Pontífice Universidade Católica do Rio Grande do Sul

# Relatório
# Túnel ICMP

Turma: 128

Nomes: João Carlos O. Pereira, Rafael Zalamena

# Introdução

O trabalho que nos foi proposto para a disciplina de Redes de Computadores II de 2011/2 é a programação de um túnel ICMPv6, utilizando o protocolo TCP com o envio de pacotes do tipo raw. Um túnel ICMP é um método no qual pacotes do tipo ICMP "echo" são mandados de uma máquina cliente para uma máquina proxy, sendo que estes pacotes contém dentro de si dados que não são do protocolo ICMP. Normalmente, um firewall impediria a passagem de mensagens TCP com dados, porém ao inserir nas mensagens ICMP estes dados, estas passaram pelo firewall sem que este as barre.

Para isto, é necessário o uso de três máquinas: uma para servir como o cliente, que enviará as mensagens, outro como servidor proxy, que irá recebê-las, e uma terceira que agirá como firewall, interceptando as mensagens enquanto deixando que as mensagens ICMP sejam repassadas à máquina agindo como servidor proxy.

Para que as máquinas cliente e servidor montem, enviem e recebam os pacotes ICMP era necessário se escrever um programa que utilizassse bibliotecas para uso de sockets, funções IPv6, assim como as bibliotecas "arpa/inet" e "linux" que permitem manipular os endereços e tipos de pacotes.

Para que as máquinas repassassem suas mensagens para aquela agindo como firewall, foi necessário modificar suas tabelas de roteamento para que estas repassassem as mensagens para a máquina do firewall.

## Desenvolvimento

O trabalho foi desenvolvido da forma mais clara e intuitiva, para que sirva de demonstração do poder de um socket raw. A montagem do pacote foi feita em etapas, possibilitando o código tomar uma forma mais próxima de pseudocódigo para quem o lê.

O programa resultante da compilação do código pode agir tanto como cliente quanto como proxy, a escolha dos tipos de mensagens é feita internamente e só necessita que se passe os parâmetros certos.

Todo o programa é centrado numa estrutura de contexto que contêm todos os itens necessários para montar um pacote ou guardar os endereços dos dados de entrada do usuário formatados. Dentro dele é encontrado: parâmetros de entrada do programa, informações da interface por onde o pacote irá sair ou entrar, buffer do pacote recebido, buffer do pacote enviado, marcadores de posição do pacote (posição atual, posição do campo packet lenght do IPv6, etc..), comportamento do programa (cliente ou proxy) e porta do pacote TCP.

Cada passo do envio do pacote no socket raw pode ser visto pelas seguintes funções em ordem:

- Main – Decide qual o comportamento: cliente ou proxy.
- Associa_interface – Pega índice da interface por ioctl e a armazena na struct de contexto.
- Inicializa_socket – Zera estrutura de contexto e inicializa um novo socket raw.
- Macstr2bin – Recebe um mac da entrada do usuário, válida e tranforma em binário.
- Constrói_header – Constroi headers de ethernet, IPv6, ICMPv6 e os coloca no buffer de pacote da estrutura de contexto.
- Constrói_header_link – Preenche a estrutura do sockaddr que escolhe a interface por onde irá sair ou receber o pacote raw.
- Constrói_header_tcp – Constrói header TCP que irá ser usado somente para o cliente antes de colocar a mensagem.
- Constrói_mensagem – Recebe uma mensagem por parâmetro, atualiza o campo “packet length” do IPv6 do pacote, e cria checksum do header ICMPv6 do pacote.
- Manda_mensagem – Envia efetivamente a mensagem pelo socket.
- Le_mensagem – Após receber resposta do proxy, o cliente parseia a mensagem para descobrir se interessa e imprimi a resposta.

As funções do proxy, em ordem:

- Run_proxy – Roda o código do proxy, recebe mensagem e responde para o cliente, e talvez, futuramente, desmonta parte ICMPv6 do pacote e o repassa para o destino.
- Le_mensagem – Após receber a mensagem no run_proxy, ele chama esta função para parsear o conteúdo da mensagem, filtrar para descobrir se ela interessa ao proxy, pois o proxy recebe todas mensagens que entram na interface de rede por usar socket raw sem filtro.

- Responde_mensagem – Gera uma mensagem de resposta para o cliente e a envia.

## Utilização

Através da linha de comando, após invocar o arquivo executável, existe uma sequência de informações a ser inserida nas duas máquinas, do contrário o programa não poderá prosseguir. Estas informações devem ser passadas na sequinte ordem e com as seguintes indicações:

Para máquina rodando como cliente –> -i “interface para onde será enviado pacote” -s “mac da origem” -d “mac do destino” -o “endereço IPv6 do destino” -w “endereço IPv6 da origem” -p “porta utilizada para comunicação”

Para máquina rodando como proxy -> -c -i “interface para onde será enviado pacote” -s “mac da origem” -d “mac do destino” -o “endereço IPv6 do destino” -w “endereço IPv6 da origem” -p “porta utilizada para comunicação”

Para a máquina rodando como gateway -> sh ./iptables.sh