message Authentication Code) usando funções de Hash MD5 ou SHA-1.

**Encapsulamento de Segurança do *Payload***

Estabelece mecanismos de garantia da privacidade e integridade do conteúdo, utilizando técnicas de criptografia e código Hash, respectivamente. Na criptografia, normalmente são utilizados algoritmos DES, 3DES ou AES. Para o código Hash são utilizadas funções MD5 ou SHA-1.

12

### 3.1.6- Cabeçalho ESP

Observe a imagem:

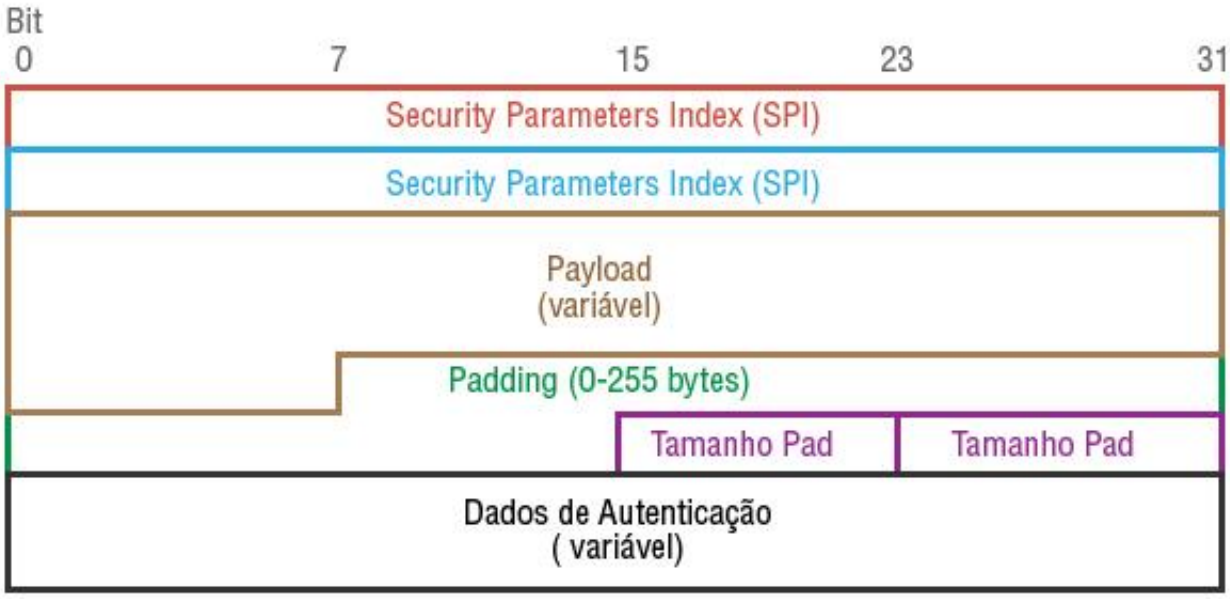



**Figura: Cabeçalho ESP.**
**Fonte: Internet, 2015.**

O campo **SPI** possui um valor que identifica a associação de segurança (SA) de um tráfego IPSec. O campo **Número de Sequência** possui um contador, que é incrementado a cada pacote enviado, com o objetivo de proteger contra ataques replay, no qual o atacante captura um tráfego e o repete mais à frente. O **Payload** contém o pacote original que está sendo protegido pelo ESP. O **Padding** é utilizado para completar os dados de modo a caber no tamanho de bloco do algoritmo de criptografia. **Tamanho Pad** contém o tamanho do campo anterior e **Next Header** indica o tipo do próximo cabeçalho.

13

### 3.1.7- Cabeçalho AH

No cabeçalho AH, alguns campos são invertidos em relação ao ESP, e não há a cifragem do pacote original.

Veja a imagem:

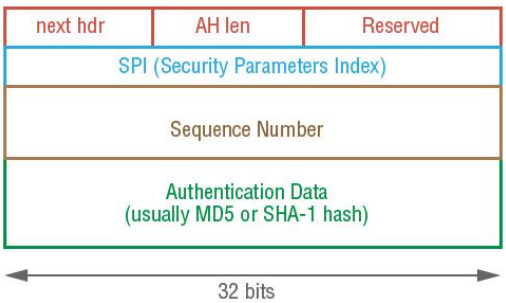



**Figura: Cabeçalho AH.**
**Fonte: Internet, 2015.**

Os campos **Next header**, **SPI** e **Sequence number** possuem a mesma finalidade dos correspondentes no cabeçalho ESP. O campo **AH length** indica o tamanho do cabeçalho AH.

14

## 4 – VPN SSL

Com o uso de VPNs baseadas em SSL, é possível ter acesso a aplicações ou redes remotas, tendo como acesso qualquer tipo de conectividade à internet.

A VPN SSL pode ser implementada via:

a) Cliente VPN SSL;
b) Navegador web e
c) Instalação simplificada do agente.

Quando se precisa de segurança apenas em uma aplicação específica, como navegação na internet, envio de correio eletrônico e mensagens instantâneas, utiliza-se a criptografia na comunicação entre essas aplicações. As escolhas mais populares de criptografia para esse cenário são **TLS** (Transport Layer Security: TLS 1.0 ou TLS1) e **SSL** (Security Sockets Layer: SSL 3.0 ou SSL 3.1).


Os dois protocolos têm a mesma finalidade, com pequenas diferenças entre eles. Ambos os protocolos suportam uma variedade de algoritmos de criptografia ou cifras para realizar algumas funções, como a autenticação do servidor e do cliente, transmissão de certificados e estabelecimento das chaves de sessão.

Para a **criptografia em massa dos dados**, são utilizados algoritmos simétricos. Algoritmos assimétricos são utilizados para **autenticação e troca de chaves**. O Hash é utilizado como parte do processo de autenticação.

Com o uso de VPNs baseadas em SSL, é possível ter acesso a aplicações ou redes remotas, tendo como acesso qualquer tipo de conectividade à internet, sendo necessário apenas um navegador da internet ou um *software* cliente instalado na máquina do usuário. Essa flexibilidade permite às VPNs baseadas em SSL prover acesso de qualquer lugar a recursos computacionais de uma empresa. Dessa forma, colaboradores de uma empresa podem utilizar VPNs baseadas em SSL para ter acesso remoto a aplicações de uma empresa.

15

Existem algumas etapas no estabelecimento da sessão VPN SSL que podem ser descritas em fases, conforme ilustra a figura a seguir:

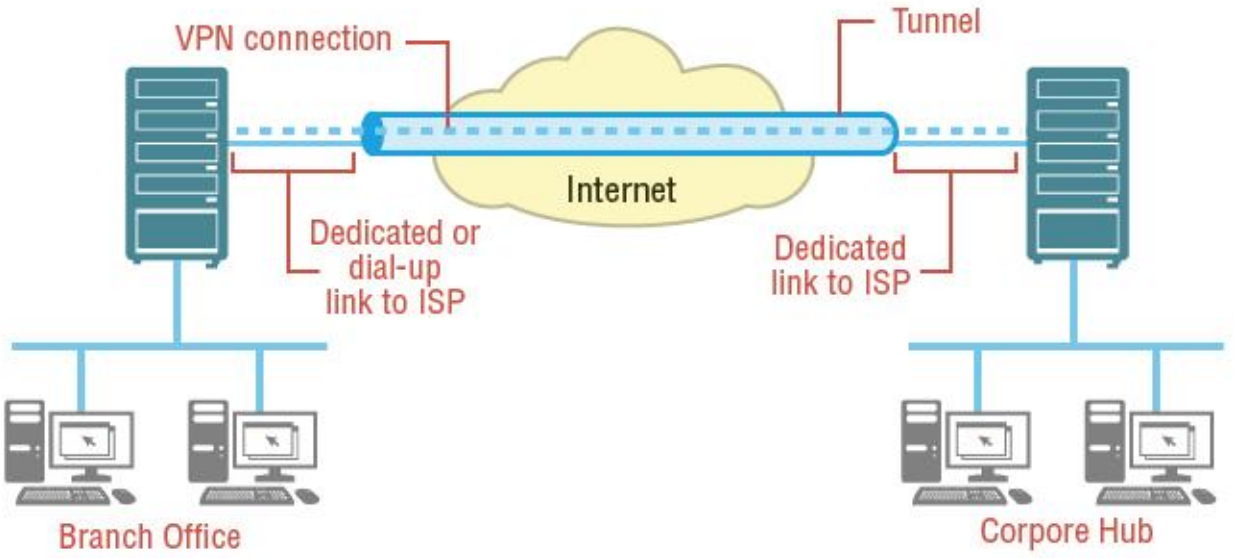



**Figura: Estabelecimento de uma sessão VPN SSL.**
**Fonte: Internet, 2015.**

O processo ocorre do seguinte modo:

a) Um equipamento usuário do Branch Office estabelece uma conexão TCP na porta 443 do servidor ISP.
b) O servidor SSL apresenta um certificado digital que contém a chave pública digitalmente assinada por uma Autoridade Certificadora confiável.
c) O computador do usuário gera uma chave simétrica compartilhada entre as duas partes, cliente e servidor.
d) A chave pública do servidor é utilizada para criptografar a chave compartilhada e transmitir para o cliente. O *software* do servidor utiliza a chave privada para descriptografar a chave compartilhada enviada pelo cliente. Assim que o servidor realizar esse processo, ambos terão acesso à chave compartilhada.
e) A chave compartilhada então é utilizada para criptografar os dados transmitidos na sessão SSL.


O OpenVPN é um exemplo de *software* livre, que utiliza SSL para criar túneis VPN. Uma vantagem das VPN SSL em relação ao IPSec é que a liberação do acesso através de um firewall é bem mais simples, pois envolve apenas uma porta (443 TCP), que normalmente já é liberada para acessos www seguros (HTTPS).

16

**4.1 - Implementação IPSec no Linux**

Existem várias implantações de VPN IPSec no mercado, variando de fabricantes de sistemas operacionais, fabricantes de dispositivos dedicados (appliances) e comunidades de *software* livre.

O **Openswan** é uma implentação de IPSec para Linux e é um projeto de muito sucesso na comunidade de *software* livre.

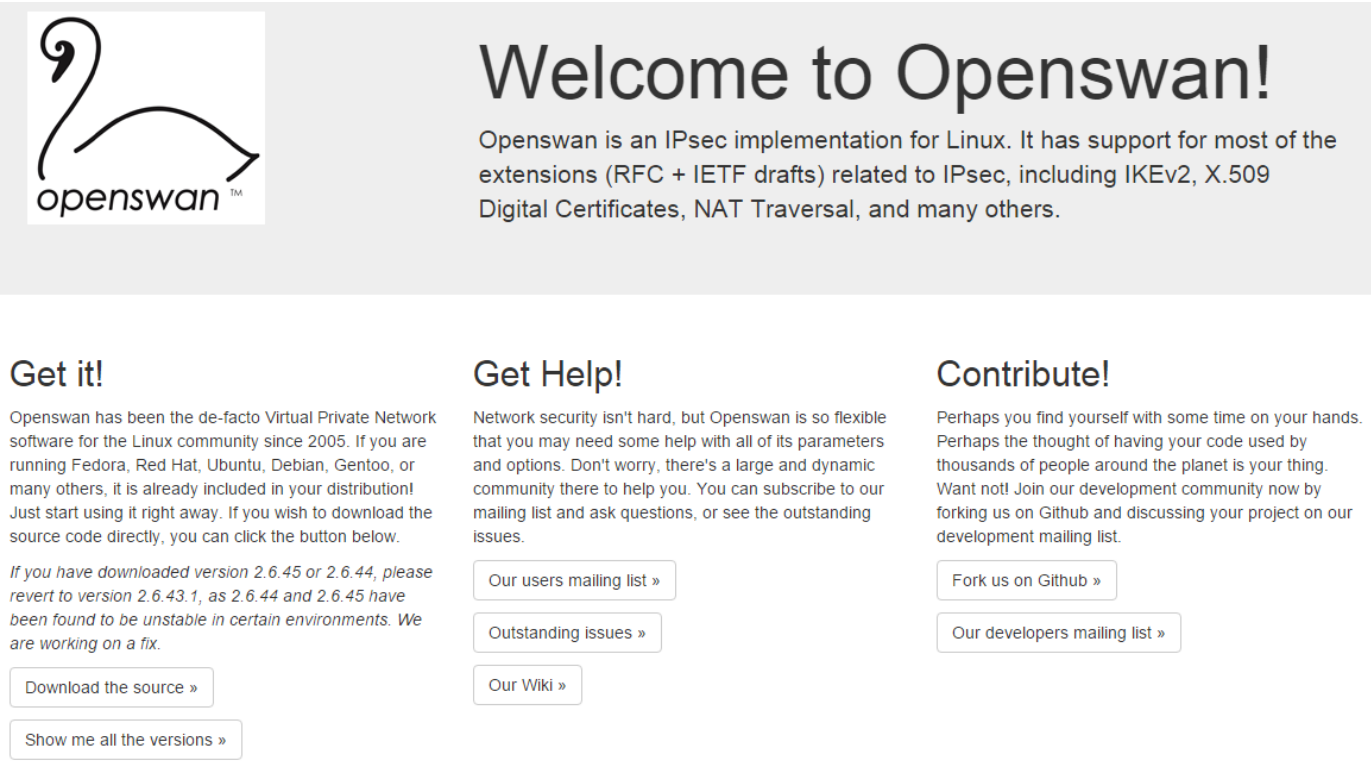



**Figura: www.openswan.org.**
**Fonte: Internet, 2015.**

O Openswan é uma derivação de um antigo projeto chamado Free S/Wan, que foi descontinuado.

**Openswan**

Para acessar o site do projeto, acesse www.openswan.org

17

### 4.2 - Instalação do Openswan

A instalação abaixo listada é de uma versão antiga, licenciada pela GPL versão 2. A versão indicada na figura anterior pode ter algumas diferenças, porém  não muito, da apresentada abaixo.

A instalação e configuração, da solução de VPN, foi feita no SO Debian.

```
debian:~# apt-get install openswan
Reading package lists... Done
Building dependency tree... Done
The following extra packages will be installed:
  ca-certificates iproute ipsec-tools libatm1 libcurl3 libgmp3c2 openssl
Suggested packages:
  openswan-modules-source linux-patch-openswan curl
Recommended packages:
  iproute-doc
The following NEW packages will be installed:
  ca-certificates iproute ipsec-tools libatm1 libcurl3 libgmp3c2 openssl
  openswan
0 upgraded, 8 newly installed, 0 to remove and 32 not upgraded.
Need to get 3782kB of archives.
After unpacking 10.5MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Ao ser apresentada a tela de configuração inicial logo após a instalação, a pergunta: “Do you want to create a RSA public/private keypair for this host?” foi respondido: “No”, pois foi utilizada uma Pre-shared Key (Chave pré-compartilhada). Após essa tela a instalação pode ser considerada como terminada.

Veja a seguir.

**18**

Verificando a versão:

```
debian:~# ipsec verify
Checking your system to see if IPsec got installed and started correctly:
Version check and ipsec on-path                                 [OK]
Linux Openswan U2.4.6/K2.6.18-5-k7 (netkey)
Checking for IPsec support in kernel                            [OK]
NETKEY detected, testing for disabled ICMP send_redirects       [FAILED]
 Please disable /proc/sys/net/ipv4/conf/*/send_redirects
  or NETKEY will cause the sending of bogus ICMP redirects!
NETKEY detected, testing for disabled ICMP accept_redirects     [FAILED]
  Please disable /proc/sys/net/ipv4/conf/*/accept_redirects
  or NETKEY will accept bogus ICMP redirects!
Checking for RSA private key (/etc/ipsec.secrets)               [DISABLED]
  ipsec showhostkey: no default key in “/etc/ipsec.secrets”
Checking that pluto is running                                  [OK]
Two or more interfaces found, checking IP forwarding            [FAILED]
Checking for ‘ip’ command                                       [OK]
Checking for ‘iptables’ command                                 [OK]
Opportunistic Encryption Support                                [DISABLED]
```

**4.3 - Configuração do Openswan**

O servidor de VPN deve estar configurado de acordo com a estrutura de sua unidade. Sempre lembrando que a instalação mostrada abaixo é de uma versão mais antiga, licenciada pela GPL versão 2. A versão indicada da figura anterior pode ter algumas diferenças.

Para a configuração da VPN siga os passos seguintes.

1. Habilitar o roteamento no Linux:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Configurar o arquivo de configuração do Openswan:

```
vim /etc/ipsec.conf
debian:~# cat /etc/ipsec.conf
# /etc/ipsec.conf - Openswan IPsec configuration file
# RCSID $Id: ipsec.conf.in,v 1.15.2.4 2006/07/11 16:17:53 paul Exp $
# This file:  /usr/share/doc/openswan/ipsec.conf-sample
#
# Manual:     ipsec.conf.5
version 2.0     # conforms to second version of ipsec.conf specification
# basic configuration
config setup
        interfaces=%defaultroute
        # plutodebug / klipsdebug = “all”, “none” or a combation from below:
        # “raw crypt parsing emitting control klips pfkey natt x509 private”
        # eg:
        # plutodebug=”control parsing”
        #
        # Only enable klipsdebug=all if you are a developer
        #
        # NAT-TRAVERSAL support, see README.NAT-Traversal
        nat_traversal=yes
        virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4:172.16.0.0/12
        #
        # enable this if you see “failed to find any available worker”
        nhelpers=0
# Add connections here
conn %default
        keyingtries=0
        disablearrivalcheck=no
conn vpnpeer1
        left=200.200.40.16
```

```
      leftsubnet=192.168.1.0/255.255.255.0
      right=200.200.140.10
      rightsubnet=10.61.0.0/255.255.0.0
      ike=aes256-sha-modp1024
      esp=aes256-sha1
      pfs=no
      ikelifetime=8h
  keylife=8h
      dpddelay=30
      dpdtimeout=120
      dpdaction=hold
      authby=secret
      auto=start
# nameserver 202.21.11.100
# sample VPN connections, see /etc/ipsec.d/examples/
#Disable Opportunistic Encryption
include /etc/ipsec.d/examples/no_oe.conf
----------------------------------------------------------------------
debian:~# cat /etc/ipsec.secrets
# RCSID $Id: ipsec.secrets.proto,v 1.3.6.1 2005/09/28 13:59:14 paul Exp $
# This file holds shared secrets or RSA private keys for inter-Pluto
# authentication.  See ipsec_pluto(8) manpage, and HTML documentation.
# RSA private key for this host, authenticating it to any other host
# which knows the public part.  Suitable public keys, for ipsec. conf, DNS,
# or configuration of other implementations, can be extracted conveniently
# with “ipsec showhostkey”.
200.200.40.16 200.200.140.10 : PSK “vpnipsec”
```

**20**

**4.4 - Implementação de VPN SSL no Linux**

O uso de VPN SSL está em expansão no cenário atual de Tecnologias da Informação e Comunicação (TIC), uma vez que a sua implementação costuma ser mais simples do que as implementações de VPN IPSec.

O projeto OpenVPN possui utilização destacada pela comunidade de *software* livre. É importante lembrar que o IPv6 já traz o IPSec nativo, de modo que é preciso conhecer bem as duas tecnologias.

**21**

**4.5 - Instalação do OpenVPN**

O OpenVPN faz parte do repositório padrão do Debian. Lembr-se sempre que a instalação abaixo é de uma versão mais antiga, licenciada pela GPL versão 2.

A instalação pode ser feita com o utilitário apt-get, como ilustra o exemplo a seguir:

```
debian:~# apt-get update
debian:~# apt-get install openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  liblzo2-2 libpkcs11-helper1 openssl openssl-blacklist openvpn-blacklist
Suggested packages:
  ca-certificates resolvconf
The following NEW packages will be installed:
  liblzo2-2 libpkcs11-helper1 openssl openssl-blacklist openvpn openvpn-blacklist
0 upgraded, 6 newly installed, 0 to remove and 14 not upgraded.
Need to get 8948kB of archives.
After this operation, 18.5MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://linorg.usp.br lenny/main openssl 0.9.8g-15+lenny7 [1034kB]
Get:2 http://linorg.usp.br lenny/main openssl-blacklist 0.4.2 [6338kB]
Get:3 http://linorg.usp.br lenny/main liblzo2-2 2.03-1 [61.5kB]
Get:4 http://linorg.usp.br lenny/main libpkcs11-helper1 1.05-1 [42.4kB]
Get:5 http://linorg.usp.br lenny/main openvpn-blacklist 0.3 [1068kB]
Get:6 http://linorg.usp.br lenny/main openvpn 2.1~rc11-1 [404kB]
Fetched 8948kB in 2min20s (63.6kB/s)
Preconfiguring packages ...
tar: ./conffiles: time stamp 2010-06-08 17:45:39 is 5089826.308233209 s in the future
tar: ./postinst: time stamp 2010-06-08 17:45:39 is 5089826.262979826 s in the future
tar: ./control: time stamp 2010-06-08 17:45:38 is 5089825.261959074 s in the future
tar: ./md5sums: time stamp 2010-06-08 17:45:39 is 5089826.261059126 s in the future
tar: .: time stamp 2010-06-08 17:45:39 is 5089826.260466338 s in the future
Selecting previously deselected package openssl.
(Reading database ... 19366 files and directories currently installed.)
Unpacking openssl (from .../openssl_0.9.8g-15+lenny7_i386.deb) ...
Selecting previously deselected package openssl-blacklist.
Unpacking openssl-blacklist (from .../openssl-blacklist_0.4.2_all.deb) ...
Selecting previously deselected package liblzo2-2.
Unpacking liblzo2-2 (from .../liblzo2-2_2.03-1_i386.deb) ...
Selecting previously deselected package libpkcs11-helper1.
Unpacking libpkcs11-helper1 (from .../libpkcs11-helper1_1.05-1_i386. deb) ...
Selecting previously deselected package openvpn-blacklist.
Unpacking openvpn-blacklist (from .../openvpn-blacklist_0.3_all.deb) ...
Selecting previously deselected package openvpn.
Unpacking openvpn (from .../openvpn_2.1~rc11-1_i386.deb) ...
Processing triggers for man-db ...
Setting up openssl (0.9.8g-15+lenny7) ...
Setting up openssl-blacklist (0.4.2) ...
```

```
Setting up liblzo2-2 (2.03-1) ...
Setting up libpkcs11-helper1 (1.05-1) ...
Setting up openvpn-blacklist (0.3) ...
Setting up openvpn (2.1~rc11-1) ...
Restarting virtual private network daemon.:.
debian:~#
```

O projeto disponibiliza ainda, no seu site, um cliente de VPN SSL para as plataformas Microsoft Windows, Apple Mac OS X e Linux. A instalação desses clientes para Microsoft Windows é realizada pelo método de instalação padrão de aplicativos para Windows.

22

**4.6 - Configuração do OpenVPN**

A configuração do OpenVPN pode ser realizada com a edição do arquivo de configuração padrão ou pela criação de um novo arquivo de configuração. No caso de criar um novo arquivo, é necessário fazer referência a esse novo arquivo de configuração na inicialização do serviço.

Clique aqui para ver um exemplo de configuração do servidor OpenVPN.

Experimente instalar em sua máquina e verifique os resultados.


É importante ressaltar que a VPN pode trabalhar em modo transparente (bridge) ou roteada (routing). Desta forma, será necessário configurar o kernel do Linux para que trabalhe de acordo com um dos modos. Será necessário também ajustar as regras de firewall para permitir o fluxo de pacotes para a interface virtual do OpenVPN.

Aqui termina nosso estudo sobre as ferramentas. No próximo módulo entraremos na auditoria de Segurança da informação.

**Clique aqui**

Exemplo de configuração do servidor OpenVPN

```
;local a.b.c.d
port 1194
;proto tcp
proto udp
;dev tap
dev tun
;dev-node MyTap
ca ca.crt
```

```
cert server.crt
dh dh1024.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100
;server-bridge
;push “route 192.168.10.0 255.255.255.0”
;push “route 192.168.20.0 255.255.255.0”
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
;learn-address ./script
;push “redirect-gateway def1 bypass-dhcp”
;push “dhcp-option DNS 208.67.222.222”
;push “dhcp-option DNS 208.67.220.220”
;client-to-client
;duplicate-cn
keepalive 10 120
comp-lzo
;max-clients 100
;user nobody
;group nogroup
persist-key
persist-tun
status openvpn-status.log
;log         openvpn.log
;log-append  openvpn.log
verb 3
;mute 20
```

23

## 5 - RESUMO

Neste módulo, foram abordados os aspectos teóricos do uso de VPN, IPSec e VPN SSL, além de abordarmos aspectos práticos dessas aplicações.
VPN (Virtual Private Network) é muito utilizada atualmente. A possibilidade de uso de uma rede pública como a internet para interligar escritórios comerciais e grandes empresas tem permitido a redução de custos e viabiliza negócios que têm como premissa requisitos de comunicação eficiente.

Foi visto que uma solução efetiva de VPN visa transportar os dados de modo seguro e sigiloso, usando um canal compartilhado para interligar duas redes privadas protegidas e para que isso ocorra é necessário alcançar quatro objetivos importantes:

a) Confidencialidade dos dados: garantia de que a mensagem não poderá ser interpretada por origens não autorizadas;
b) Integridade dos dados: garantia de que o conteúdo da mensagem não foi alterado durante a transmissão entre o emissor e o receptor;
c) Não repúdio do emissor: o emissor não poderá repudiar o envio da mensagem, ou seja, dizer que ele não enviou a mensagem questionada, com embasamento legal;
d) Autenticação da mensagem: garantia de que a mensagem foi enviada por uma fonte autêntica e será entregue a um destino autêntico.

Outra segurança importante é o IPSec, que é a capacidade que pode ser acrescentada a qualquer versão atual do protocolo Internet (IPv4 e IPv6) por meio de cabeçalhos adicionais. Ele é um conjunto de protocolos, também conhecido como suíte de segurança IP. Os protocolos inclusos na suíte de segurança IP estão focados em:

a) Entrega da mensagem autêntica.
b) Integridade dos dados.
c) Confidencialidade dos dados.
d) Não repúdio do emissor.

**24**

Outro conceito importante é que com o uso de VPNs baseadas em SSL, é possível ter acesso a aplicações ou redes remotas, tendo como acesso qualquer tipo de conectividade à internet. Pode ser implementada via:

a) Cliente VPN SSL;
b) Navegador web e
c) Instalação simplificada do agente.

Com o uso de VPNs baseadas em SSL, é possível ter acesso a aplicações ou redes remotas, tendo como acesso qualquer tipo de conectividade à internet, sendo necessário apenas um navegador da internet ou um *software* cliente instalado na máquina do usuário. Essa flexibilidade permite às VPNs baseadas em SSL prover acesso de qualquer lugar a recursos computacionais de uma empresa. Dessa forma, colaboradores de uma empresa podem utilizar VPNs baseadas em SSL para ter acesso remoto a aplicações de uma empresa.

# UNIDADE 4 – REDES PRIVADAS VIRTUAIS, AUDITORIA DE SEGURANÇA DA INFORMAÇÃO E CONFIGURAÇÕES DE SERVIDORES
# MÓDULO 2 – AUDITORIA DE SEGURANÇA DA INFORMAÇÃO

01

## 1 – O QUE É AUDITORIA?

Pode-se definir auditoria como a **medição de algo contra um padrão**. Apesar de tratarmos de Segurança da Informação, o conceito de auditoria pode ser aplicado em qualquer área, como qualidade, ambiental, financeira, de conformidade e outros.

Ao tratar especificamente de **auditoria de SI (segurança da informação)**, estamos a auditar o cumprimento de uma política de segurança, a eficácia de um novo sistema de segurança (como um *firewall*), se um sistema está com todas as correções conhecidas aplicadas, entre outros.

Neste momento trataremos especificamente de auditoria de segurança, utilizando a ferramenta **Nmap** e técnicas para verificar se as implementações de segurança realizadas nos módulos anteriores estão provendo o nível de segurança especificado. Entre as técnicas utilizadas em auditorias, as mais comuns são:

- análise de vulnerabilidades e
- testes de penetração (*penetration testing* ou *pentest*).

É importante ressaltar que este módulo trata apenas da **auditoria de dispositivos de segurança**, sem entrar em questões de políticas, análise de risco e outros tópicos relacionados à governança e normatização.

02

## 2 - ANÁLISE DE VULNERABILIDADES

Uma vulnerabilidade pode ser definida como uma **brecha em um sistema computacional**. Quando tratamos de programas (*software*), essas vulnerabilidades são muitas vezes chamadas de bugs (bug=falha ou vulnerabilidade em um programa ou sistema).

Um sistema vulnerável pode ser um *software*, um sistema operacional, um roteador, um protocolo ou até um *hardware*.

Essas vulnerabilidades podem ser exploradas com o intuito de subverter o sistema em questão, causando indisponibilidade, obtendo controle sobre ele, acessando dados sensíveis ou utilizando o sistema para atacar outros sistemas.

Em consequência, as **vulnerabilidades** podem ser classificadas como sendo:

a) Falha em um sistema computacional;
b) Bugs (*software*);
c) Falha de configuração

Os sistemas vulneráveis podem ocorrer em:

a) Aplicativos (ferramentas);
b) Sistema operacional (SO);
c) Roteador (dispositivo);
d) Protocolo (configuração);
e) *Hardware* (máquinas e conexões);

03

Existem diversos **tipos de vulnerabilidades**, sendo os mais comuns:

- as **vulnerabilidades de *software***, causadas muitas vezes por validação insuficiente dos parâmetros recebidos, e
- as **vulnerabilidades em protocolos ou serviços**.

Essas vulnerabilidades podem levar a:

a) estouros de pilha (*buffer overflow*),
b) negação de serviços (Denial of Service);
c) acesso irrestrito ao sistema vulnerável.

Elas são descobertas por pesquisadores, que podem ser da própria empresa que fabrica o produto ou pesquisadores independentes, que costumam notificar as empresas sobre a falha para que elas possam lançar correções antes da divulgação pública.


Às vezes, muitos administradores não aplicam as correções de segurança dos fabricantes nos sistemas sob sua administração, de modo que estes ficam vulneráveis a falhas conhecidas e amplamente divulgadas.

Uma forma eficiente de verificar se uma rede, aplicação ou sistema operacional está suscetível a determinadas falhas é com o **uso de ferramentas de análise de vulnerabilidades**. Essas ferramentas utilizam assinaturas ou regras que simulam falhas conhecidas e produzem um relatório com os problemas encontrados e possíveis soluções.

**negação de serviços**

A negação de serviço pode ser:

- DoS-Denial of Service, que é a negação ou indisponibilidade de um serviço causada por um ataque e
- DDoS-Distributed Denial of Service, que é o ataque de negação de serviço realizado de forma distribuída e coordenada).

04


É importante salientar que a análise de vulnerabilidade não substitui o controle da aplicação de correções dos fabricantes dos produtos utilizados em uma organização, pois confiar na ferramenta pode levar à não aplicação de uma correção caso ela esteja desatualizada ou mesmo não tenha sido atualizada para verificar uma vulnerabilidade específica.

Como novas falhas são encontradas todos os dias, uma boa ferramenta de análise de vulnerabilidades deve ser constantemente atualizada, de modo que possa detectar as falhas mais recentes descobertas.

Existe atualmente uma série de ferramentas de análise de vulnerabilidades, gratuitas e comerciais. Algumas das **ferramentas gratuitas/*open source*** são:

- Nmap,
- OpenVas,
- Microsoft MBSA.

Das **ferramentas comerciais** destacamos:

- Rapid7 NeXpose,
- eEye Retina,
- GFI LANguard,
- IBM Internet Scanner.

A seguir, detalharemos o uso da ferramenta Nmap (Network Mapper), que tem fins não comerciais e pode ser obtida livremente na internet.

05

## 2.1 – Instalação e características do Nmap

### 2.1.1 – Instalação do Nmap

Vale lembrar que o Nmap (*Network Mapper*) é uma ferramenta de segurança usada para detectar computadores e serviços numa rede, criando um "mapa" dessa mesma rede.

A instalação do Nmap no ambiente Windows já foi mostrada anteriormente em nossa disciplina, de modo que não repetiremos o processo neste momento.

No ambiente Linux, podelos instalar a ferramenta no servidor:

a) No RedHat/CentOS: **#yum install nmap**
b) No Debian/Ubuntu: **#apt-get install nmap**

06

### 2.1.2 – Características do Nmap

Conforme o site https://nmap.org/man/pt_BR/, o Nmap normalmente é utilizado para auditorias de segurança. É uma ferramenta de código aberto para exploração de rede e foi desenhada para escanear rapidamente redes amplas, mas funciona muito bem contra hosts individuais.

O Nmap utiliza pacotes IP para determinar:

- quais hosts estão disponíveis na rede,
- quais serviços (nome da aplicação e versão) os hosts oferecem,
- quais sistemas operacionais (e versões de SO) eles estão executando,
- que tipos de filtro de pacotes/*firewall*s estão em uso, e dezenas de outras características.

A saída do Nmap é uma lista de alvos escaneados, com informações adicionais de cada um dependendo das opções utilizadas. Uma informação chave é a "tabela de portas interessantes". Essa tabela lista o número da porta e o protocolo, o nome do serviço e o estado. O estado pode ser:

a) aberto (open)
b) filtrado (filtered)
c) fechado (closed)
d) não-filtrado (unfiltered)

O Nmap reporta as combinações aberta|filtrada (open|filtered) e fechada|filtrada (closed|filtered) quando não consegue determinar qual dos dois estados descreve melhor a porta. A tabela de portas também pode incluir detalhes de versão de *software* quando a detecção de versão for solicitada.

Quando um scan do protocolo IP é solicitado (-sO), o Nmap fornece informações dos protocolos IP suportados ao invés de portas que estejam abertas.

| |
|---|
| **aberto (open)**<br><br>Significa que uma aplicação na máquina-alvo está escutando as conexões/pacotes naquela porta. |
| **filtrado (filtered)**<br><br>Significa que o *firewall*, filtro ou outro obstáculo de rede está bloqueando a porta de forma que o Nmap não consegue dizer se ela está aberta (open) ou fechada (closed). |
| **fechado (closed)**<br><br>Não possuem uma aplicação escutando nelas, embora possam abrir a qualquer instante. |
| **não-filtrado (unfiltered)**<br><br>Quando elas respondem às sondagens do Nmap, mas o Nmap não consegue determinar se as portas estão abertas ou fechadas. |

07

**2.2 - Auditoria com o Nmap**

Após a instalação, feita conforme item anterior, deverá ser criada a política de segurança e da varredura. Após essas ações, pode-se iniciar a varredura.

Como já informamos anteriormente, esta ferramenta utiliza inúmeras técnicas, vamos mostrar alguns comandos como exemplo:

a) Escanear um host ou endereço ip.

```
#nmap 192.168.100.5
```

b) Escanear mais de um host:

```
#nmap 192.168.100.5 192.168.100.10
```

c) Escanear um range de endereços:

```
#nmap 192.168.100.5-10
```

d) Escanear uma sub-rede:

```
#nmap 192.168.100.0/24
```

e) Escanear uma sub-rede excluindo um host:

```
#nmap 192.168.100.0/24 --exclude 192.168.100.5
#nmap 192.168.100.0/24 --exclude 192.168.100.5 192.168.100.10
```

f) Identificando qual é a versão do Sistema Operacional:

```
#nmap -v -A 192.168.100.5
```

g) Identificando serviços remotos e sua versão:

```
#nmap -sV 192.168.100.5
```

h) Realizar análise com endereço mac camuflado:

```
#nmap --spoof-mac ENDREÇO-MAC-AQUI 192.168.100.5
```

i) Realizar análise com endereço mac camuflado aleatório:

```
#nmap --spoof-mac 0 192.168.100.5
```

j) Salvando as informações obtidas em um arquivo texto:

```
#nmap -sV 192.168.100.5 > nmap.txt
```

08

**2.3 - Iniciando uma varredura**

Abaixo, apresenta-se um exemplo de uma varredura simples. Tomou-se como exemplo o IP 192.168.1.0/24, com o Nmap, neste caso com a versão 6.49BETA4, da figura a seguir.

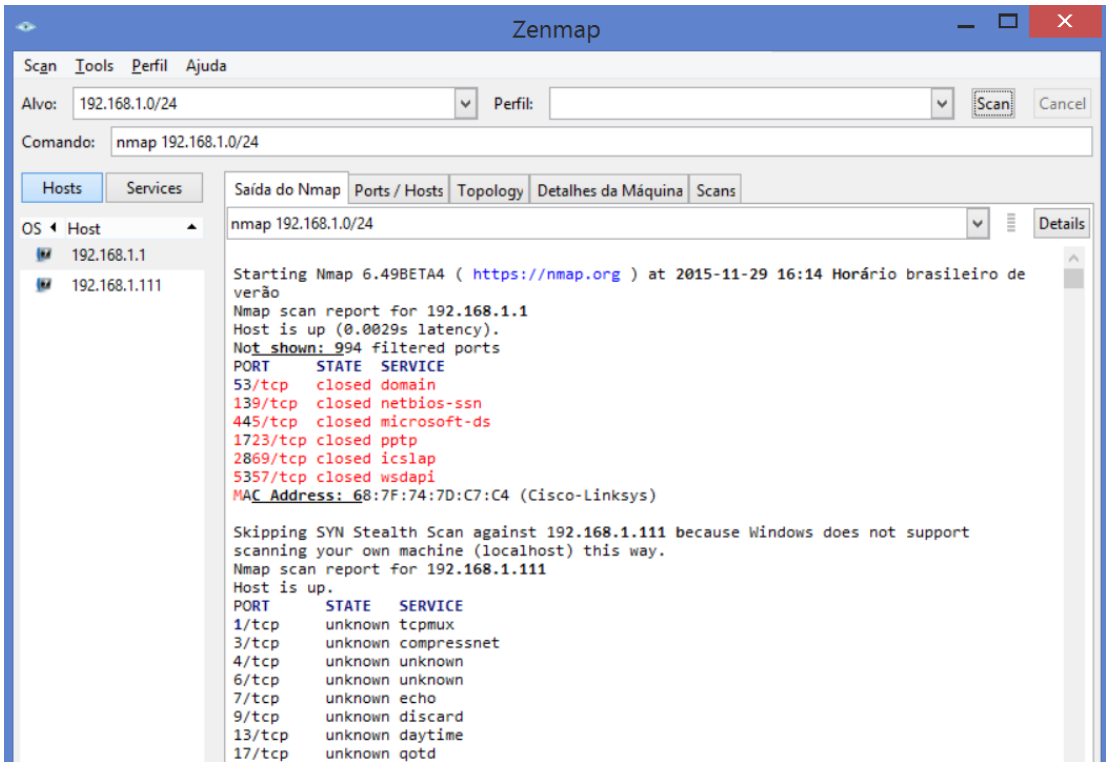



**Figura 1: Exemplo de varredura.**
**Fonte: O Autor, 2015.**

Verifique as portas 53, 139, 445, 1723, 2869, 5357 todas fechadas. Procure em http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?&page=53 a porta 2869 corresponde a icslap/udp. Essa consulta foi feita em 29/Nov/2015.

09

## 3 - EXEMPLO DE AUDITORIA EM FIREWALL

Os únicos argumentos que o Nmap utiliza nesse exemplo são -A, para habilitar a detecção de SO e a versão, -T4 para execução mais rápida, e os hostnames de dois alvos.

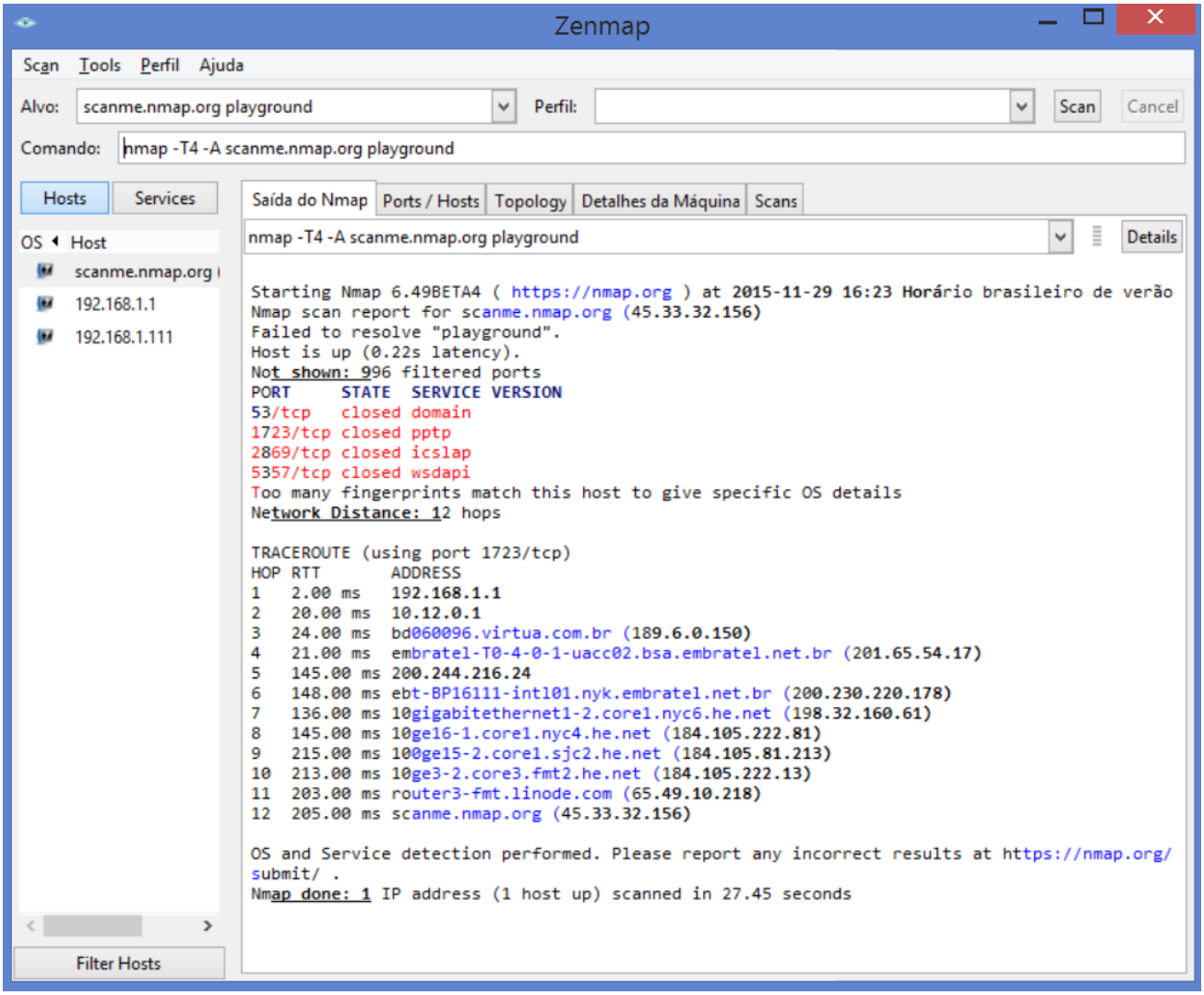



**Figura 2: Exemplo de auditoria em *Firewall*.**
**Fonte: O Autor, 2015.**

10

### 3.1 - Arquitetura do *firewall*

Neste ponto da auditoria, estamos preocupados se a arquitetura definida realmente cumpre os objetivos que foram definidos.

O auditor deve verificar as seguintes questões, entre outras:

a) Diferentes redes ligadas ao *firewall* estão fisicamente separadas?
b) Existem hubs (Hubs podem ter todo o tráfego que passa por ele monitorado) sendo usados na rede?
c) Como o *firewall* está controlando o fluxo de informação?
d) O diagrama lógico do perímetro está correto?
e) A segmentação realizada é suficiente?
f) Deve-se adicionar ou remover um *firewall*?
g) Deve-se adicionar ou remover interfaces de rede?

11

### 3.2 - Testando o *firewall*

Existem duas categorias diferentes de *firewall*:

a) os que **rodam sobre de um sistema operacional** e
b) os chamados ***appliances***, que são equipamentos específicos que fazem o papel de *firewall* (ex.: um switch com *firewall* embutido, um roteador com filtros de pacotes).

Cada tipo tem suas vantagens e desvantagens:

| Baseados em sistemas operacionais | *Appliance* |
|---|---|
| • são mais flexíveis, porém são suscetíveis a vulnerabilidades no sistema operacional usado. | • são normalmente mais seguros “de fábrica”, porém normalmente são proprietários e será preciso confiar no fabricante no que tange à segurança. |

Neste ponto da auditoria, as seguintes questões são importantes:

a) Quais serviços estão executando no *firewall*?
b) Eles são necessários?
c) Eles são seguros?

d) Existem correções de segurança que podem ser aplicadas no appliance ou no sistema operacional?
e) Quais as recomendações básicas de configuração do fabricante?
f) Elas foram aplicadas?
g) Existem acessos de administrador ao *firewall*?
h) Eles estão com o mínimo de permissão possível?
i) Uma ferramenta de auditoria relata algum problema com o *firewall*?
j) Existem recursos de segurança específicos para a plataforma que está sendo auditada?
k) Eles estão bem configurados?

**Appliance**

Serviço que é executado dentro de um *hardware* dedicado e otimizado para a aplicação em questão.

12

### 3.3 - Testando as regras do *firewall*

As bases de regras de um *firewall* costumam crescer com o tempo, por conta de solicitações de inclusão de novos servidores e novos serviços oferecidos na rede, e também de conexões com novas redes. Após alguns meses de manutenção das regras de um *firewall*, elas podem se tornar bastante complexas. Essa complexidade pode esconder regras redundantes, isto é, regras temporárias que nunca foram removidas ou até mesmo regras incorretas que ficaram esquecidas.


O auditor deve analisar as regras do *firewall*, de modo a encontrar e eliminar essas inconsistências, além de procurar simplificar as regras para facilitar uma visualização futura. A ideia final é minimizar ao máximo a quantidade de regras. Essa redução não só tornará o seu *firewall* mais simples, como mais rápido, visto que terá menos regras para processar.

A seguir algumas considerações que devem ser avaliadas pelo auditor:

a) Eliminar regras desnecessárias;
b) Combinar regras repetitivas;
c) Eliminar regras não autorizadas;
d) Terminar com o mínimo possível de regras;
e) Documentar as regras;
f) Verificar regras que realizam registros de acesso:
    1) somente registrar o necessário e
    2) registros excessivos podem ocupar muito espaço e diminuir o desempenho;

g) Verificar a existência de regras de bloqueio padrão;
h) Verificar se as regras são específicas;
i) Princípio do menor privilégio;

Utilizar ferramentas de varredura para validar as regras, como Nmap.

13

O Nmap suporta diversos tipos de varredura: S (SYN), T (Connect), A (ACK), W (Window), M (Maimon), U (UDP), N (Null), F (FIN), X (Xmas), I (Idle), Y (SCTP), O (IP protocol).

Alguns parâmetros interessantes do Nmap:

a) **–O**
b) **-PO**
c) **–v**
d) **-s<tipo>**

Um exercício interessante é realizar scans utilizando diversos tipos diferentes e verificar o tipo de registro que aparece no servidor remoto. A seguir outro exemplo do Nmap, agora utilizando a opção "–O", que procura adivinhar a versão do sistema operacional do sistema-destino:

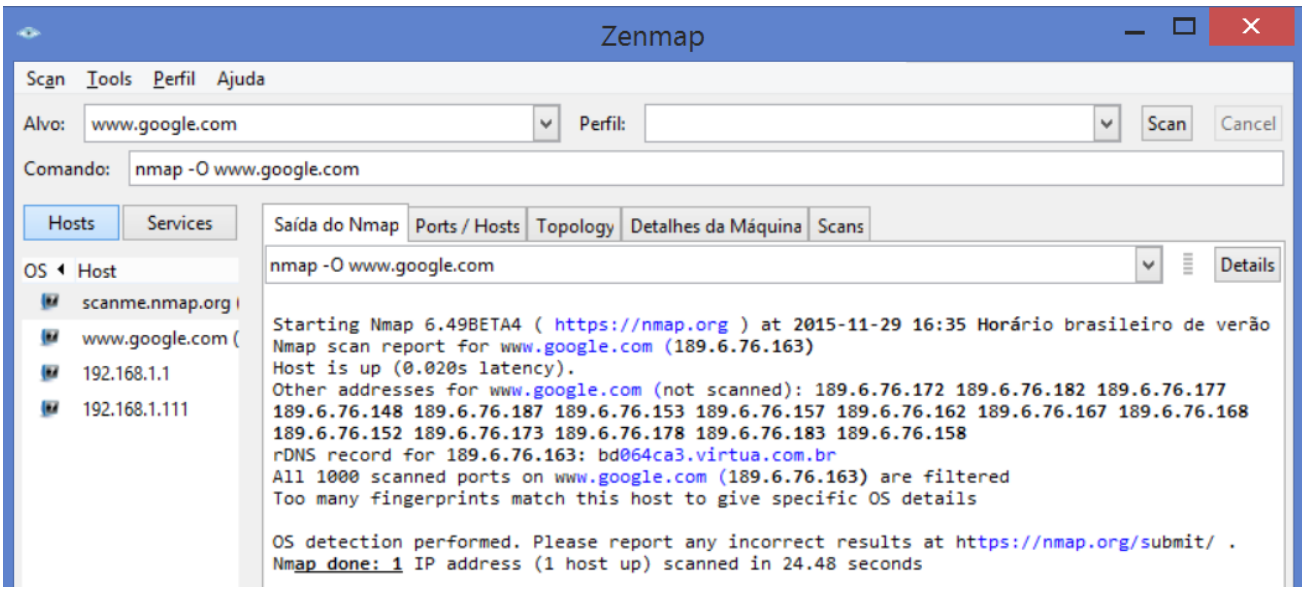



**Figura 3: Tentativa de verficar o SO.**
**Fonte: O Autor, 2015.**

**–O**

Realiza uma tentativa de detectar o sistema operacional da máquina analisada.

| |
|---|
| **-PO**<br><br>Realiza a varredura da máquina, mesmo que ela não responda ao ping. Útil em servidores que estão sendo filtrados por *firewall*s. |
| **–v**<br><br>Aumenta a quantidade de informação apresentada. |
| **-s<tipo>**<br><br>Tipo de varredura utilizada. Algumas varreduras procuram evitar que o sistema destino registre as tentativas de acesso. |

14

Verifique que o Nmap tentou detectar o sistema operacional, porém foi com insucesso devido ao servidor do Google estar protegido (todas as 1000 portas examinadas estão filtradas).

Outra ferramenta interessante para testar configuração de *firewall* é a ferramenta Netcat. Ela é conhecida como o "canivete suíço" das redes, devido à sua versatilidade. Imagine que se deseja testar a porta 3500 TCP se disponível em um determinado servidor, porém não se tem nenhum serviço em execução nesta porta. Com o Netcat, pode-se registrar um serviço nesta porta:

```
root# nc -l 3500
```

Em outra estação, pode-se realizar uma conexão com o servidor, na porta 3500:

```
root# nc 192.168.1.6 3500
```

Dessa forma, caso o tráfego esteja permitido, tudo o que escrevermos na estação será apresentado no servidor.

15

**3.4 - Alertas e registros**

Registros e alertas são itens importantes em uma política de segurança, mas, se eles não forem vistos periodicamente pela equipe responsável, de nada adiantam. *Firewall*s com muitos registros sendo gerados podem ser facilmente esquecidos pelo administrador, que fica perdido entre tantos dados.

Alertas podem ser configurados para envio por e-mail ou SMS, de modo que possam ser mais facilmente vistos pelo administrador. Revisar os registros periodicamente pode ser útil para detectar tentativas de ataque e permitir aos responsáveis a tomada de ações proativas.

Todas as ferramentas indicadas podem ser instaladas a partir do apt-get do Debian ou baixadas do site de cada ferramenta (Netcat, Hping, Nmap). Algumas delas também possuem versões para Windows, Mac OS X e outras plataformas.

A seguir, listamos algumas recomendações que devem ser observadas pelo auditor de segurança da rede:

a) Os registros de log estão precisos?
b) Estão sendo gerados mais registros do que o necessário?
c) Existe procedimento para analisar os alertas?
d) Eles são enviados para um local de rápida verificação?
e) Os registros estão em local seguro?
f) O horário do *firewall* está correto?
**g)** Ele está sendo sincronizado com uma fonte de tempo confiável?

Por fim, verificamos que a tarefa de auditoria não é uma tarefa simples. Apesar de existirem ferramentas que auxiliam o auditor em algumas questões, elas não resolvem todos os problemas. Bom senso e conhecimento ainda são fundamentais. Durante as atividades práticas, vamos exercitar o uso do Nmap, e teremos oportunidade de utilizar as demais ferramentas apresentadas neste módulo.

16

## 4 - RESUMO

Neste módulo vimos como realizar uma auditoria com a ferramenta Nmap, diferenciar análise de vulnerabilidades de testes de penetração e estudar conceitos relacionados à auditoria de Segurança de Informação.

A auditoria foi conceituada como uma medição de algo contra um padrão. Esse conceito de auditoria pode ser aplicado em qualquer área, como qualidade, ambiental, financeira, de conformidade e outros.

A auditoria se inicia com a análise de vulnerabilidade, que pode ser definida como uma brecha em um sistema computacional. Quando tratamos de programas (*software*), essas vulnerabilidades são muitas vezes chamadas de bugs (bug=falha ou vulnerabilidade em um programa ou sistema). Um sistema vulnerável pode ser um *software*, um sistema operacional, um roteador, um protocolo ou até um *hardware*.

As vulnerabilidades podem ser classificadas como: falha em um sistema computacional, bugs (*software*) e falha de configuração.

As vulnerabilidades podem levar a: a) estouros de pilha (buffer overflow), b) negação de serviços (DoS-Denial of Service que é a negação ou indisponibilidade de um serviço causada por um ataque e DDoS-Distributed Denial of Service que é o ataque de negação de serviço realizado de forma distribuída e coordenada) e c) acesso irrestrito ao sistema vulnerável.

Conforme o site https://nmap.org/man/pt_BR/, o Nmap (“Network Mapper”) é uma ferramenta de código aberto para exploração de rede e auditoria de segurança. Desenhada para escanear rapidamente redes amplas, funciona muito bem contra hosts individuais.

O auditor deve analisar as regras do *firewall*, de modo a encontrar e eliminar as inconsistências, além de procurar simplificar as regras para facilitar uma visualização futura. A ideia final é minimizar ao máximo a quantidade de regras. Essa redução não só tornará o seu *firewall* mais simples, como mais rápido, visto que terá menos regras para processar.

Considerações que devem ser avaliadas pelo auditor:

a) Eliminar regras desnecessárias;
b) Combinar regras repetitivas;
c) Eliminar regras não autorizadas;
d) Terminar com o mínimo possível de regras;
e) Documentar as regras;
f) Verificar regras que realizam registros de acesso: 1) somente registrar o necessário e 2) registros excessivos podem ocupar muito espaço e diminuir o desempenho;
g) Verificar a existência de regras de bloqueio padrão;
h) Verificar se as regras são específicas;
i) Princípio do menor privilégio;
j) Utilizar ferramentas de varredura para validar as regras, como Nmap.

# UNIDADE 4 – REDES PRIVADAS VIRTUAIS, AUDITORIA DE SEGURANÇA DA INFORMAÇÃO E CONFIGURAÇÕES DE SERVIDORES
# MÓDULO 3 – CONFIGURAÇÃO DE SERVIDORES WINDOWS

01

## 1 – NECESSIDADE DE CONFIGURAÇÃO DE UM BASTION *HOST*

Prevenir acesso não autorizado a dados sensíveis é essencial em qualquer ambiente em que múltiplos usuários têm acesso aos recursos físicos ou via rede. Um sistema operacional deve ser configurado de forma segura antes de ser exposto em uma rede pública não controlada, como o caso da internet. Este processo de reforçar a segurança é chamado de "*hardening*".


Apesar de muito importante, o administrador de segurança não deve confiar inteiramente na segurança do servidor após o "*hardening*", pois alguma configuração insegura pode ter passado despercebida, ou alguma nova vulnerabilidade, desconhecida quando o "*hardening*" foi implantado, podendo ter afetado o servidor em questão.

Seguindo o princípio da defesa em profundidade, recomenda-se que o servidor seja ainda protegido por outros recursos, como *firewall*s, proxies reversos, IDS (HIDS e NIDS) e IPS.

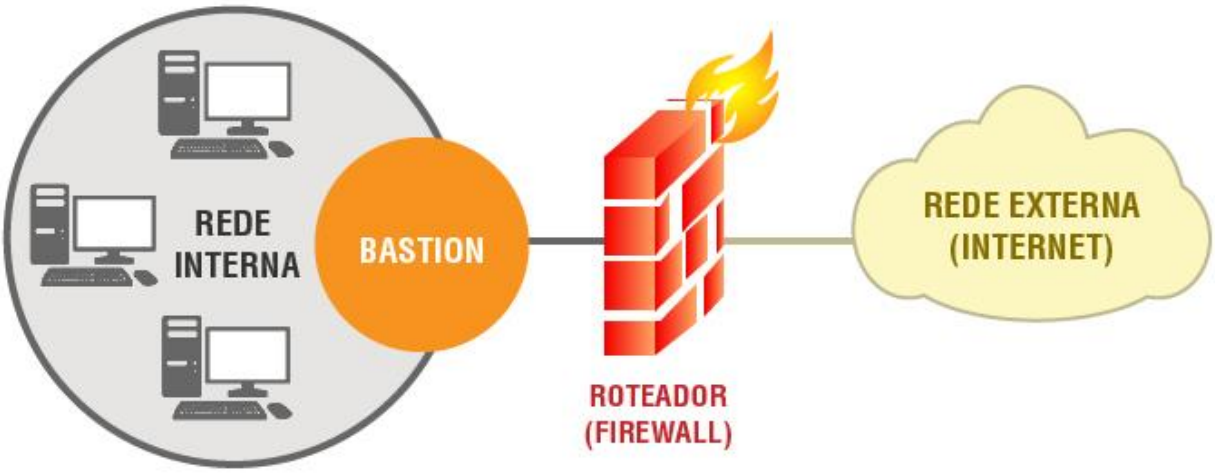
É fortemente recomendado, para que haja melhor e maior aproveitamento do presente módulo que instale o **Windows server 2008**, colocado no link de recursos complementares da disciplina.

02

Em um ambiente Microsoft Windows é vital utilizar o conceito de *bastion host* para garantir a integridade do sistema.

**Bastion host** é um termo aplicado a um *host* que age como um check-point entre a rede interna e a Internet, ou entre sub-redes da Intranet.

Para um Bastion *Host* conectado à internet maior atenção deve ser dada à segurança - é o ponto mais exposto e, por essa razão, deve ser o mais forte.




Existem inúmeras falhas de segurança documentadas que permitem ao invasor acesso total à máquina-alvo, quando esta é disponibilizada em uma rede pública e com a instalação padrão do sistema operacional.

Até o ano de 2013, esse valor era inferior a 10 minutos, de modo que uma máquina conectada à internet por um tempo superior provavelmente já estará infectada por algum *worm* ou foi invadida.

03

O Internet Storm Center (ISC) publica uma estatística sobre o tempo em que uma máquina sem nenhuma correção de segurança "sobrevive" na internet. Para obter mais informações clique aqui.

O bastion *host* será uma máquina exposta na rede pública disponibilizando recursos e serviços. Por ser uma máquina com serviços públicos, essa será a primeira barreira a ser vencida por um invasor para tentar obter acesso aos sistemas da rede privada.

Existem várias implementações possíveis de bastion *host*s, de acordo com os serviços que ele oferece. Alguns exemplos:

a) *Firewall gateways*;
b) Servidores web;
c) Servidores FTP;
d) Servidores de nome DNS;
e) Transportadores de e-mail.

No caso, um bastion *host* pode oferecer mais de um serviço, conforme as topologias.

**Clique aqui**

Veja em: http://www.dshield.org/infocon.html e https://isc.sans.edu/dashboard.html.

04

## 2 - CHECK-LIST

É recomendado planejar a instalação e escrever um ***check-list* das atividades** a serem realizadas e auditadas nos servidores públicos:

a) Remover ou desabilitar todos os serviços não necessários no *host*;
b) Remover ou desabilitar todas as contas de usuário não necessárias;
c) Remover ou desabilitar todos os protocolos de rede não utilizados;
d) Configurar adequadamente os registros de log do sistema para que possam identificar possíveis ataques ou atividade suspeita;
e) Implantar um sistema de detecção de intrusão de *host*;
f) Atualizar o sistema operacional com as últimas correções de segurança disponibilizadas pelo fabricante;
g) Filtrar todas as portas que não são necessárias para o *host*;
*h)* Utilizar conexão criptografada para conectar no *host*;
*i)* Evitar a instalação de aplicativos não necessários e notadamente vulneráveis, como Flash, PDF Viewers, Java.

A seguir, veremos com mais detalhes as configurações de segurança recomendadas para servidores que utilizam o sistema operacional Microsoft Windows.

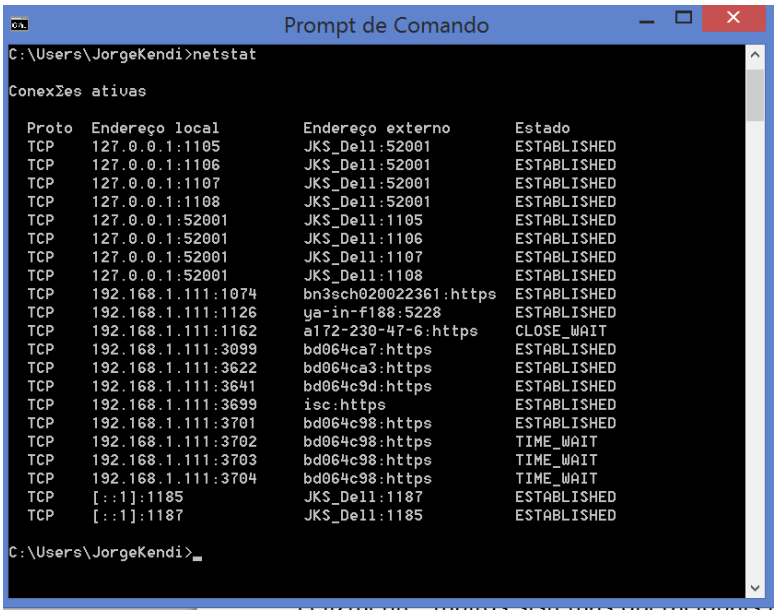
05

## 3 - CONFIGURAÇÃO DE FILTROS DE PACOTES

O Windows XP, 2003, 2008 e 7 trazem no próprio sistema operacional um aplicativo para controlar o filtro de pacotes. No XP e 2003, o filtro de pacote é simples, o que justifica o uso de aplicativo de terceiros para melhor controle do filtro, como o Zone Alarm, da Check Point. No Windows 7 e Windows Server 2008, o aplicativo recebeu atualizações permitindo a configuração de perfis e a importação e exportação de regras, entre outras funcionalidades.

Pode-se utilizar uma ferramenta que acompanha o sistema operacional: netstat, ou ferramentas adicionais como o TCPview da suíte Sysinternals.

Por exemplo:



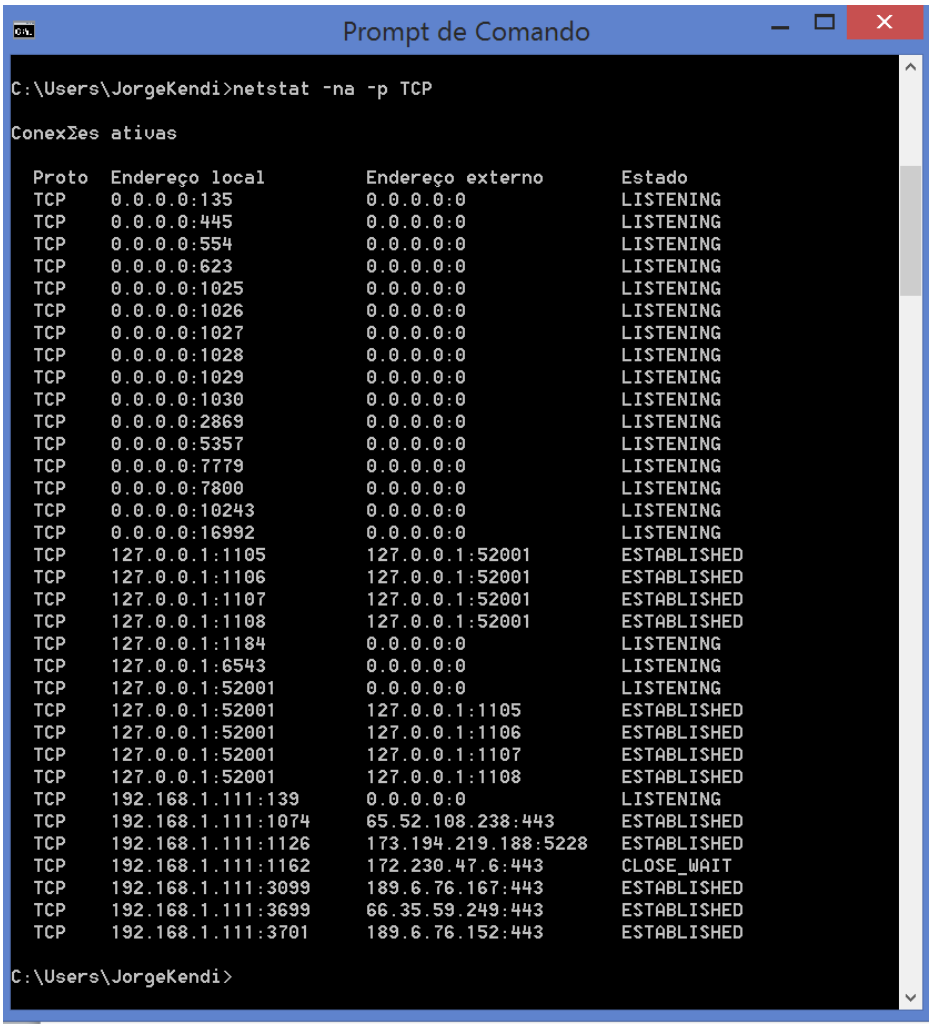

**Exame das portas feitas pelo Netstat.**
**Fonte: O Autor, 2015.**

Através do netstat e do TCPview, verificamos as portas abertas no servidor para localizar e desabilitar o serviço em questão, ou filtrar a porta.

06

Cada serviço de rede presente em um servidor pode escutar uma porta, TCP ou UDP, para receber conexões de outros servidores ou clientes. Alguns desses serviços são importantes para o bom funcionamento do servidor, e nem sempre podem ser desabilitados. Quando verificamos as portas abertas em uma configuração padrão de um servidor Windows, vemos que existe uma série de portas que são abertas por padrão no sistema. Colocar um sistema de forma pública na internet, sem a devida filtragem dos serviços que não estão em uso, é arriscado e pode comprometer a segurança do servidor.

Felizmente, muitos sistemas operacionais permitem que seja configurado um **filtro de pacotes** para controlar as portas que estarão disponíveis para serem conectadas por *host*s externos. A versão padrão do Windows XP, 2003, 2008 e 7 traz no próprio sistema operacional um aplicativo para controlar o filtro de pacotes.




**Conexões de rede.**
**Fonte: O Autor, 2015.**


Lembre-se de que a filtragem é **local**, e não deve ser utilizada para substituir uma filtragem de perímetro, propriamente feita através de um *firewall*, mas apenas como um mecanismo adicional de segurança (defesa em profundidade).

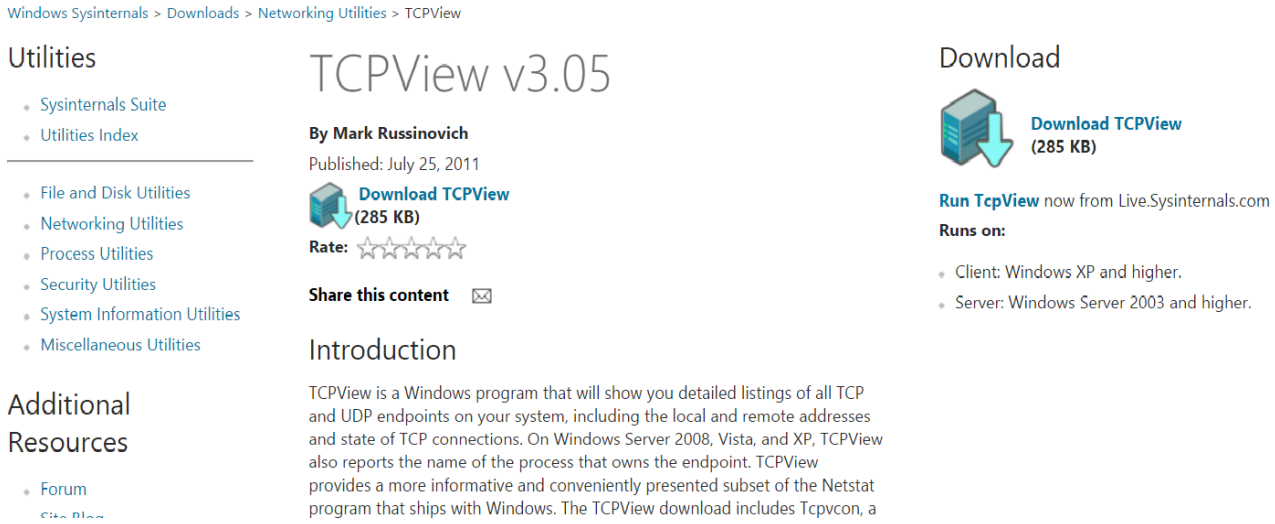
07

Você deve observar também que, caso haja muitos servidores disponibilizando serviços públicos, a configuração de filtros de pacotes locais em cada servidor pode tornar o gerenciamento do ambiente complexo, de modo que o uso de filtros em cada servidor deve ser feito com parcimônia.

Para listar as portas que estão aguardando conexão de rede ou as conexões estabelecidas, pode-se usar a ferramenta do próprio sistema operacional, neste caso o comando netstat, como mostrou a figura anterior.

A suíte de ferramentas **SYSInternals** oferece outra ferramenta para listar com mais detalhes as conexões de rede estabelecidas e seus respectivos processos, além de também listar as portas que estão aguardando por conexões de rede.

Clique aqui para obtenção da ferramenta TCPview.



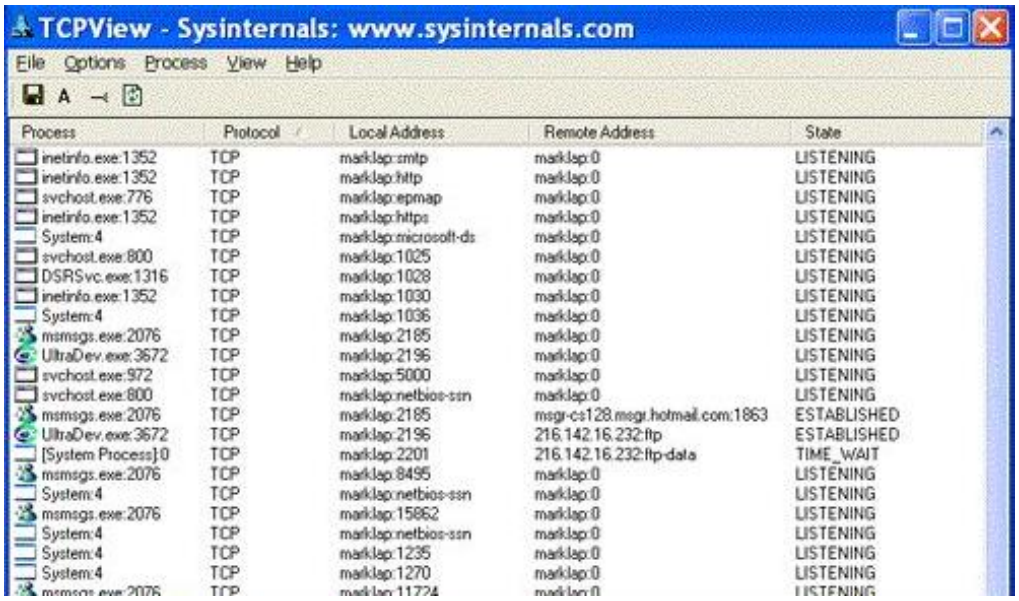

**Site da Microsoft para a obtenção do TCPView.**
**Fonte: https://technet.microsoft.com, 2015.**

**Clique aqui**

https://technet.microsoft.com/en-us/sysinternals/tcpview.aspx

08

Abaixo um exemplo de listagem de portas da ferramenta TCPView.




**Conexões de rede com o TCPView.**
**Fonte: Internet, 2015.**

Através do netstat e do TCPview, verifica-se as portas que estão abertas no servidor, de modo a localizar e desabilitar o serviço em questão ou filtrar a porta.

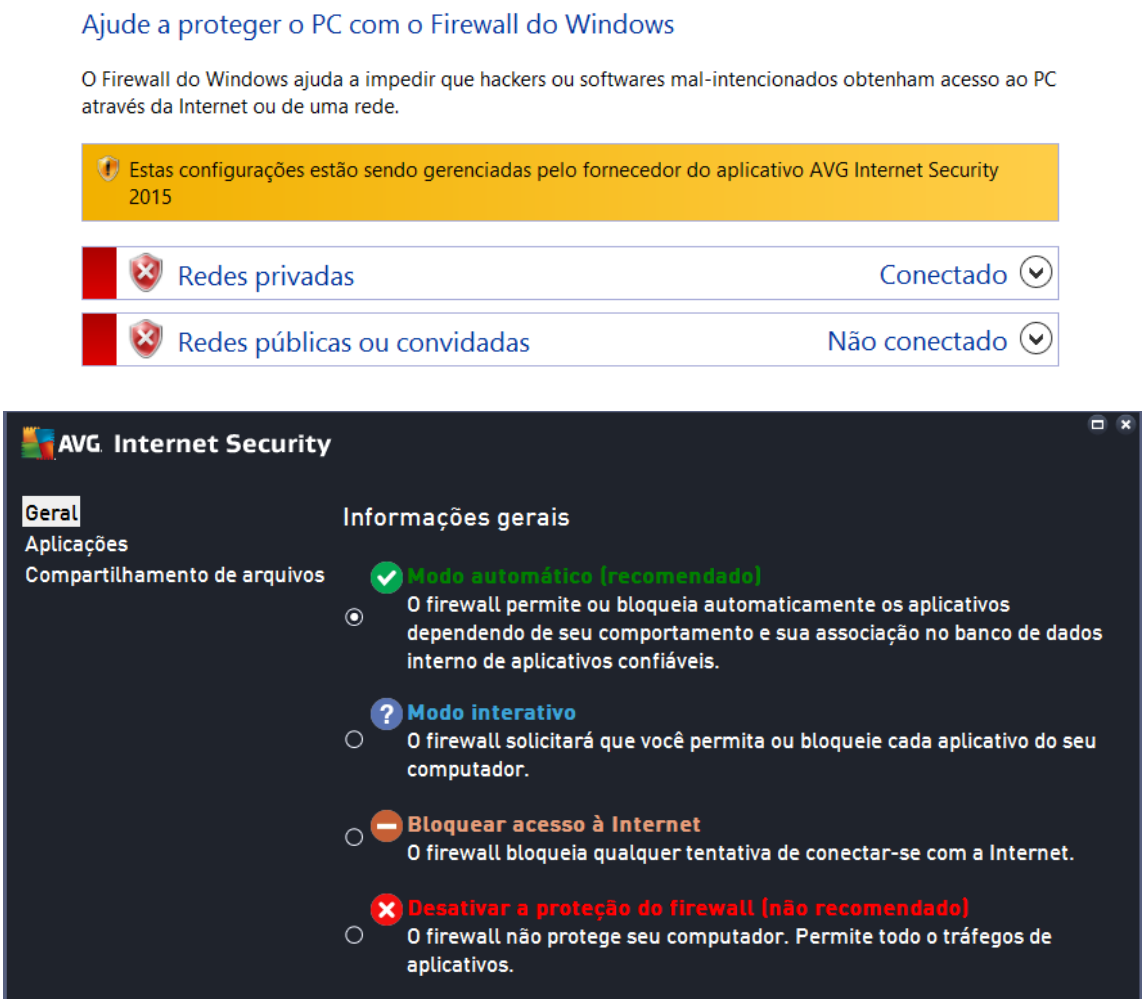
Outra forma de verificar as portas abertas é utilizando o **Nmap**.

Desabilitar serviços em sistemas Windows é um processo que demanda certa paciência. Caso um serviço essencial seja desabilitado, algum comportamento inesperado pode acontecer. Recomenda-se que seja utilizada uma máquina de testes, para se criar familiaridade com o processo, antes de desabilitar serviços em servidores em produção.

**09**

O Windows *Firewall* é o aplicativo que acompanha o sistema operacional para controle de conexões de rede, que normalmente vem configurado por padrão na instalação padrão do Windows.

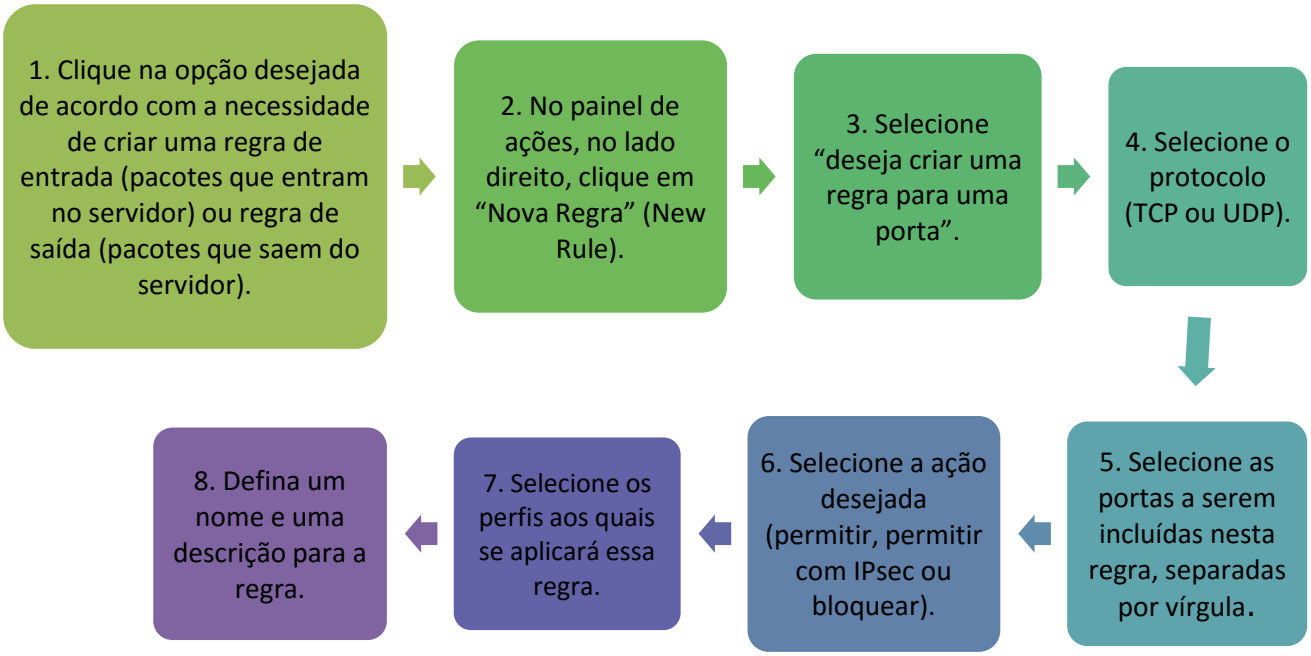
Nas versões Windows 7 e Windows Server 2008, o aplicativo recebeu atualizações que permitiram a configuração de perfis, importação e exportação de regras, entre outras funcionalidades.




**Exemplo de aplicação do *Firewall*.**
**Fonte: O Autor, 2015.**

10

Abaixo, seguem os passos para criar uma regra no *firewall*. O cenário supõe o Windows Server 2008:




Por fim, verifique a regra criada no painel da ferramenta. Você pode testar as suas regras com os comandos já apresentados.

11

## 4 - Criação de uma linha base de segurança (baseline)

Uma ***baseline*** é uma referência inicial de segurança, um ponto inicial para a evolução para uma configuração segura.

Além da *baseline*, é preciso criar um **mapa do tempo** (***timeline***) dos servidores da rede.

No mapa do tempo devem ser registrados:

- a data de instalação do sistema operacional,
- das principais correções e
- da instalação de aplicativos.

Essa linha do tempo será útil para manter atualizado o inventário dos sistemas e principalmente para uma eventual auditoria.

Os servidores Windows possuem alguns perfis padrão de segurança, de acordo com o papel que aquele servidor irá desempenhar. Mais adiante, serão vistas algumas ferramentas que permitem a criação de uma *baseline* e a posterior auditoria para verificar se a configuração atual atende ao mínimo necessário de segurança.

12

## 5 - DESABILITANDO SERVIÇOS DESNECESSÁRIOS

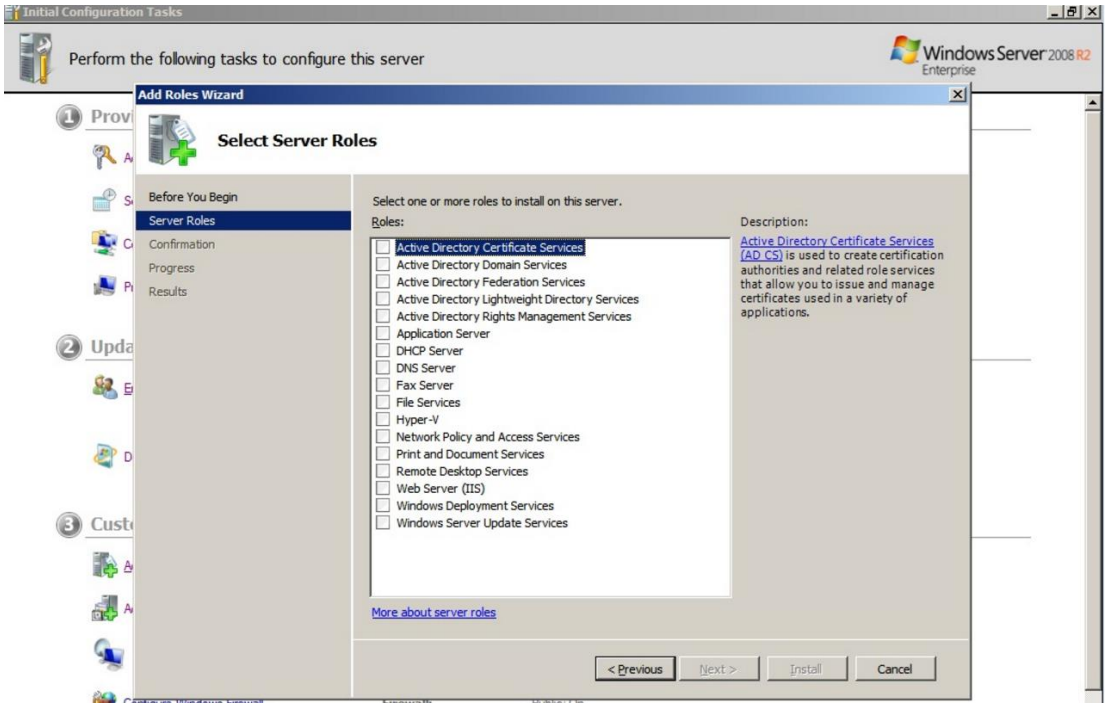
O Windows Server (no caso referido aqui, trata-se do 2008) trouxe alguns avanços de segurança que auxiliam na criação de um bastion *host*: os **papéis**. O conceito de papéis (**roles**) ajuda no processo de habilitar somente programas necessários e evitar comprometer a segurança do sistema.

O **sistema de papéis**, quando aplicado, irá:

a) Iniciar somente os serviços necessários;
b) Liberar exceções nos filtros de pacotes nas interfaces de rede que forem necessárias.

Logo após a instalação do sistema operacional, depois do primeiro acesso ao sistema, a ferramenta de configuração de papéis é apresentada. Com a escolha de um determinado "role", o aplicativo iniciará somente os serviços e liberará exceções nos filtros de pacotes nas interfaces de rede em que isso for necessário.

Essa é uma grande evolução, que facilita o processo de desabilitar serviços desnecessários e liberar regras de acesso para os serviços que estão sendo usados.



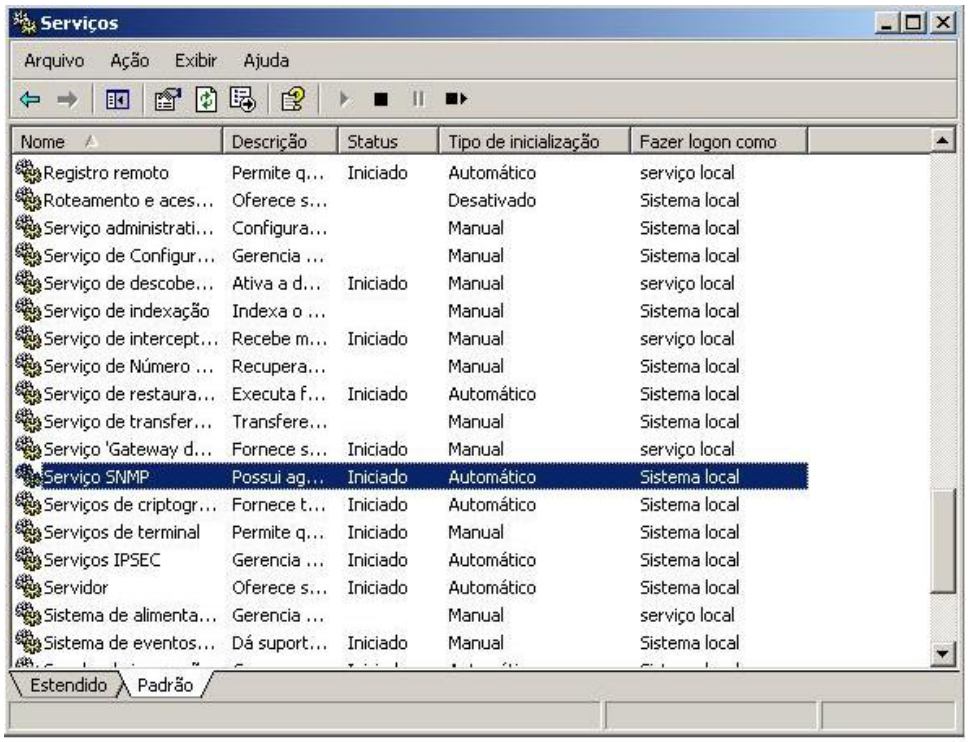

**Regras de configuração do Windows 2008 Server**
**Fonte: http://social.technet.microsoft.com/.../18.jpg**

13

Nos sistemas operacionais que possuem o recurso de roles, é importante verificar se os roles realmente disponibilizam **somente os serviços necessários**. Nos sistemas operacionais mais antigos da Microsoft (Windows XP e 2003 Server), os serviços se tornam o principal controle de recursos disponíveis no sistema, de modo que devem ser desabilitados de forma manual.

A ferramenta **Services MMC**, que acompanha o sistema operacional, ajuda nessa tarefa. Ela pode ser encontrada no Painel de Controle, no item “Ferramentas Administrativas”.

Ao iniciar a ferramenta Services, obtemos a tela a seguir.




**Serviços do Windows XP.**
**Fonte: Internet, 2015.**

Através da ferramenta, pode-se iniciar, encerrar e desabilitar serviços. Para iniciar e encerrar um serviço basta clicar em cima do serviço e escolher uma das opções, que se assemelham aos controles de um programa tocador de música.


Note que alguns serviços não podem ser encerrados, pois são serviços essenciais para o funcionamento do sistema operacional.

Para desabilitar um serviço, basta clicar no serviço desejado e mudar o “startup type” para “disabled”.

14

Uma tarefa importante para a configuração segura de servidores consiste em saber a porta TCP ou UDP associada a um determinado serviço, de modo que possamos desabilitar os serviços desnecessários que abrem portas de rede no servidor. Essa tarefa pode ser realizada utilizando alguns utilitários disponíveis na internet.

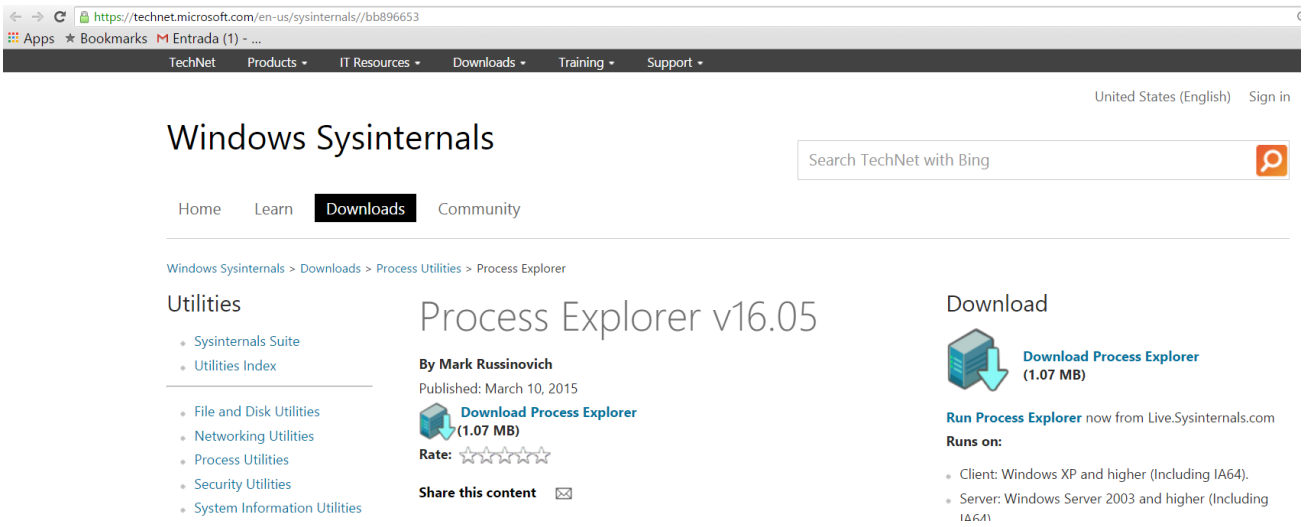
A seguir, um exemplo de **como descobrir um serviço que corresponda a uma porta específica**.

**1.** Através do **netstat –ano** verificamos uma porta da qual desejamos saber o serviço correspondente. Na figura a seguir podemos verificar as portas e respectivos processos rodando nas mesmas.


**C:\Users\JorgeKendi>netstat –ano.**
**Fonte: O Autor, 2015.**

15

**2.** A partir do **número do processo**, verificamos no Process Explorer as propriedades do processo em questão.




**Site para obtenção do Windows Sysinternals.**
**Fonte: site www.microsoft.com, 2015.**

**3.** No caso, um dos serviços apresentados é o serviço que procuramos. O próximo passo é **desabilitar os serviços em questão**, um a um, até que a porta em questão desapareça do netstat ou do TCPview.

**4.** Clique aqui para **consultar a lista das portas** bem conhecidas e os processos correspondentes, onde se pode confirmar que a porta 1900 UDP corresponde de fato ao serviço SSDP, que possui relação com a descoberta de dispositivos PnP.

**Process Explorer**

Process Explorer pode ser encontrado no endereço: http://technet.microsoft.com/wn-us/sysinternals/bb896653)

**Clique aqui**

http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

16

**5 - Ferramentas de análise da segurança do Windows**

Destacamos abaixo, algumas das várias ferramentas de análise da segurança do Windows:

a) Windows Management Instrumentation Console (WMIC);
b) Windows Server Update Services (WSUS);
c) Microsoft *Baseline* Security Analyzer (MBSA).

As ferramentas que acompanham o sistema operacional são preteridas por normalmente continuarem funcionando mesmo após grandes atualizações, como no caso de um Service Pack. Porém, ferramentas

terceiras normalmente trazem informações mais detalhadas e com relatórios úteis na gestão dos servidores. Veremos alguns exemplos de ferramentas úteis para a análise de segurança de servidores Windows.

17

**5.1 - WMIC**

Windows Management Instrumentation Console (WMIC), também conhecida como “canivete suíço do Windows”, é executada em linha de comando e pode ser executada no servidor local ou remoto pela rede de dados.

Essa ferramenta está disponível em todas as versões do Windows a partir do Windows NT.

Acesse o console de sua máquina, no prompt de comando digite WMIC e você obterá o prompt: **wmic:root\cli>**

Para ajuda digite /? e pressione enter. Para sair, digite **exit** e pressione enter.

A sintaxe do WMIC é sempre: wmic <objeto> <ação>

Exemplo:


**wmic useraccount list brief.**
**Fonte: O Autor, 2015.**

Principais objetos no auxílio de auditoria de segurança:

a) startup;
b) process;
c) cpu;
d) group;

e) useraccount.

O WMIC é capaz de gerar relatórios em vários formatos, como CSV, XML, HTML, através da diretiva /FORMAT:<formato>.

| |
|---|
| **Startup**<br><br>Lista todos os processos que são iniciados junto com o sistema operacional. |
| **Process**<br><br>Lista dos processos executados pelo sistema. |
| **Cpu**<br><br>Informações sobre o processador físico. |
| **Group**<br><br>Lista de grupos cadastrados no sistema. |
| **Useraccount**<br><br>Lista dos usuários cadastrados no sistema. |

18

**5.2- SYSInternals**

A suíte de ferramentas desenvolvidas inicialmente por Mark Russinavich e Bryce Cogswell oferece a possibilidade de uma verificação mais detalhada do funcionamento do sistema operacional.

As ferramentas podem ser baixadas gratuitamente do site da Microsoft. Instale a ferramenta em sua máquina para você poder obter as informações aqui discutidas.

Com o objetivo de auxiliar o analista a gerenciar o *host*, resolver problemas e diagnosticar o sistema operacional e aplicativos, a Microsoft adquiriu em 2006 a suíte de ferramentas SYSInternals e contratou Mark Russinavich para continuar na equipe de desenvolvimento da suíte de ferramentas.

A seguir algumas ferramentas importantes da suíte do SysInternals:

a) Autoruns;
b) Diskmon;
c) EFSDump;
d) ProcDump;
e) PsService;
f) RootkitRevealer;

g) Process Monitor.

| **site da Microsoft**<br><br>http://technet.microsoft.com/en-us/sysinternals/ |
|---|

| **Autoruns**<br><br>Mostra os programas que inicializam automaticamente com o sistema. |
|---|
| **Diskmon**<br><br>Captura toda a atividade do disco rígido. |
| **EFSDump**<br><br>Verifica informações sobre arquivos cifrados. |
| **ProcDump**<br><br>Captura área de memória de processos. |
| **PsService**<br><br>Visualiza e controla serviços. |
| **RootkitRevealer**<br><br>Verifica o sistema em busca de programas maliciosos do tipo rootkit. |
| **Process Monitor**<br><br>Monitora diversas informações sobre processos, incluindo alterações do registry e do sistema de arquivos. Muito útil para verificar o comportamento de certos programas. |

19

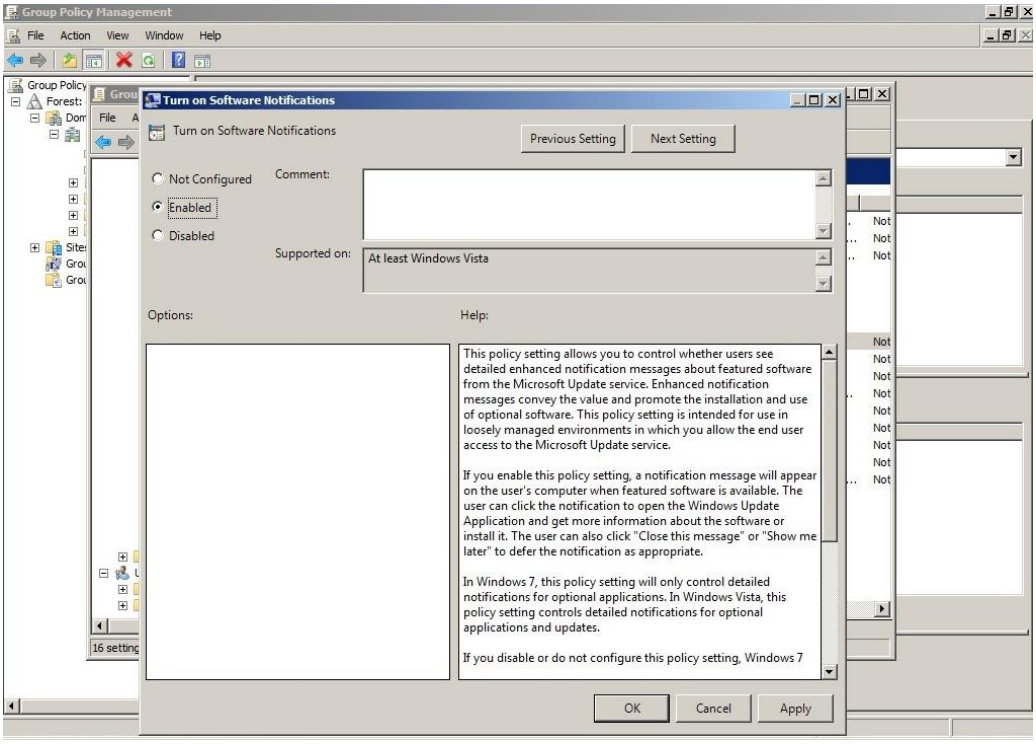
**5.3 – WSUS**

| Windows Server Update Services (WSUS) é uma ferramenta da Microsoft que auxilia no processo de atualizações dos sistemas e aplicativos Microsoft. |
|---|

É um serviço que pode ser executado em versões do Windows Server 2000, 2003 e 2008. Esse serviço é responsável por baixar as atualizações dos servidores da Microsoft e distribui-las para as estações de trabalho e servidores da rede. Essa distribuição pode ser realizada automaticamente de forma pré-aprovada pelo administrador do sistema ou com aprovações manuais para cada atualização, em casos mais críticos.

O WSUS também é capaz de gerar relatórios personalizados da situação de cada *host* da rede, exibindo detalhes da atualização do sistema operacional e aplicativos Microsoft.

A instalação do WSUS, pela sua complexidade, não será coberta por esse curso, porém existem diversos guias na internet que explicam em detalhes a instalação da ferramenta.  A seguir, uma tela do WSUS em execução:



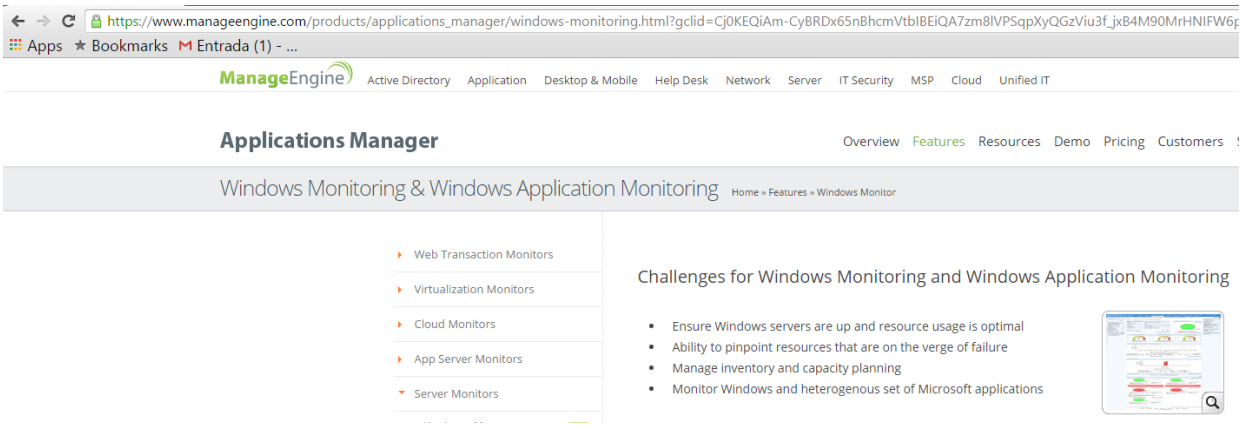

**Tela de atualização do WSUS.**
**Fonte: Internet, 2015.**

20

### 5.4 - MBSA

Microsoft *Baseline* Security Analyzer (MBSA) é uma ferramenta capaz de verificar se servidores – com Windows Server 2003 e 2008, e estações de trabalho com Windows XP, Vista e 7 – estão de acordo com as recomendações de segurança da Microsoft e ainda se estão com as últimas versões das correções de segurança instaladas.

Para utilizar o MBSA, será necessário ter conta no sistema com privilégio de administrador. Veja a figura a seguir.




**MBSA**

**Fonte: https://www.manageengine.com/products/applications_manager/..., 2015.**

Além de verificar a instalação de correções de segurança, o MBSA também verifica falhas comuns na configuração dos servidores, como o serviço de atualização desligado, contas de usuário que nunca expiram, contas sem senhas, configurações com fragilidade de segurança do Internet Explorer, entre outras. O MBSA é gratuito para usuários Windows.

21

### 5.5 - Microsoft Security Compliance Manager

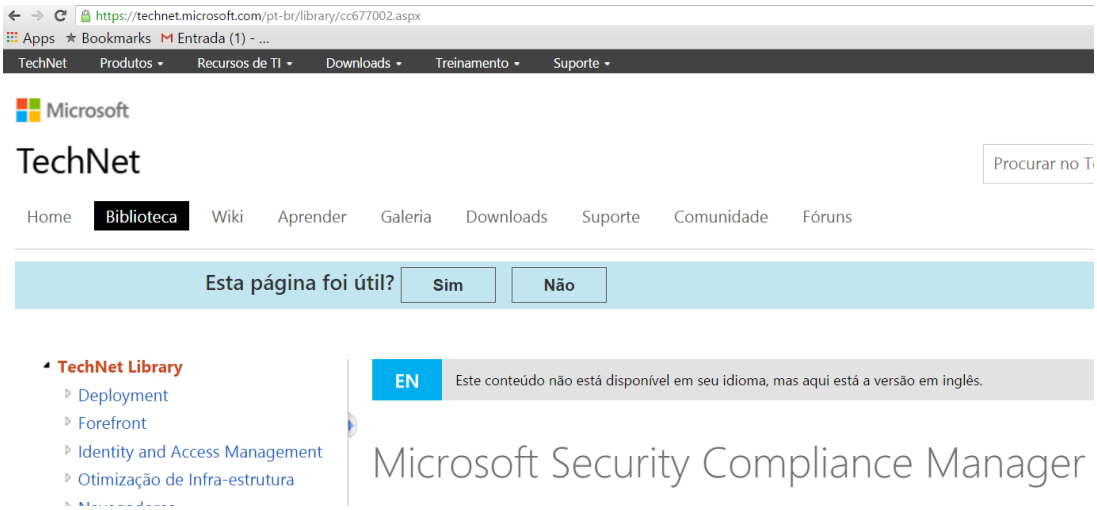
A ferramenta Microsoft Security Compliance Manager é completa para a segurança de servidores e estações Windows.

Essa ferramenta tem por objetivo concentrar uma série de conhecimentos sobre a segurança de servidores e facilitar o processo de *hardening* de servidores e estações. Ela permite aplicar uma série de parâmetros de segurança, a customização de *baseline*s e a exportação em formatos fáceis de aplicar em um ambiente.

Recursos presentes na ferramenta:

a) Gerenciamento centralizado de *baseline*s;
b) Provê uma interface centralizada de gerenciamento para prover, planejar e customizar *baseline*s, incluindo as recomendadas para os sistemas operacionais Windows;
c) Customização de *baseline*s que permite customizar, comparar, juntar e revisar as *baseline*s;
d) Exportação para múltiplos formatos. Permite a exportação das configurações em diversos formatos, incluindo XLS, GPO, DCM e SCAP, para permitir a automação da implantação e o monitoramento da aderência às *baseline*s definidas. Suporta Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, Hyper-V, Windows 7, Windows Vista, Windows XP, BitLocker Drive Encryption, Windows Internet Explorer 8, Microsoft Office 2010 e Microsoft Office 2007 SP2.

A figura mostra site para o *download* da ferramenta:



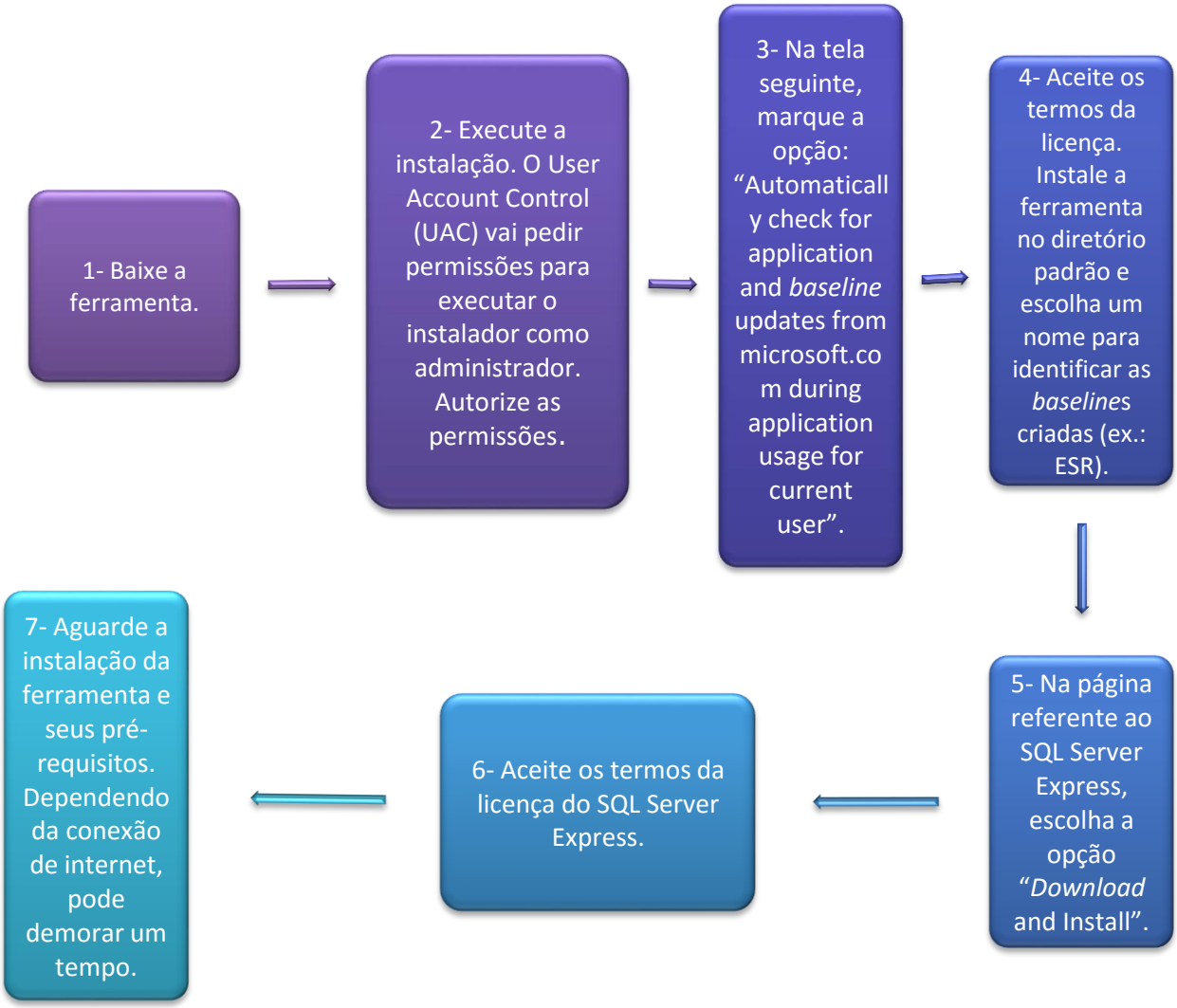

**MSCM**

**Fonte: www.microsoft.com, 2015.**

23

Microsoft Security Compliance Manager requer o .NET runtime e o SQL Server Express.

Os seguintes passos são necessários para a instalação da ferramenta (vale lembrar que, dependendo da versão, os passos podem apresentar diferenças, porém a instalação é o padrão Windows, portanto sem dificuldades):





Durante a instalação, caso o .NET Runtime não esteja instalado, o MSCM não será instalado e será necessário baixar e instalar esse componente.

24

Ao iniciarmos o MSCM pela primeira vez, o aplicativo automaticamente busca e instala as versões mais recentes dos templates de segurança para as plataformas suportadas por ele. Essa atualização é importante para garantir que as últimas versões dos templates estão sendo utilizadas.

Essa atualização pode também ser realizada através do menu "Tools", opção "Check for *baselines*".

Na tela principal, temos então os seguintes componentes:

**a) *Baseline* Library**

Lista todas as *baseline*s numa estrutura hierárquica. Ao clicar em uma *baseline* com o botão direito, um menu apresenta alguns comandos que podem ser aplicados.

**b) *Baseline* Information Pane**

Apresenta informações sobre a *baseline* selecionada;

**c) Actions**

Apresenta comandos para a gerência das *baseline*s.

25

### 5.6 - Sistemas de arquivos e gerenciamento de usuários

Um sistema seguro deve utilizar um sistema de arquivos que **suporte a criação de permissões**, de modo a limitar o acesso dos usuários para minimizar um potencial estrago em caso de comprometimento de uma conta de usuário, além de incluir o mínimo de usuários possíveis.

O sistema de arquivos padrão do Windows, a partir da versão NT, é o NTFS. Apesar de suportar outros sistemas, como o FAT32, recomenda-se que seja usado sempre o NTFS por questões de segurança, pois ele possui a capacidade de ajuste de permissões por usuário (ACL), para que o sistema de arquivos possa ser configurado de modo que os usuários só tenham acesso ao que realmente for necessário para a utilização do sistema (princípio do menor privilégio).

Verificam-se as permissões de um determinado arquivo ou pasta no sistema, através das propriedades, na aba Segurança. Pode-se editar as propriedades de segurança desse objeto, adicionando ou removendo permissões.


É importante lembrar que os servidores que utilizam NTFS como sistema de arquivos já possuem uma configuração inicial razoavelmente restritiva, de modo que um usuário não administrador tenha poucos privilégios no ambiente, não conseguindo instalar novos programas e com permissão de gravação apenas na sua pasta de trabalho.

Ao configurar o sistema, devemos verificar ainda as contas de usuário, removendo todas as contas que não estão em uso, especialmente contas de convidado (guest), e definindo senhas complexas para os usuários e para a conta de administração.

Outra prática comum é renomear a conta de administrador, de modo que dificulte a ação de ataques de força bruta, com o objetivo de encontrar a senha dessa conta.

26

**5.7 - Group Policy Objects**

Uma das grandes tarefas de um administrador de redes é o gerenciamento de usuários, grupos e computadores de uma rede e, dependendo do tamanho da estrutura da organização, essa tarefa pode demandar horas e mais horas de planejamento e execução.

Uma ferramenta de vital importância para suprir essa demanda é sem dúvida o **Group Policy Objects** (**GPO**), presente nos sistemas operacionais de rede da Microsoft desde a versão 2000. Com ele controla-se boa parte do comportamento tanto das estações de trabalho que compõem nosso parque de máquinas quanto do próprio servidor.

Essas políticas facilitam a configuração de várias máquinas ao mesmo tempo, bastando apenas escolher qual política utilizar para toda a empresa, para um grupo específico ou mesmo para apenas uma estação de trabalho.

Com o GPO pode-se:

a) Restringir ícones e botões da área de trabalho ou do menu Iniciar;
b) Limitar o número de programas a serem executados;
c) Restringir opções do Active Desktop;
d) Remover programas desnecessários;
*e)* Programar instalações remotas de programas;
*f)* Configurar Internet Explorer.

27

As configurações executadas via GPOs são aplicadas para usuários, computadores, *member servers*, *Domain Controlers*, mas apenas para computadores rodando Windows 2000 (Server ou Professional), Windows XP, Windows Vista, Windows Server 2003 e Windows Server 2008.

A primeira aproximação que a Microsoft fez com políticas de grupos foi introduzida no Windows NT 4 através do Police Editor, mas foi com o lançamento do Windows 2000 Server que foi introduzido ao mundo Windows o Group Policy Editor. Saiba+

Com o lançamento do Windows 2003 Server, a implementação de GPOs ficou ainda mais fácil, principalmente com o Group Policy Management Console.

No Windows 2008 Server, o Group Policy ganhou mais opções, incorporando muitos dos serviços antes feitos apenas através de scritps, tais como mapeamento de impressoras, discos e aspectos do desktop do usuário, inclusão de filtros WMI (Windows Management Instrumentation), permitindo a criação de GPOs específicas conforme o hardware da estação e criação de modelos com o recurso “Starter GPO”.

Referir-se ao GPO é referir às Diretivas de Grupo. Uma **diretiva de grupo** é um conjunto de regras a serem utilizadas a fim de facilitar o gerenciamento, configuração e segurança de computadores e usuários.

Pode-se atribuir diretivas em uma GPO. Essa GPO com essas regras podem ser atribuídas a um:

a) Site
b) Domínio
c) OU

**Saiba+**

O Group Policy Editor incluía:

a) Configuração dos principais componentes do Windows;
b) Configuração dos principais recursos dos Active Desktop;
c) Configuração das regras de segurança;
d) Instalação de Softwares do tipo Windows Installer;
e) Configurações por computador ou usuário;
f) Herança, bloqueio de herança e regras mandatórias.

**Site**

É o mais alto nível e normalmente atribuído a GPOs mais genéricas, válidas para qualquer usuário/computador/domínio nesse site.

**Domínio**

Vem em segundo nível. Configurações feitas nesse nível afetaram usuários/ computadores dentro do domínio.

**OU**

O que se aplica nas OUs afetarão todos os usuários/computadores dentro dela.
Para criar uma GPO, basta clicar com o botão direito em uma das opções acima, clicar em Propriedades e na aba Group Policy.

28

**5.8 - Políticas de usuários e de computador**

O console GPO é dividido em duas partes:

| Computer Configuration | User Configuration |
|---|---|
| •Permite aplicar políticas que sempre estarão ativas nas estações de trabalho, independente do usuário logado. Como essas políticas são permanentes, a chave de registro modificada em questão é HKEY_LOCAL_MACHINE. | •Permite a implementação de políticas diretamente nos usuários, não sendo permanente na estação de trabalho. Essa política estará associada ao usuário e será aplicada em qualquer estação na qual o usuário faça login. |

Se houver algum conflito entre as configurações dos computadores e dos usuários, as configurações dos usuários vão prevalecer.

Opções de GPO:

a) **Software Settings**;
b) **Windows Settings**;
c) **Administrative templates**.

| |
|---|
| **Software Settings**<br><br>Nessa categoria são configurados, por exemplo, a distribuição de aplicações para o usuário. |
| **Windows Settings**<br><br>Permite ao administrador customizar as configurações do Windows. Essas opções são diferentes para usuários e computadores. |
| **Administrative templates**<br><br>Modelos utilizados para configurar as definições de políticas de segurança de usuários e computadores, essas opções acessam diretamente as chaves de registro HKEY_LOCAL_MACHINE e HKEY_CURRENT_USER. |

29

### 5.9 - Ordem das GPOs

Dependendo da estrutura da organização, pode-se utilizar várias GPOs para as mais diversas tarefas e, muitas vezes, essas GPOs podem ser aplicadas a um mesmo objeto, seja diretamente ou por herança.

Nesse caso, as GPOs possuem uma hierarquia para serem aplicadas:

a) GPO local;

b) GPO de site;
c) GPO de domínio e
d) GPO de OU.

As GPOs são baseadas em modelos que possuem uma lista de opções configuráveis de forma bastante intuitiva. Em sua maioria oferecem as seguintes opções:

a) Habilitada;
b) Desabilitada;
c) Não configurada.

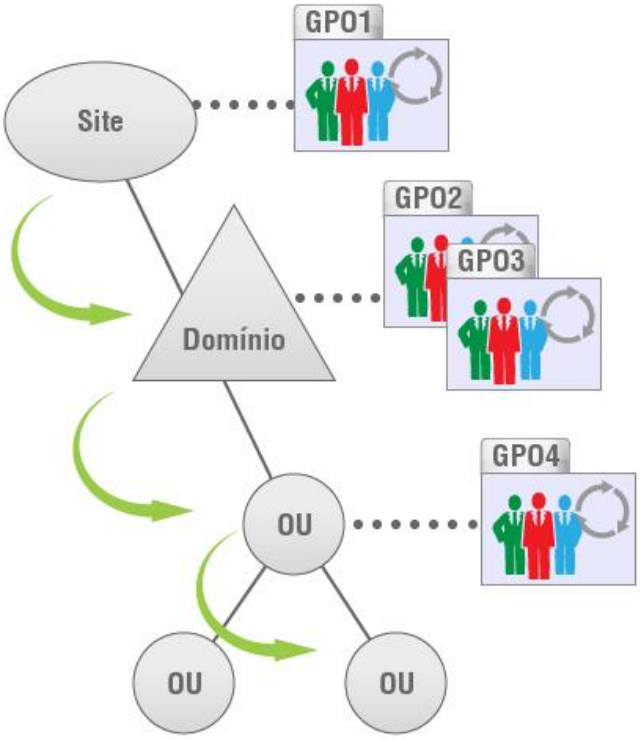
| |
|---|
| **Habilitada**<br><br>Especifica o item que será ativado. |
| **Desabilitada**<br><br>Especifica o item que será desativado. |
| **Não configurada**<br><br>Deixa a opção neutra: não está ativada nem desativada, essa é a opção padrão. |

30

**5.10 - Heranças de GPO**

Para facilitar a criação de GPOs, pode-se especificar em um site uma política global, como mudança de senha ou papel de parede específico, e em domínio ou OU uma política mais restritiva e personalizada. Por padrão, as GPOs podem ser sobrepostas, caso existam políticas habilitadas em um site e em um domínio, seguindo sempre a precedência da mais próxima.

Por exemplo, pode-se configurar em nível de site uma GPO para que os usuários troquem a senha a cada 30 dias. Porém, no domínio foi configurada uma GPO desativando essa política. Nesse caso, a política vigente será a do domínio.




**Heraças de GPO**
**Fonte: Internet, 2015.**

Esse comportamento pode ser alterado através das opções:

a) **Block Policy Inheritance** (Bloquear Herança de Políticas);

b) **Force Policy Inheritance** (Forçar Herança de Políticas).

| |
|---|
| **Block Policy Inheritance (Bloquear Herança de Políticas)**<br><br>Especifica que as configurações da GPO para um determinado objeto não será herdada de seu nível superior. |
| **Force Policy Inheritance (Forçar Herança de Políticas)**<br><br>Especifica que você não permitirá que níveis filhos possam sobrescrever suas configurações de GPO. Por exemplo, se o administrador de rede da empresa deseja criar uma GPO que force os usuários a utilizarem uma senha de nove dígitos e não deseja que os administradores do domínio nem das OUs dos domínios alterem essa política, ele pode criar a GPO, aplicar para todo o site e marcar a opção de Force Policy Inheritance. |

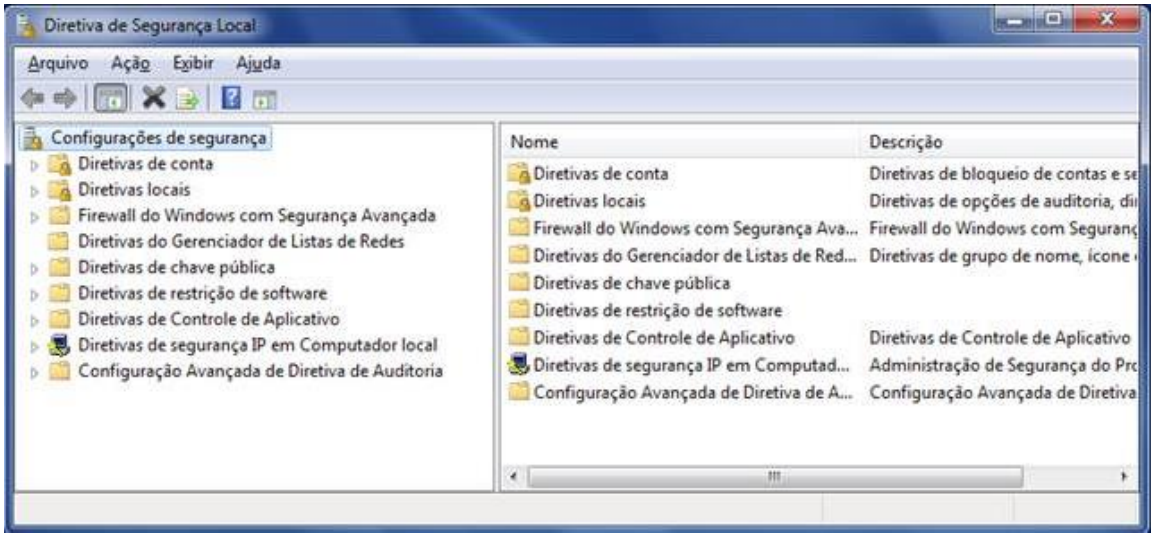
31

## 6 - DIRETIVAS DE SEGURANÇA LOCAL

Quando trabalhamos com o Windows Server em um domínio, a maior parte das configurações de segurança pode e deve ser feita através de GPOs do Active Directory, facilitando a configuração automática de parâmetros de segurança em todas as estações de trabalho automaticamente, eliminando a necessidade de configuração máquina a máquina. Porém, nem todas as organizações trabalham com domínios. Dessa maneira, não dispoem de GPOs para configuração de segurança. Para configurar esses parâmetros de segurança nos servidores pode-se utilizar a diretiva de segurança local.

Uma **diretiva de segurança local** permite ao administrador controlar:

a) Quem acessa os computadores;
b) Quais recursos os usuários estão autorizados a usar no computador;
*c)* Se as ações de um usuário ou grupo são registradas no Log de eventos.




**Diretivas de Segurança local**
**Fonte: Internet, 2015.**

Entre as mais diversas possibilidades de implementação de itens de segurança que o Windows Server possibilita a um administrador de sistemas, destacam-se as seguintes diretivas:

a) Política de senhas;
b) Auditoria de contas;
c) Direitos de usuários;
d) Opções de segurança.

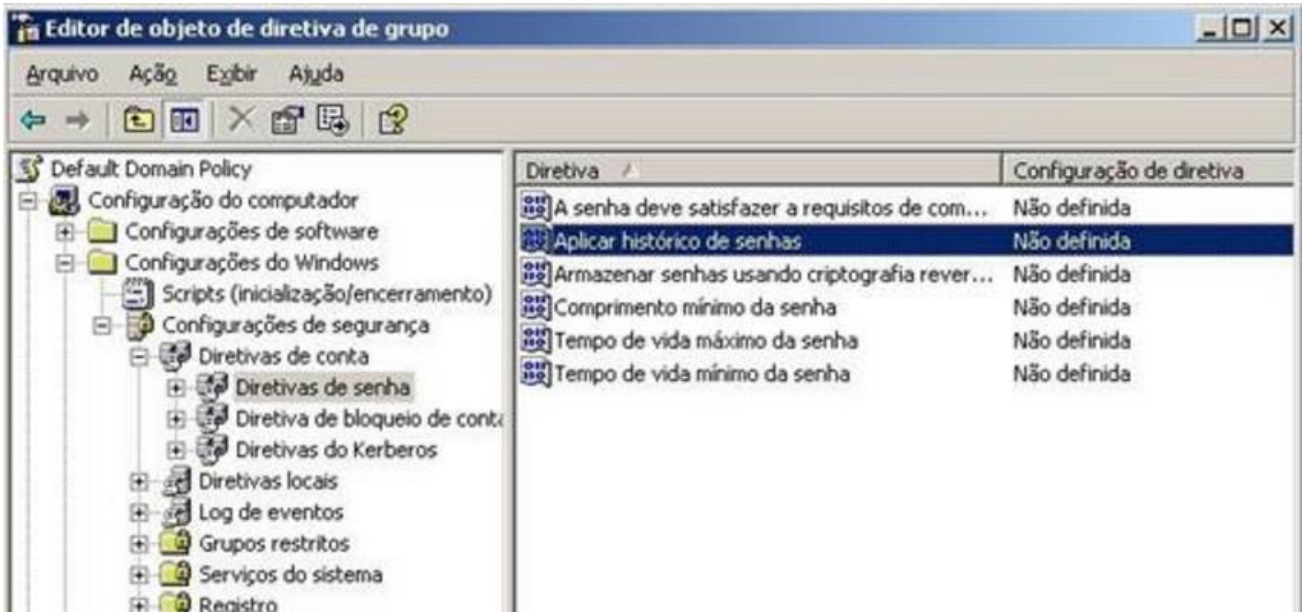
32

**6.1 - Diretiva de senhas**

Sempre é bom lembrar que uma boa política de senhas é fundamental para uma organização. É um controle de segurança, e antes de defini-lo o administrador precisa entender exatamente quais os riscos envolvidos em se definir como o usuário vai trabalhar com suas senhas, o valor do que está sendo protegido com essa senha e os demais controles de acesso que existem adicionalmente além da própria senha.


A organização deve estar preocupada não somente com ataques de força bruta ou por dicionário, mas também evitar que o usuário esqueça a senha e tenha de redefini-la diversas vezes. Porém, uma fraca política de autenticação invalida todas as outras barreiras implementadas, tais como *firewall*s, criptografia e outros.

Para se defender contra essas vulnerabilidades, faz-se necessário uma correta aplicação de diretivas de senha utilizando o console Diretiva de segurança local ou do domínio, se o servidor for um controlador de domínio.




**Diretivas de Segurança local**
**Fonte: Internet, 2015.**

Opções:

a) A senha deve satisfazer a requisitos de complexidade;
b) Aplicar histórico de senhas;
c) Armazenar senhas usando criptografia reversível;
d) Comprimento mínimo de senha;
e) Tempo de vida máximo da senha;

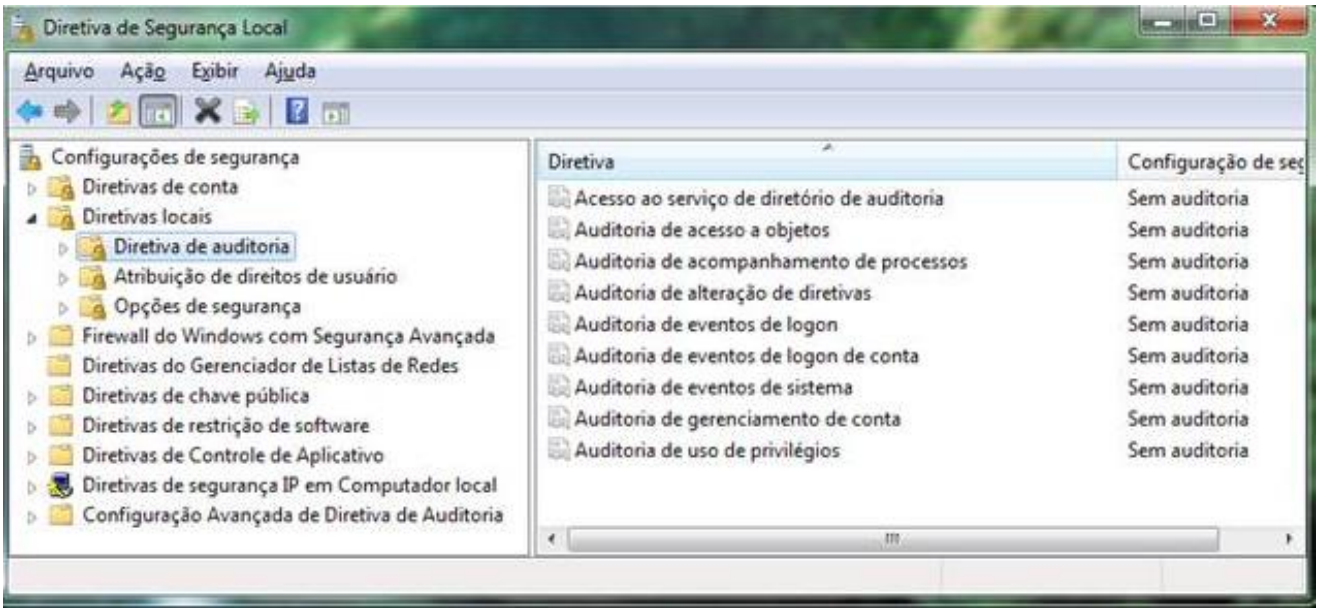
f) Tempo de vida mínimo da senha.

| |
|---|
| **A senha deve satisfazer a requisitos de complexidade**<br><br>Se essa diretiva estiver habilitada, as senhas deverão atender aos itens abaixo informados. |
| **Aplicar histórico de senhas**<br><br>Configuração de segurança que determina o número de novas senhas exclusivas que devem ser associadas a uma conta de usuário, para que uma senha antiga possa ser utilizada. Valor entre 0 e 24 senhas. |
| **Armazenar senhas usando criptografia reversível**<br><br>Diretiva que oferece suporte a aplicativos que necessitam armazenar a senha original do usuário, essa opção só deve ser utilizada se realmente for necessário. |
| **Comprimento mínimo de senha**<br><br>Configuração de segurança que determina o tamanho mínimo de caracteres que uma conta de usuário pode conter. Valor entre 0 (desativa) e 14. |
| **Tempo de vida máximo da senha**<br><br>Configuração que determina o período de tempo em dias em que uma senha pode ser utilizada antes de o sistema solicitar sua alteração. Valor 0 desativa o tempo máximo. De 1 a 999 define o espaço de tempo. |
| **Tempo de vida mínimo da senha**<br><br>Configuração que determina o período de tempo em dias em que uma senha deve ser utilizada antes de o usuário poder alterá-la. Valor 0 habilita o usuário a trocar a senha imediatamente. |

33

**6.2 - Diretiva de auditoria**

A auditoria de segurança é uma das ferramentas mais poderosas para ajudar a manter a segurança do sistema. A auditoria deve identificar ataques, bem-sucedidos ou não, que representam algum tipo de ameaça a sua rede ou ataques contra os recursos determinados em sua avaliação de riscos.




**Diretivas de Segurança local**
**Fonte: Internet, 2015.**

Principais opções:

a) Acesso aos serviços de diretório de auditoria;
b) Auditoria de alteração de diretivas;
c) Auditoria de eventos de logon;
d) Auditoria de eventos de sistema;
e) Auditoria de gerenciamento de conta.

| |
|---|
| **Acesso aos serviços de diretório de auditoria**<br><br>Determina se o sistema operacional fará a auditoria das tentativas dos usuários de acessar os objetos do Active Directory. |
| **Auditoria de alteração de diretivas**<br><br>Determina se o sistema operacional fará a auditoria de cada instância de tentativas de alteração da diretiva de atribuição de direitos, diretivas de auditoria, diretivas de contas ou diretivas de confiança do usuário. |
| **Auditoria de eventos de logon**<br><br>Determina a necessidade de o sistema operacional fazer auditoria de cada instância de tentativa de logon ou logoff de um usuário no computador. |
| **Auditoria de eventos de sistema**<br><br>Determina a necessidade de o sistema operacional fazer auditoria dos seguintes itens:<br><br>1) Alteração do horário do sistema;<br>2) Inicialização ou desligamento do sistema; |

| |
|---|
| 3) Carregamento de componentes de autenticação extensível;<br>4) Perda de eventos que passaram por auditoria, devido a falha no sistema de auditoria e<br>5) Se o tamanho do log de segurança exceder o nível de limite de aviso configurável. |
| **Auditoria de gerenciamento de conta**<br><br>Determina a necessidade de o sistema operacional auditar eventos de gerenciamento de contas, tais como criação, alteração e exclusão de contas de usuário e grupos, definição de senhas etc. |

34

**6.3 - Atribuição de direitos de usuários**

Em determinadas situações, não basta apenas ativar ou desativar certas diretivas de segurança. É necessário, em algumas situações, informar quais usuários devem ou não ter acesso às opções de segurança de um sistema.

A atribuição de direitos de usuário determina quais contas ou grupos têm direitos ou privilégios no computador.

Principais opções:

a) Acesso a este computador pela rede;
b) Adicionar estações de trabalho ao domínio;
c) Permitir logon pelos serviços de terminal;
d) Alterar a hora do sistema;
e) Apropriar-se de arquivos ou de outros objetos;
f) Fazer *backup* de arquivos e pastas.

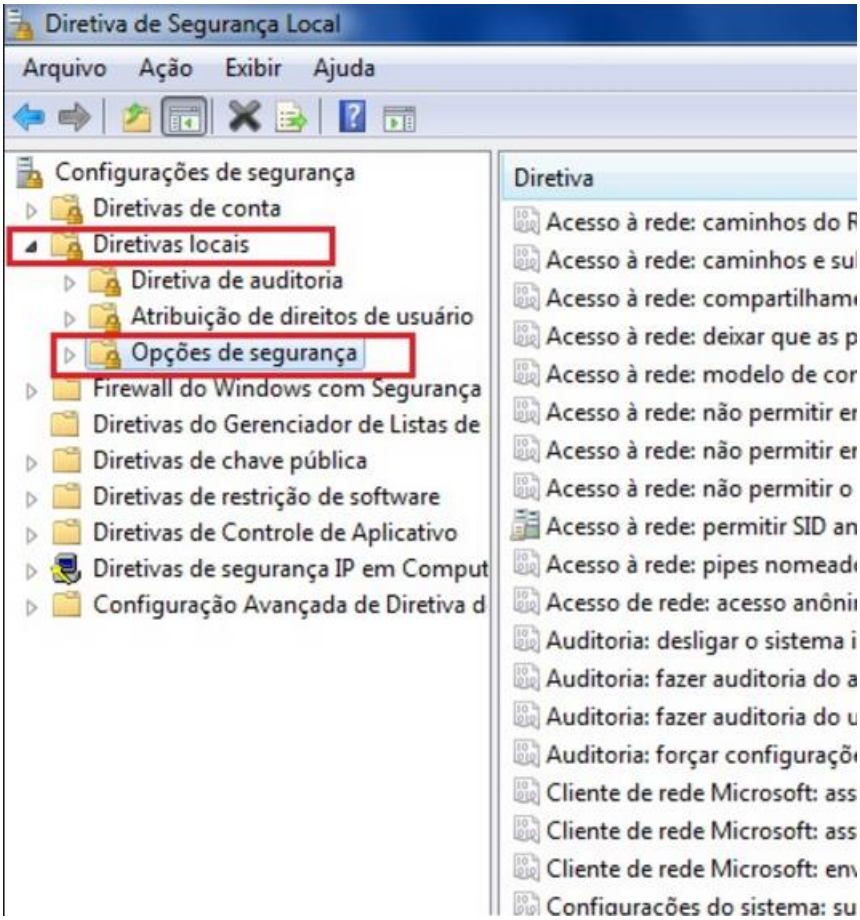
| |
|---|
| **Acesso a este computador pela rede**<br><br>Esse direito de usuário determina quais usuários e grupos têm permissão para se conectar com o computador pela rede. |
| **Adicionar estações de trabalho ao domínio**<br><br>Essa configuração de segurança determina quais grupos ou usuários podem adicionar estações de trabalho a um domínio. |
| **Permitir logon pelos serviços de terminal**<br><br>Essa configuração de segurança determina quais usuários ou grupos têm permissão para fazer logon como um cliente de serviços de terminal. |
| **Alterar a hora do sistema**<br><br>Permite informar quais usuários têm permissão para alterar a data e a hora do computador. |

| |
|---|
| **Apropriar-se de arquivos ou de outros objetos**<br><br>Determina quais usuários podem apropriar-se de objetos protegidos do sistema, incluindo objetos do Active Directory, arquivos ou pastas, impressoras, chaves do registro, processos e segmentos. Por padrão, apenas os administradores possuem essa opção. |
| **Fazer *backup* de arquivos e pastas**<br><br>Direito do usuário que determina os usuários que podem ignorar permissões de diretório, registro e outros objetos persistentes, com a finalidade de fazer *backup* do sistema. Especificamente esse direito de usuário é semelhante a conceder permissões a usuários e grupos para todos os arquivos e pastas do sistema, mesmo que essas pastas tenham permissões diferentes. |

35

### 6.4 - Opções de segurança

Além das políticas e controles de segurança constantes nas diretivas locais, destaca-se, também, o conjunto de opções de segurança. Essas opções complementam as políticas de segurança local.




**Opções de Segurança**
**Fonte: Internet, 2015.**

As opções de segurança são divididas em:

| | |
|---|---|
| Acesso à Rede | Controles DCOM |
| Auditoria | Desligamento |
| Cliented de rede Microsoft | Dispositivos |
| Configurações do Sistema | Logon Interativo |
| Console de Recuperação | Membro do domínio |
| Contas | Objetos do Sistema |
| Controlador de domínio | Segurança de rede |
| Controle da Conta de Usuário | Servidor de rede Microsoft |
| Criptografia | x |

36

Para sua prática, baixe e instale o Windows server 2008 e utilize as técnicas vistas na parte teórica. Modifique a configuração do servidor Windows 2008 de modo a torná-lo mais seguro, atendendo aos itens descritos no texto.

No decorrer do texto, foram mencionados alguns pontos de atenção na configuração de um servidor Windows Seguro, sendo eles:

a) Remover ou desabilitar todos os serviços não necessários no *host*;
b) Remover ou desabilitar todas as contas de usuário não necessárias;
c) Remover ou desabilitar todos os protocolos de rede não utilizados;
d) Configurar adequadamente os registros de log do sistema, para que possam identificar possíveis ataques ou atividade suspeita;
e) Implantar um sistema de detecção de intrusão de *host*;
f) Atualizar o sistema operacional com as últimas correções de segurança disponibilizados pelo fabricante;
*g)* Filtrar todas as portas desnecessárias ao *host*;
*h)* Utilizar conexão criptografada para conectar ao *host*. Para nos auxiliar nesta tarefa, vamos utilizar a ferramenta da MBSA (Microsoft Microsoft *Baseline* Security Analyser).

37

## 7 - RESUMO

Neste módulo foram apresentadas técnicas básicas de configuração segura de servidores Windows, configuração de filtros de pacotes, análise de processos ativos, criação de uma configuração inicial e desabilitação de processos e serviços desnecessários.

Em um ambiente Microsoft Windows é vital utilizar o conceito de bastion *host* (termo aplicado a um *host* que age como um check-point entre a rede interna e a Internet, ou entre sub-redes da Intranet.

Para um Bastion *Host* conectado à Internet maior atenção deve ser dada à segurança - é o ponto mais exposto e, por essa razão, deve ser o mais forte) para garantir a integridade do sistema.

O bastion *host* será uma máquina exposta na rede pública disponibilizando recursos e serviços. Por ser uma máquina com serviços públicos, essa será a primeira barreira a ser vencida por um invasor para tentar obter acesso aos sistemas da rede privada.

Existem várias implementações possíveis de bastion *host*s, de acordo com os serviços que ele oferece. Alguns exemplos:

a) *Firewall gateways*;
b) Servidores web;
c) Servidores FTP;
d) Servidores de nome DNS;
e) Transportadores de e-mail.

**38**

Foi visto, também, a relevância do planejamento da instalação e escrituração das atividades a serem realizadas e auditadas nos servidores públicos:

a) Remover ou desabilitar todos os serviços não necessários no *host*;
b) Remover ou desabilitar todas as contas de usuário não necessárias;
c) Remover ou desabilitar todos os protocolos de rede não utilizados;
d) Configurar adequadamente os registros de log do sistema para que possam identificar possíveis ataques ou atividade suspeita;
e) Implantar um sistema de detecção de intrusão de *host*;
f) Atualizar o sistema operacional com as últimas correções de segurança disponibilizadas pelo fabricante;
g) Filtrar todas as portas que não são necessárias para o *host*;
h) Utilizar conexão criptografada para conectar no *host*;
i) Evitar a instalação de aplicativos não necessários e notadamente vulneráveis, como Flash, PDF Viewers, Java.

Cada serviço de rede presente em um servidor pode escutar uma porta, TCP ou UDP, para receber conexões de outros servidores ou clientes. Alguns desses serviços são importantes para o bom funcionamento do servidor, e nem sempre podem ser desabilitados. Quando verificamos as portas abertas em uma configuração padrão de um servidor Windows, vemos que existe uma série de portas que são abertas por padrão no sistema. Colocar um sistema de forma pública na internet, sem a devida filtragem dos serviços que não estão em uso, é arriscado e pode comprometer a segurança do servidor.

O Windows *Firewall* é o aplicativo que acompanha o sistema operacional para controle de conexões de rede, que normalmente vem configurado por padrão na instalação padrão do Windows.

Um sistema seguro deve utilizar um sistema de arquivos que suporte a criação de permissões, de modo a limitar o acesso dos usuários para minimizar um potencial estrago em caso de comprometimento de uma conta de usuário, além de incluir o mínimo de usuários possíveis.

Além das políticas e controles de segurança constantes nas diretivas locais, destaca-se, também, o conjunto de opções de segurança. Essas opções complementam as políticas de segurança local.

# UNIDADE 4 – REDES PRIVADAS VIRTUAIS, AUDITORIA DE SEGURANÇA DA INFORMAÇÃO E CONFIGURAÇÕES DE SERVIDORES
# MÓDULO 4 – CONFIGURAÇÃO DE SERVIDORES LINUX

01

## 1 – INSTALAÇÃO DO LINUX

Apesar de ser algo considerado no momento de implantação do perímetro, em alguns casos **não é possível realizar um isolamento completo**. Dessa forma, é imperativo que haja um investimento significativo na proteção dos servidores presentes na DMZ.

Essa proteção de servidores é muitas vezes chamada, como visto anteriormente, de **hardening** e envolve tanto configurações seguras quanto a instalação de *software* que aumente a segurança do servidor.

Em momento anterior vimos como configurar um servidor seguro utilizando o sistema operacional Microsoft Windows Server 2008. Neste momento, veremos como configurar um servidor utilizando o sistema operacional Linux. Durante a parte teórica, as informações apresentadas, na medida do possível, serão genéricas, de modo que a configuração seja independente da distribuição Linux utilizada.

Normalmente, quando instalamos um sistema operacional utilizando as opções padrão, uma série de programas e serviços instalados pode ser desnecessária para o propósito do servidor. Dessa forma, é importante ter em mente o papel que o servidor desempenhará, de modo a realizar uma instalação com o mínimo indispensável para o funcionamento do servidor.

Para efetivar o seu aprendizado, é altamente recomendável que seja feita a instalação do Linux em qualquer versão. Clique aqui para baixar.


Vale destacar que os comandos que aparecem ao longo do texto são de versões antigas. Caso você instale uma versão mais recente, muito provavelmente as mensagens decorrentes da instalação serão diferentes, porém com os mesmos objetivos.

**Hardening**

É um processo de mudança na configuração de um servidor com o intuito de torná-lo mais seguro.

**Clique aqui**

http://www.uniriotec.br/~morganna/guia/distribuicao.html.

02

No caso do Debian (2012), existe uma mídia de instalação denominada ***"netinst"***, que possui o mínimo de pacotes para montar um sistema básico. Essa abordagem é interessante, visto que é possível instalar um sistema mínimo e, após, adicionar pacotes para prover as funcionalidades necessárias. Em outros sistemas ou distribuições, normalmente existe uma opção de **instalação avançada**, onde o administrador pode configurar o que será instalado no sistema.

Outra decisão importante refere-se ao particionamento do disco rígido do servidor. Abaixo são listadas algumas **regras** interessantes a observar durante a instalação:

a) Qualquer árvore de diretórios em que um usuário puder escrever, tais como **/home, /tmp**, deve estar em uma partição separada e usar porções do disco. Isto reduz o risco de um usuário encher seu sistema de arquivos e realizar um ataque de negação de serviço.

b) Diretórios de uso comum, tais como **/home e /tmp** podem ser colocados em partições separadas e configurados para não permitir a execução de arquivos (atributo noexec). Na mesma linha, o atributo **nosuid** ignorará o bit de **SUID** e vai tratá-lo como um arquivo normal, impedindo que um script mal configurado seja executado com permissões de outro usuário. Esses atributos são configurados no arquivo **/etc/fstab**.

c) Dados estáticos podem ser colocados em uma partição separada, somente como leitura. Um exemplo é a partição **/etc/**, que após a configuração do servidor, poderia ser montada em uma mídia em formato de somente leitura, como CD-ROM.

03

É comum administradores utilizarem ferramentas de imagens de disco para criar uma imagem de um sistema mínimo já instalado, com os recursos básicos para o bom funcionamento de um servidor. Essa imagem pode tornar-se padrão para a criação de um novo servidor e o ponto de partida para a configuração.

Recursos como **NTP** para sincronismo de tempo, **SSH** para acesso criptografado de administração e configurações como fuso horário, senhas de administração e permissões de acesso devem ser consideradas nessa imagem.

Uma vez definida a imagem, esta pode ser usada para futuras instalações, de modo a reduzir o trabalho de implantar um novo servidor, além ter uma imagem com uma configuração mínima segura.

Em tecnologias de virtualização o uso de imagens pode facilitar e acelerar muito o processo de criação de novos servidores.

| |
|---|
| **NTP**<br><br>Network Time Protocol - é o protocolo de sincronização de tempo na internet. |
| **tecnologias de virtualização**<br><br>Processo de conversão de servidores físicos em servidores virtuais. Estes têm desempenho menor que os servidores físicos, porém vários servidores podem compartilhar os mesmos recursos, de modo que um servidor pode disponibilizar os seus recursos ociosos para outros servidores. |

04

## 2 - Desabilitando Serviços Desnecessários

Em instalações padrão de um sistema operacional, muitos serviços e programas são incluídos sem que sejam necessariamente importantes para o serviço em implantação. Sendo assim, após a instalação do sistema, devemos conferir os processos e serviços executando na máquina de modo a desativar todos os serviços que não sejam indispensáveis para o funcionamento do sistema. Essa é uma tarefa complexa para um iniciante, visto que ele provavelmente não saberá o que cada serviço faz e poderá ter receio de desabilitar alguns serviços.

A regra geral nesses casos é desabilitar todos os serviços que abram alguma porta, TCP ou UDP no sistema, e que não façam parte de um serviço legítimo que se queira oferecer.

Algumas ferramentas auxiliam nessa tarefa, mas o administrador deve ter paciência para pesquisar todos os serviços com o intuito de determinar se estes podem ser desabilitados. Recomenda-se ainda que sejam usadas distribuições Linux ou sistemas Unix que lhe sejam familiares, pois será mais fácil configurá-las.

A seguir serão apresentados alguns comandos úteis.

05

a) **netstat –an | more**: mostra todas as portas abertas no sistema.

As portas em estado LISTEN são portas aguardando conexão. É importante registrar essas portas e posteriormente tentar descobrir o processo associado. Em algumas distribuições, o comando **netstat –anp** mostra o processo responsável por aquela conexão.

Segue abaixo um exemplo da saída do comando **netstat –anp**.

```
LinServer:~# netstat -anp | more
```

Active Internet connections (servers and established)

| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State | PID/Program name |
|---|---|---|---|---|---|---|
| tcp | 0 | 0 | 0.0.0.0:45416 | 0.0.0.0:* | LISTEN | 1541/rpc.statd |
| tcp | 0 | 0 | 0.0.0.0:111 | 0.0.0.0:* | LISTEN | 1530/portmap |
| tcp | 0 | 0 | 172.16.1.10:53 | 0.0.0.0:* | LISTEN | 1765/named |
| tcp | 0 | 0 | 127.0.0.1:53 | 0.0.0.0:* | LISTEN | 1765/named |
| tcp | 0 | 0 | 0.0.0.0:22 | 0.0.0.0:* | LISTEN | 1794/sshd |
| tcp | 0 | 0 | 127.0.0.1:5432 | 0.0.0.0:* | LISTEN | 1814/postgres |
| tcp | 0 | 0 | 127.0.0.1:25 | 0.0.0.0:* | LISTEN | 2096/exim4 |
| tcp | 0 | 0 | 127.0.0.1:953 | 0.0.0.0:* | LISTEN | 1765/named |

**06**

b) **lsof –i <protocolo>:<porta>**: lista o processo associado a uma determinada porta.

Exemplo: lsof –i TCP:25. Nem sempre a ferramenta **lsof** está instalada, então em alguns casos é necessário baixar e instalar a ferramenta. No caso do Debian, o comando **apt-get install lsof** é suficiente.

Considerando o exemplo acima, o comando **lsof –i TCP:111** nos dará o seguinte resultado:

```
LinServer:~# lsof -i TCP:111

COMMAND  PID   USER   FD   TYPE DEVICE SIZE NODE NAME
portmap 1530 daemon    5u  IPv4   4255      TCP *:sunrpc (LISTEN)
```

c) **ps aux | more**: lista todos os processos do sistema.

Processos em execução, que não fazem parte dos serviços que se quer configurar no servidor, devem ser registrados para serem desabilitados. Alguns serviços podem ser importantes para o sistema, de modo que desabilitá-los pode causar comportamento inesperado. Caso o aluno seja inexperiente em Linux ou na distribuição em questão, recomendamos que instale um servidor em uma máquina e que sejam feitos experimentos até que seja desabilitado o máximo de serviços e processos, mantendo os serviços que se deseja oferecer em funcionamento.

d) **man <serviço>**: obtém informações sobre um serviço a partir de suas páginas de manual (man pages).

Observe que nem todo processo é necessariamente um serviço.

Um serviço consiste em um ou mais processos, executados continuamente no servidor com o intuito de oferecer algum serviço para a máquina ou a rede.

**07**

Para desabilitar um serviço, é necessário remover o link simbólico correspondente no diretório referente ao **runlevel** padrão do sistema. Ele pode ser determinado ao se examinar o arquivo **/etc/inittab**, em uma linha como tal como: "id:2:initdefault:".

No exemplo acima, o runlevel padrão é 2, de forma que os serviços que devem ser desabilitados encontram-se no diretório **/etc/rc.d/rc2.d**. Em alguns sistemas, esses serviços podem estar localizados em outro diretório.

Consulte a documentação do sistema em questão para determinar onde eles se encontram. No caso do Debian considerado, os serviços habilitados no diretório /etc/rc2.d são:

```
LinServer:/etc/rc2.d# ls
README    S12acpid  S15lwresd  S19postgresql-8.3  S20nfs-common    S89atd   S91apache2
S99rmnologin
S10rsyslog  S15bind9  S16ssh  S20exim4    S20openbsd-inetd
S89cron  S99rc.local  S99stop-bootlogd
```

Os arquivos nessa pasta são links para scripts de inicialização do serviço no diretório **/etc/init.d**, onde comandos podem ser enviados, como start e stop para iniciar e terminar o serviço, respectivamente.

```
LinServer:/etc/rc2.d# ls -la S10rsyslog
Lrwxrwxrwx 1 root root 17 2010-07-11 20:24 S10rsyslog -> ../init.d/ rsyslog
…  restante da listagem omitida propositadamente……
Lrwxrwxrwx 1 root root 19 2010-07-11 20:28 K05keytable -> ../int.d/keytable
```

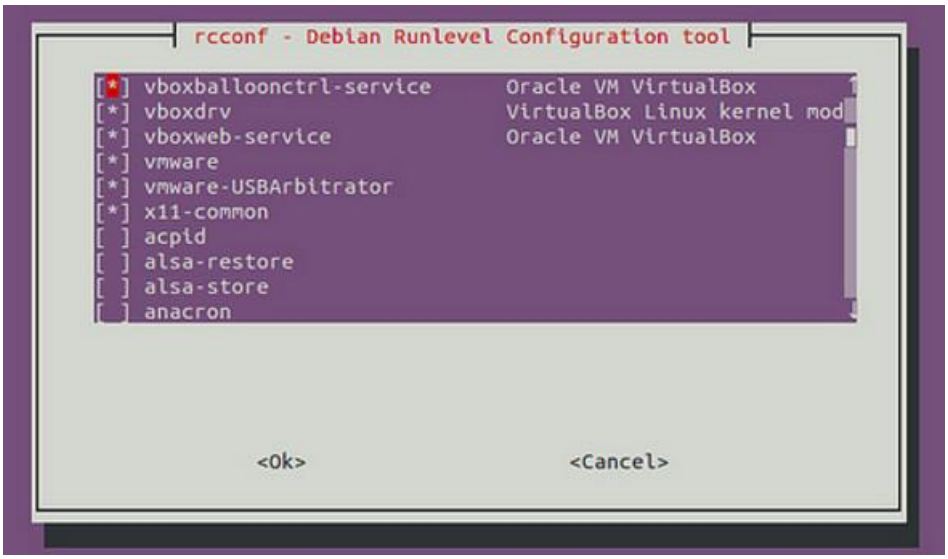
A letra "S" antes do nome do serviço indica que o serviço será iniciado (start). Para finalizar o serviço, há a letra K (kill). O número a seguir indica a prioridade. Serviços com menor número serão iniciados primeiro pelo sistema.

**Runlevel**

Nível de execução que corresponde a um número que indica o modo de exceção em que se encontra um sistema operacional Unix. Por exemplo, o runlevel 1 corresponde a um modo de execução onde só um usuário pode usar o sistema.

08

Outra forma de gerenciar serviços é com o comando **rcconf**. Ele pode ser instalado através do comando **apt-get install rcconf**. Após instalado, basta executá-lo para ser apresentado a um menu bastante prático, onde se pode habilitar e desabilitar os serviços desejados.




**Tela do rcconf**
**Fonte: Internet, 2015.**

O administrador não deve ter receio de desabilitar os serviços. Caso algum serviço seja desabilitado e impeça o sistema de funcionar corretamente, ele pode ser habilitado posteriormente (treine antes!!!). Ao final, uma nova execução do comando **netstat** pode confirmar que os serviços desnecessários foram desabilitados. Alguns serviços utilizam o serviço **inetd** (Internet Super Server) e devem ser desabilitados no arquivo **/etc/inetd.conf**.

Para mais informações sobre o **inetd** (ou **xinetd** em sistemas mais modernos), consulte as páginas de manual correspondentes com o comando **man** (ex.: man inetd.conf). Caso o inetd não seja necessário no seu servidor, ou seja, nenhum serviço esteja configurado nele, ele pode ser desabilitado.


É importante lembrar que desabilitar um serviço não o desinstala do servidor, de modo que um atacante que tenha acesso ao servidor pode iniciar novamente o serviço. Após desabilitar o serviço e ter certeza de que ele não será usado, é importante remover o pacote correspondente do sistema.

09

## 3 - PACOTES E PROGRAMAS

Desabilitar os serviços de rede desnecessários reduz a superfície de ataque ao servidor, que em combinação com regras de filtragem no *firewall*, proveem uma boa camada de proteção. Apesar disso, um atacante pode ainda conseguir comprometer um serviço válido e obter acesso ao servidor comprometido.


Esse fato pode possibilitar ao atacante obter acesso de administrador ou ainda comprometer outros servidores na rede. Para reduzir o que um atacante é capaz de fazer no servidor, este deve possuir um conjunto mínimo de pacotes.

Deixe para desabilitar pacotes ao final da configuração do servidor, pois você pode precisar deles para alguma tarefa administrativa. Em especial, pacotes que proveem ferramentas para o atacante tentar comprometer outros sistemas.

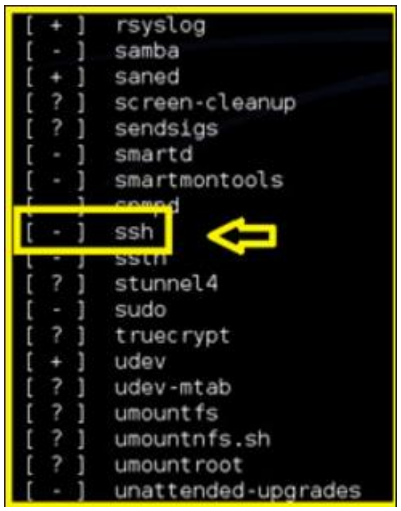
A lista a seguir sugere alguns **pacotes para serem desabilitados**:

a) Compiladores de linguagens (gcc, g++ e javac);
b) Pacotes de monitoramento de conexões de rede (TCPdump, Nmap e netcat);
c) Pacotes de produtividade (editores de texto e planilhas de cálculo), pois um servidor não deve ser usado como estação de trabalho;
d) Ambiente gráfico (X11). Devem ser utilizados os ambientes gráficos das estações para realizar tarefas de configuração. Muitos serviços possuem ambientes de configuração web ou aplicações cliente-servidor, de modo que não é necessário manter o ambiente gráfico instalado. Caso seja indispensável, considere filtrar as portas do XWindows (X11) no seu *firewall* de borda;
**e)** Serviços de rede não criptografados (Telnet, pop3 e Imap). Dê preferência sempre a serviços criptografados (SSH, SFTP, pop3s e imaps). O SSH - Secure Shell - é um serviço criptografado que surgiu para substituir serviços inseguros como o scp, rsh, rcopy e telnet;

10

Desabilitar pacotes é uma tarefa complexa e cansativa por isso, recomenda-se que se parta do inverso, ou seja, realizar uma instalação mínima e ir adicionando pacotes à medida que forem sendo necessários. Ao final da implantação do servidor, devem ser removidos pacotes temporários que porventura tenham sido instalados para a configuração do servidor. Paciência, conhecimento e persistência são princípios fundamentais para essa tarefa.

No Debian, o comando **tasksel** pode ajudar, pois através dele é possível selecionar as funcionalidades de que o servidor disporá, de modo que automaticamente os pacotes correspondentes são adicionados ou removidos.




**Execução de comando Tasksel.**
**Fonte: Internet, 2015.**

Os pacotes podem ser desinstalados através de dois comandos:

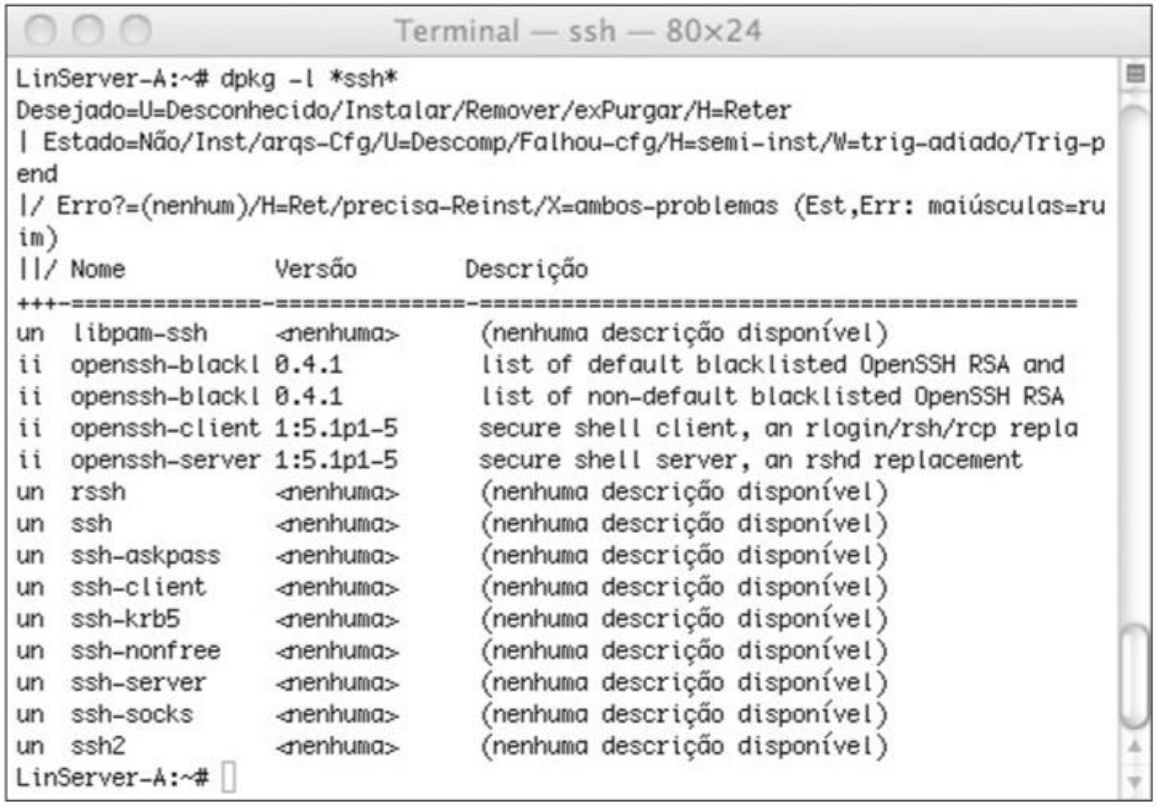
```
apt-get remove <pacote>
dpkg –R <pacote>
```

11

Para remover um pacote, necessitamos primeiro saber o seu nome. Para tanto, utiliza-se algumas ferramentas para procurar o pacote desejado. Como exemplo, imagine que queremos remover o pacote referente ao SSH:

```
dpkg –l *ssh*
```

Esse comando listará todos os pacotes instalados que se referem ao SSH:




**Lista de pacotes instalados no sistema.**
**Fonte: Peixinho, 2013.**

No exemplo acima, os pacotes marcados com ii estão instalados no sistema. Dessa forma, removem-se os pacotes desejados:

```
dpkg –R openssh-client
dpkg –R openssh-server
```

12

## 4 - Configuração Segura de Serviços

Configurar um serviço de forma segura também é complicado e depende do serviço a implantar, porém existem alguns princípios básicos que devem ser observados em qualquer serviço Unix.

Princípios básicos:

a) Usuários sem privilégios;
b) Chroot;
c) Desabilitar funcionalidades desnecessárias;
d) Acessos administrativos;
e) Acessos criptografados;

f) Controle de acesso por estação;
g) Autenticação mais forte;
*h)* Conta de usuário comum (sudo).

**4.1 - Usuários sem privilégios**

Muitos serviços possuem no próprio arquivo de configuração a opção de escolher um usuário para executar o serviço, de modo a ter o direito desse usuário. A ideia é escolher um usuário que tenha um mínimo de direitos sobre o sistema, seguindo o **princípio do menor privilégio**. Caso o serviço não tenha essa opção, ainda assim é possível criar um usuário e executar o serviço com os direitos do usuário, com o comando **su**.

Por exemplo, para executar o serviço **serverd**, poderíamos utilizar o comando: **su serverd_user –c /caminho/serverd**. Alguns serviços necessitam de privilégios especiais e não são capazes de executar como usuários comuns do sistema. Deve-se tomar um cuidado extra com esses serviços, como registrar todos os acessos e eventos do serviço em um servidor de logs seguro.

13

**4.2 - Chroot**

Esse é um recurso presente em sistemas Unix, no qual um determinado processo do sistema enxerga apenas uma subárvore do sistema operacional. Dessa forma, o processo não será capaz de ler, gravar ou executar arquivos fora desta subárvore. Assim como o usuário sem privilégios, o chroot é utilizado por diversos serviços. Normalmente a própria distribuição possui pacotes que instalam o serviço com o recurso.


Configurar um serviço para rodar com esse recurso pode ser uma tarefa complicada, pois deverão ser previstos todos os recursos que o serviço necessita utilizar, de modo que eles devem estar em um diretório acessível, dentro da árvore ao qual o processo foi “empacotado”.

No caso do Debian, procura-se a existência de versão chroot para o serviço a ser instalado, como no exemplo abaixo para um servidor web (Apache):

```
LinServer:~# apt-cache search apache | grep chroot
libapache2-mod-chroot - run Apache in a secure chroot environment
mod-chroot-common - run Apache in a secure chroot environment
```

**4.3 – Desabilitar Funcionalidades Desnecessárias**

Muitos serviços possuem recursos adicionais, como plugins, que podem não ser necessários para o funcionamento do servidor a ser configurado. Dessa forma, o administrador deve analisar cuidadosamente os arquivos de configuração do serviço, de modo a desabilitar qualquer recurso que

seja indesejado. Essa tarefa também não é simples e depende da experiência do administrador em um determinado serviço. **Quanto menos recursos o serviço oferecer, mais seguro ele será**. O difícil é achar o equilíbrio.

Um exemplo de funcionalidade desnecessária seria um servidor www Apache instalado com suporte a PHP, caso não existam scripts em PHP sendo utilizados, eles podem perfeitamente ser desabilitados.

14

**4.4 – Acessos Administrativos**

É comum em servidores existir um acesso administrativo para que o administrador não necessite se deslocar fisicamente até o equipamento para obter acesso. Normalmente se utiliza uma emulação de terminal remoto, como o Secure Shell (SSH), ou um console web, como o webmin. Apesar do acesso remoto ser um recurso prático, deve ser usado com muita cautela, pois pode permitir a um atacante obter acesso privilegiado ao servidor.

Alguns cuidados básicos devem ser tomados ao tratar de acessos administrativos:

a) **Utilizar sempre acessos criptografados para garantir que os dados não serão interceptados em trânsito na rede**. Saiba+

b) **O administrador deve possuir um conjunto de estações definido para acessar o servidor e só deve aceitar conexões dessas estações**. Essa configuração pode ser realizada através de permissões de acesso no próprio servidor ou através de filtros de pacotes. Saiba+

c) **Em caso de servidores públicos (DMZ), nenhum acesso administrativo deve ser permitido diretamente a partir da internet**. Administradores fora da organização devem utilizar canais seguros, como VPN, para administrar os servidores sob a sua responsabilidade.

d) Caso seja viável, pois envolve custo financeiro, deve-se utilizar uma **autenticação mais forte**, como tokens e certificados digitais para acessos administrativos nos servidores e em conexões VPN.

**Saiba + (Utilizar sempre acessos criptografados para garantir que os dados não serão interceptados em trânsito na rede.)**

Protocolos como Telnet, RSH, RCP e Xwindows devem ser substituídos por versões seguras, como SSH. Acessos remotos devem ser realizados através de recurso VPN com criptografia. Os Consoles de administração web devem sempre utilizar o HTTP seguro (HTTPS).

**Saiba+ (O administrador deve possuir um conjunto de estações definido para acessar o servidor e só deve aceitar conexões dessas estações.)**

Caso haja a necessidade de acesso de muitos locais distintos, pode-se configurar um servidor com apenas o serviço de acesso administrativo e utilizar esse servidor para acessar os demais.

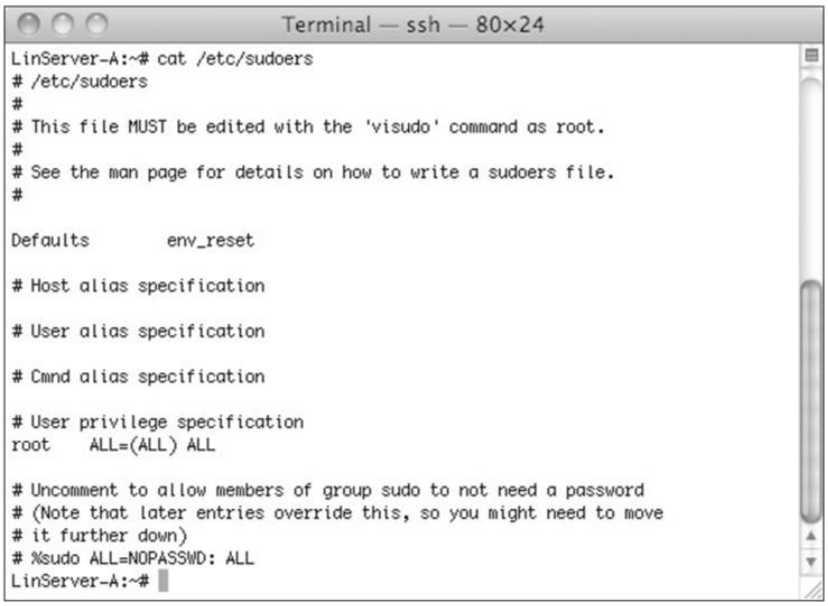
**Tokens**

São dispositivos de segurança que armazenam chaves de cripgtografia e certificados digitais. Algo que o usuário possua para comprovar sua identidade.

15

e) **Utilizar sempre uma conta de usuário comum para acesso, utilizando posteriormente o comando *su* ou sudo para obter acesso de administrador**. Dessa forma, ficará no sistema o registro do administrador que realizou determinado acesso. Caso seja possível, devem-se criar contas de administração restritas para administradores que realizam tarefas específicas. O comando **sudo** permite que seja dado acesso de administrador apenas a alguns comandos, de modo que o usuário pode executar **sudo <comando>,** para executar um comando autorizado como administrador. O arquivo **/etc/sudoers** contém a configuração do serviço **sudo**. Caso o **sudo** não esteja instalado, o comando **apt-get install sudo** é suficiente para instalar. Mais informações sobre o **sudo** podem ser obtidas com o comando **man sudo**.

O exemplo abaixo mostra uma configuração de **sudo**, de modo a permitir que o usuário peixinho.icp possa acessar qualquer comando como administrador.




**Configuração de sudo.**
**Fonte: Peixinho, 2013.**

Atenção para o parâmetro **PASSWD**, pois caso seja **NOPASSWD**, o usuário poderá executar tarefas como administrador sem a necessidade de entrar com a senha do seu usuário.

16

f) **Evite que um usuário possa realizar o primeiro login no sistema como root.** Tal configuração obriga o usuário a autenticar com um usuário comum e, depois, utilizar o sudo para se tornar administrador do sistema. Para ativar esse recurso, basta apagar o conteúdo do arquivo **/etc/securetty**. Observe que alguns sistemas não utilizam o sistema de autenticação do Linux (PAM) para autenticação (ex.: OpenSSH), assim será necessário verificar se essa opção está disponível na aplicação.

g) **Não permita que qualquer usuário possa se tornar root utilizando o sudo**. O PAM permite que você crie um grupo especial e que somente membros desse grupo possam se tornar root. Essa configuração vai impedir que um usuário comum, mesmo sabendo a senha do root, possa se promover como root do sistema. Saiba+

h) **Proteja o sistema de inicialização do Linux (GRUB) com senha**. Sem essa proteção, um atacante que tenha acesso físico à máquina poderá reiniciar o sistema como root utilizando um modo conhecido como Single User Mode. Para desativar essa opção, edite o arquivo **/boot/grub/menu.lst** e insira no final a linha "password mudeme".

i) Um problema simples no Linux é que, se um atacante tiver acesso ao console e apertar as teclas CTRL + ALT + DEL (utilizadas para autenticar em máquinas Windows), reiniciará, automaticamente, o servidor. Para evitar esse comportamento, comente a linha "# ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now" no arquivo **/etc/inittab**.

**Saiba+**

Para evitar esse comportamento, você deverá criar um grupo chamado wheel (# groupadd wheel) e modificar o arquivo **/etc/pam.d/su** (inserindo no final do arquivo a linha "auth required pam_wheel.so group=wheel"), de forma a permitir que apenas os membros desse grupo possam se tornar root do sistema. Não se esqueça de colocar um usuário como membro do grupo wheel (# usermod –G wheel usuario) antes de realizar o logoff.

17

## 5 - Ferramentas de Segurança de Servidores

Existe uma série de ferramentas que podem aumentar a segurança de um servidor. As mais comuns são os **HIDS**, vistas anteriormente. Existem ainda ferramentas que realizam uma série de mudanças no sistema, com o intuito de torná-lo mais seguro. Dentre estas, destacam-se as ferramentas **Bastille Linux** e **Security Enhanced Linux**, disponíveis na internet e gratuitas.

Essas ferramentas não serão tratadas neste curso, por serem ferramentas avançadas. Se tiver interesse, existem diversas guias na internet sobre como instalar essas ferramentas.

### 5.1 - Testes de Configuração e Auditoria

Testar se uma configuração de um servidor está suficientemente segura é uma fase importante do processo, pois permite verificar se a configuração realizada realmente aumentou a segurança do servidor. Ferramentas de auditoria como Nmap podem ser usadas no servidor para verificar o nível de segurança atingido.


Essas auditorias devem ainda ser realizadas periodicamente, para verificar a aplicação de atualizações de segurança fornecidas pelo fabricante (patches) e se alguma configuração específica causou impacto na segurança do servidor.

18

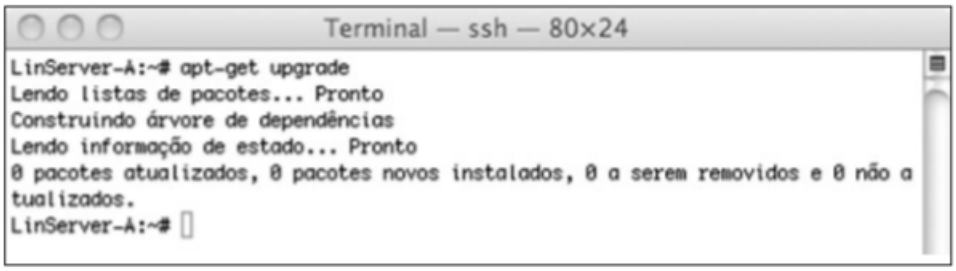
**5.2 - Atualização do sistema operacional**

É fundamental em qualquer sistema operacional a instalação das correções fornecidas pelo fabricante. No caso do Linux, cada distribuição possui uma forma diferente de instalar essas atualizações.

No Debian, mantém-se o sistema atualizado com apenas dois comandos:

```
apt-get update
apt-get upgrade
```

Um exemplo é o da figura a seguir.




**Exemplo de uso do comando de atualização.**
**Fonte: Peixinho, 2013.**

Veja que, em alguns casos, a quantidade de atualizações é grande. No exemplo acima, serão baixados vários MB de atualizações. Caso sua distribuição não seja Debian, verifique na documentação disponível como fazer para atualizar os pacotes. O Debian permite ainda atualizar a versão instalada, caso uma nova versão seja lançada. Para tal, normalmente usa-se o comando **apt-get dist-upgrade**, porém existem alguns passos que devem ser executados.

19

### 5.3 - Pacotes compilados

Em alguns casos, a versão desejada de um determinado *software* não está disponível na distribuição que se usa ou o *software* em si não foi empacotado pelo distribuidor. Nesses casos, é muito comum o administrador baixar o código-fonte do *software* e compilá-lo no próprio servidor. Alguns usuários mais avançados baixam inclusive o núcleo do sistema (kernel), para compilar e instalar um kernel customizado.


Deve-se tomar cuidado com a instalação de *software* compilado, pois eles podem se confundir com os pacotes instalados no sistema, tornando difícil a desinstalação depois. Deve-se sempre instalar *software* compilado em diretórios distintos, como o **/usr/local** ou **/opt**. Deve-se ainda fazer um controle dos arquivos instalados pelo *software*, para facilitar a desinstalação

As ferramentas de compilação (gcc, g++ e outras) podem ser usadas por um atacante para compilar seu próprio *software* malicioso, então devem ser removidas após o uso.

Por fim, a instalação de um kernel customizado é uma tarefa complexa, mas que possui a vantagem de gerar um kernel mais leve e com menos recursos (princípio do menor privilégio), de modo que uma vulnerabilidade encontrada em um recurso do kernel, que não esteja sendo utilizado pelo seu servidor, pode não afetar um kernel customizado sem esse recurso. O problema é que, a cada atualização do kernel, o administrador terá de recompilá-lo, o que leva tempo e pode sobrecarregar a administração no caso de um ambiente com diversos servidores. Recomenda-se examinar a página “The Linux Kernel HOWTO”, pois lá existem muitas informações úteis a respeito.

20

### 5.4 - Sistema de arquivos proc

Muitos parâmetros de kernel podem ser alterados através do sistema de arquivos **/proc** ou usando **sysctl**. Tais alterações podem aumentar o desempenho e segurança geral do sistema.

A seguir são listados alguns parâmetros interessantes (no formato para o arquivo /etc/sysctl.conf), que deverão ser avaliados caso a caso antes de serem implementados em ambiente de produção:

| | |
|---|---|
| net.ipv4.ip_forward = 0 | O encaminhamento IP entre placas só é necessário em servidores Linux que atuarão como roteador entre diferentes redes. |
| net.ipv4.icmp_echo_ignore_all=1 | Evita que a máquina responda a qualquer tipo de ICMP. |

| | |
|---|---|
| net.ipv4.icmp_echo_ignore_broadcasts=1 | Previne o ataque de smurf. |
| net.ipv4.conf.all.accept_source_route=0 | Não aceite pacotes de fonte roteada. Atacantes podem usar fontes roteadas para gerar tráfego, fingindo vir de dentro de sua rede. |
| net.ipv4.conf.all.accept_redirects=0 | Redirecionamento de ICMP pode ser usado para alterar tabelas de roteamento na máquina alvo. |
| net.ipv4.icmp_ignore_bogus_error_responses=1 | Proteção contra mensagens de erro ICMP falsas. |

Finalmente, para terminarmos a presente unidade, pesquise na internet sobre esse assunto, pois existem inúmeros parâmetros que, se alterados, podem aumentar o desempenho de uma aplicação. Uma última dica: procure por “tunning tcp/ip” na web e boa sorte!

21

## 6 - Resumo

Apresentamos aqui algumas técnicas de configuração segura de servidores Linux, desde a sua instalação até a publicação do servidor na internet e realizar uma aplicação prática dos conhecimentos.

Normalmente, quando instalamos um sistema operacional utilizando as opções padrão, uma série de programas e serviços instalados pode ser desnecessária para o propósito do servidor. Dessa forma, é importante ter em mente o papel que o servidor desempenhará, de modo a realizar uma instalação com o mínimo indispensável para o funcionamento do servidor.

Decisão importante refere-se ao particionamento do disco rígido do servidor. Abaixo são listadas algumas regras interessantes a observar durante a instalação:

a) Qualquer árvore de diretórios em que um usuário puder escrever, tais como **/home, /tmp**, deve estar em uma partição separada e usar porções do disco. Isto reduz o risco de um usuário encher seu sistema de arquivos e realizar um ataque de negação de serviço;
b) Diretórios de uso comum, tais como **/home** e **/tmp** podem ser colocados em partições separadas e configurados para não permitir a execução de arquivos (atributo **noexec**). Na mesma linha, o atributo **nosuid** ignorará o bit de **SUID** e vai tratá-lo como um arquivo normal, impedindo que um script mal configurado seja executado com permissões de outro usuário. Esses atributos são configurados no arquivo **/etc/fstab**;
c) Dados estáticos podem ser colocados em uma partição separada, somente como leitura. Um exemplo é a partição **/etc/**, que após a configuração do servidor, poderia ser montada em uma mídia em formato de somente leitura, como CD-ROM.

Em instalações padrão de um sistema operacional, muitos serviços e programas são incluídos sem que sejam necessariamente importantes para o serviço em implantação.

22

Alguns comandos estudados:

a) **netstat –an | more;**
b) **lsof –i <protocolo>:<porta>;**
c) **ps aux | more;**
d) **man <serviço>.**

Desabilitar os serviços de rede desnecessários reduz a superfície de ataque ao servidor, que em combinação com regras de filtragem no *firewall*, proveem uma boa camada de proteção.

Deixe para desabilitar pacotes ao final da configuração do servidor, pois você pode precisar deles para alguma tarefa administrativa. Em especial, pacotes que proveem ferramentas para o atacante tentar comprometer outros sistemas. A lista dos pacotes estudados foram:

a) Compiladores de linguagens (gcc, g++ e javac);
b) Pacotes de monitoramento de conexões de rede (TCPdump, Nmap e netcat);
c) Pacotes de produtividade (editores de texto e planilhas de cálculo), pois um servidor não deve ser usado como estação de trabalho;
d) Ambiente gráfico (X11). Devem ser utilizados os ambientes gráficos das estações para realizar tarefas de configuração. Muitos serviços possuem ambientes de configuração web ou aplicações cliente-servidor, de modo que não é necessário manter o ambiente gráfico instalado. Caso seja indispensável, considere filtrar as portas do XWindows (X11) no seu *firewall* de borda;
e) Serviços de rede não criptografados (Telnet, pop3 e Imap). Dê preferência sempre a serviços criptografados (SSH, SFTP, pop3s e imaps). O SSH - Secure Shell - é um serviço criptografado que surgiu para substituir serviços inseguros como o scp, rsh, rcopy e telnet;
Foi visto que configurar um serviço de forma segura também é complicado e depende do serviço a implantar, porém existem alguns princípios básicos que devem ser observados em qualquer serviço Unix.

Princípios básicos sugeridos a serem seguidos foram:

a) Usuários sem privilégios;
b) Chroot;
c) Desabilitar funcionalidades desnecessárias;
d) Acessos administrativos;
e) Acessos criptografados;
f) Controle de acesso por estação;
g) Autenticação mais forte;
h) Conta de usuário comum (sudo).

Testar se uma configuração de um servidor está suficientemente segura é uma fase importante do processo, pois permite verificar se a configuração realizada realmente aumentou a segurança do servidor.

É fundamental em qualquer sistema operacional a instalação das correções fornecidas pelo fabricante. No caso do Linux, cada distribuição possui uma forma diferente de instalar essas atualizações.