# Conexão segura com *Tunneling*

Nesse artigo será tratado de maneira prática e didática *tunelling ssh e udp.* Para entendimento sobre tunneling será illustrado a situação ficticia:

Em uma rede corporativa em que a **máquina A** (que está protegida por firewall) deseja-se comunicar-se com a **máquina B** que está fora da rede.

Basicamente o processo feito é esse:

**Na conexão de Ida:**

A—> Firewall/Proxy —> Internet —>B

**Na conexão de Volta:**

B —> Internet —> Firewall/Proxy —> A

Conexão OK. Sem segredos =)

Isso seria muito bom em uma conexão normal, **SEM REGRAS RESTRITIVAS** DE **FIREWALL** OU **PROXY**

Agora… Imaginem a seguinte situação:

**Na conexão de Ida:**

A—> Firewall/Proxy

* Firewall não está deixando a **Máquina A** acessar a **Máquina B**, o pedido para no firewall e sequer vai para a internet.

Um exemplo concreto?

Quando de dentro do seu trabalho você tenta acessar o site:

http://www.siteproibido.com

E aparece uma mensagem de que não foi possível acessar o site, tempo de conexão esgotado, etc.

## O que vem a ser Tunneling?

*Tunneling* ou tunelamento é uma técnica em que se cria um túnel entre dois hosts (máquinas) remotos. No *tunneling* é possível que dois hosts (ou máquinas) remotos comuniquem-se perfeitamente por intermédio de um túnel. Mas como assim?

Voltando ao nosso exemplo da **Máquina A** e **Máquina B**

A **Máquina A** está protegida por um firewall, e não consegue comunicar-se com a **Máquina B**.

Através do t*unneling* a **Máquina A** comunica-se perfeitamente com a **Máquina B**.

**Na conexão de Ida:**

A—> Firewall/Proxy —> Internet —>B

A =====Túnel através da internet===== B

* A **Máquina A** conecta-se com a **Máquina B** através de um túnel criado.

**Na conexão de Volta:**

B —> Internet —> Firewall/Proxy —> A

B =====Túnel através da internet===== A

* A **Máquina B** conecta-se com a **Máquina A** através do túnel criado.

Esse túnel apenas repassa a informação entre as duas extremidades (**Máquina A** e **Máquina B**). Como se o Firewall não existisse. Mas o firewall existe, apenas estamos burlando regras do Firewall.

## Porque usá-lo?

Com o tunelamento os seus dados serão passados da **máquina A** para a **máquina B**, através de uma conexão criptografada e SEGURA (SIM EU DISSE SEGURA !), que é feita utilizando-se por base o SSH. O ssh permite a criação do túnel e transmissão dos dados de uma ponta até a outra. Quando os seus dados estão CRIPTOGRAFADOS, alguns ataques como *Sniffing* ou **MITM**, já não são mais possíveis, de tal forma que somente a **máquina A** e a **máquina B** trocam as informações, mantendo seus dados sob sigilo e segurança.

Então por exemplo o site http://siteproibido.com que estava inacessível torna-se acessível pelo fator de que a conexão está criptografada e sobre um túnel criado.

## Mas como criar o *Tunneling*?

Nosso ambiente de teste será um Windows XP que ficará como servidor para *tunneling* (**máquina A**) e uma máquina Linux (**máquina B**, em uma rede diferente da **máquina A**) que ficará como cliente, porém o mesmo experimento pode-se ser feito utilizando-se uma máquina Windows x Windows Linux x Linux, fundamentalmente serão necessários:

* Servidor SSH

* UDPTunnel

http://code.google.com/p/udptunnel/downloads/list

Recomendo o OpenSSH para Windows:

http://sourceforge.net/projects/sshwindows/files/OpenSSH%20for%20Windows%20-%20Release/

### Importante

Será necessário também ativar o roteamento no seu roteador (Port Forwarding, Virtual Server ou DMZ) para o endereço IP interno da máquina Windows e para as portas UDP 4444 e SSH 22. Fica como lição de casa aprender a fazer essa configuração. HOHO =)

Vamos criar o túnel de duas formas.

## Primeiro Modo – SSH Tunneling

Siga as instruções do manual do OpenSSH e inicie o serviço de SSH no Windows com o comando net start opensshd

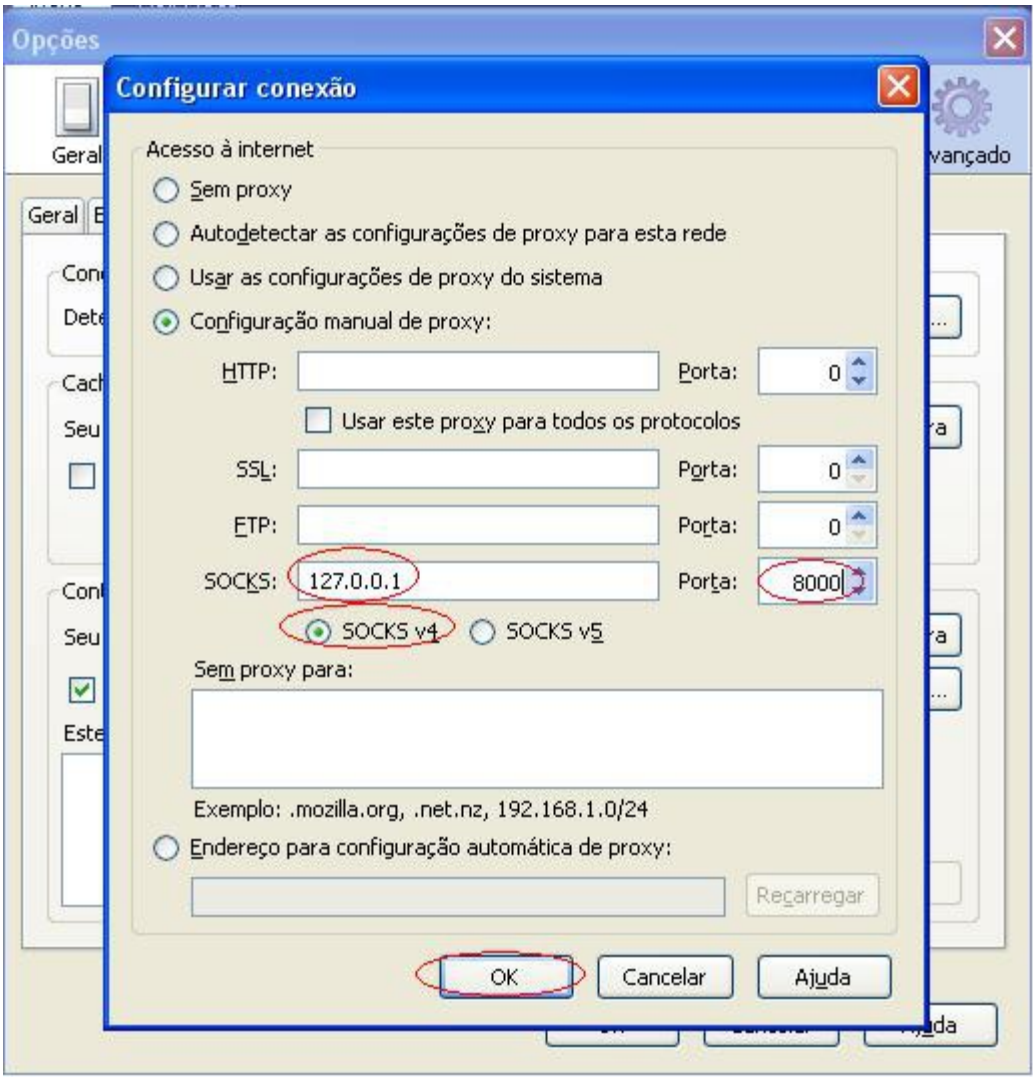
Na **máquina Linux**

**root@linux:~#** ssh -l nome_do_seu_usuário_do_windows IP_WLAN_Windows -D 8000

O IP WLAN (IP Público) pode ser obtido em http://www.meuip.com.br

Digite a Senha do Usuário Windows e após entrar no seu servidor SSH, a sua máquina local cria um SOCKS (que fica na escuta na porta 8000)

Para usar o túnel, abra seu navegador predileto e em configurações da conexão de rede, escolha a opção SOCKS4 com os endereços 127.0.0.1 8000 assim:




**PS**: Estou usando o Firefox

Pronto! Seu túnel já está feito e seus dados já estão criptografados. Agora é só navegar pelo site que estava proibido.

## Segundo Modo – UDP *tunneling*

Faça o download da ferramenta UDPTunnel (já citado no artigo) e faça a sua instalação no Windows e no Linux

Na **Máquina Windows**, digite no prompt de comando:

udptunnel.exe -v -s 4444

Ficará listando na porta UDP 4444 como servidor do túnel.

E na **Máquina linux**

**root@linux:~#** ./udptunnel –v -c 127.0.0.1 44 IP_WLAN_SERVIDOR_WINDOWS 4444 127.0.0.1 22

Pois bem, no comando cliente, você está dizendo ao **udptunnel** que atue como cliente ( -c) fique listando no IP 127.0.0.1 na porta 44, conecte-se ao IP_WLAN_SERVIDOR_WINDOWS na porta 4444. Feito isso o túnel é criado, e quando é acessado você será redirecionado ao IP 127.0.0.1 na porta 22.

Ou seja, eu conecto-me em 127.0.0.1 na porta 44, o udptunnel fecha o túnel com IP_WLAN_SERVIDOR_WINDOWS na porta 4444 e eu acesso o endereço 127.0.0.1 na porta 22 como se a minha máquina Linux fosse a máquina Windows. Faça um teste: Instale o apache no Windows e rode  o comando:

**root@linux:~#**  ./udptunnel –v -c 127.0.0.1 44 IP_WLAN_SERVIDOR_WINDOWS 4444 127.0.0.1 80

No seu navegador no Linux acesse http://127.0.0.1:44

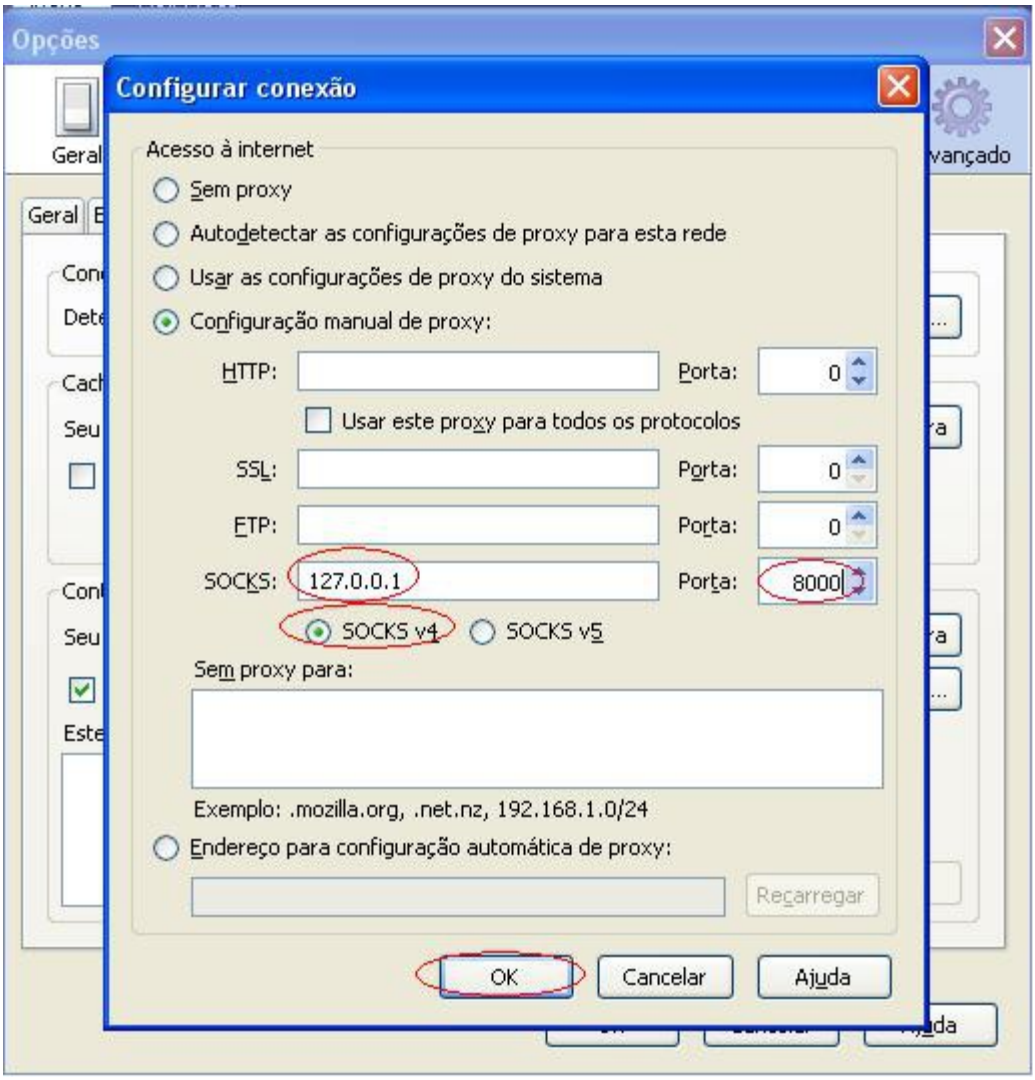
Aparece **IT WORK'S** DO APACHE QUE ESTA NO WINDOWS FORA DA REDE !

Feito o comando **root@root#** ./udptunnel -c 127.0.0.1 44 IP_WLAN_SERVIDOR_WINDOWS 4444 127.0.0.1 22

**Digite em outro terminal linux:**

**root@linux:~#** ssh -l usuario_windows -p 44 127.0.0.1 -D 8000

Entre com sua senha do servidor SSH windows e configure o seu browser da seguinte forma:




Pronto! Você já está usando o **UDP Tunneling**

Há outras maneiras de tunneling como o **ICMP, HTTP** e o **DNS** *Tunneling*, mas o foco do artigo é apenas de mostrar o básico sobre tunneling e como realizá-lo.

Grato,

Daniel Henrique Negri Moreno (a.k.a. W1ckerMan)

Contato:

danielhnmoreno@gmai.com