Redes de Computadores I

Aula 01 - Captura de Pacotes

Apresentação

Nesta primeira aula, iremos estudar sobre captura de pacotes, que é um assunto que lhe será bastante útil, tanto durante o curso, como na sua vida real, quando estiver utilizando uma rede de computadores. Assim, podemos dizer que captura de pacotes, como o próprio nome diz, consiste em obter uma cópia dos pacotes transmitidos e recebidos pelos equipamentos, de modo que se possa analisá-los em detalhes.

A análise dos pacotes transmitidos/recebidos pelos equipamentos é o método mais utilizado para identificar qualquer problema em uma rede real.

Vamos estudar duas ferramentas de captura de pacotes (uma em modo gráfico e outra em modo texto) que são usadas pela maioria dos administradores de rede.

Vídeo 01 - Apresentação

Objetivos

Após estudar esta aula, você será capaz de:

- Utilizar a ferramenta Wireshark para capturar pacotes na rede.
- Utilizar a ferramenta Tcpdump para capturar pacotes na rede.
- Entender a forma como os pacotes são exibidos, de modo que possa interpretar as informações mostradas nos pacotes capturados.
- Identificar em que máquina da rede executar a ferramenta de captura de pacotes.

Captura de Pacotes

Como você estudou na disciplina de Sistemas de Conectividade, quando se deseja transmitir uma mensagem pela rede, cada camada TCP/IP na máquina de origem acrescenta um cabeçalho a essa informação. Na máquina de destino as camadas equivalentes retiram esses cabeçalhos.

Você viu também que o conjunto *cabeçalho* + *mensagem* de cada camada recebe um nome. Na camada de enlace o chamamos de *quadro*, na camada de rede de *pacote*, na camada de transporte de segmento e na camada de aplicação de mensagem.

Nesta aula, vamos estudar uma ferramenta que permite capturarmos os quadros transmitidos e recebidos por uma placa de rede.

> Como um quadro contém um pacote que, por sua vez, pode conter um segmento, que, por sua vez, pode conter uma mensagem, essa ferramenta é genericamente chamada de *ferramenta de captura de **pacotes*** , e não captura de quadros.

Desse modo, as ferramentas de captura de pacotes nos mostram informações de todas as camadas do modelo TCP/IP que estão presentes no pacote capturado.

Suponha, por exemplo, que você capturou um pacote gerado pela aplicação *ping* em uma máquina cuja placa de rede é ethernet. Como a informação enviada pelo *ping* é uma mensagem do protocolo ICMP, que é transmitida dentro de um pacote IP, que, por sua vez, é transmitido dentro de um quadro Ethernet, a ferramenta de captura irá lhe mostrar informações sobre os cabeçalhos: Ethernet, IP, e ICMP.

Como outro exemplo, suponha que você capturou informações transmitidas entre um browser e um servidor web. Como a web utiliza o protocolo HTTP (um protocolo de aplicação), que é transmitido dentro de segmentos TCP (protocolo de

transporte), e cada segmento TCP é transmitido dentro de um pacote IP, que é transmitido dentro de um quadro Ethernet, a ferramenta de captura irá lhe mostrar informações sobre os cabeçalhos: Ethernet, IP, TCP, e HTTP.

> Ressaltamos que a ferramenta de captura de pacotes mostra as informações dos cabeçalhos, mas, naturalmente, ela mostra todos os dados contidos no pacote.

Lembre-se também que embora o comportamento padrão de uma placa de rede seja descartar todos os quadros que não são destinados a ela, nem ao endereço de broadcast, pode-se configurar a placa para receber todos os quadros, o que caracteriza o modo promíscuo. Como já visto em Sistemas de Conectividade, isso se chama modo promíscuo, ou modo espião. Quando executamos uma ferramenta de captura de pacotes, normalmente, ela trabalha utilizando o modo espião. Evidentemente, para poder usar esse modo, é necessário que o usuário seja administrador da máquina.

Uma questão importante é: em que máquina você vai executar a ferramenta de captura de pacotes? Isso vai depender da finalidade para a qual você está executando a ferramenta. Mas lembre-se que os dados devem passar por essa máquina para que possam ser capturados.

Supondo que você pretende analisar uma comunicação entre as máquinas A e B, as alternativas são:

I. Na máquina A.

II. Na máquina B.

III. Caso a rede onde A (ou B) se encontra use um hub, em alguma outra máquina ligada no mesmo hub.

IV. Em um switch no caminho entre A e B. Mas configurando a porta do switch onde a máquina de monitoramento estará ligada para receber todo o tráfego.

V. Caso A (ou B) esteja em uma rede sem fio, nessa mesma rede sem fio (desde que a rede não esteja usando criptografia).

VI. Em um roteador no caminho entre A e B.

Veja aqui a explicação, em vídeo, sobre a questão da captura de pacotes

Vídeo 02 - Captura de Pacotes

Programas para Captura de Pacotes

A seguir, iremos estudar as duas ferramentas (programas) mais utilizadas para captura de pacotes. Começaremos pelo Wireshark, que possui uma interface gráfica, e depois estudaremos o tcpdump, que é utilizado apenas através da linha de comando.

Wireshark

O Wireshark é um dos programas de captura de pacotes mais utilizados. Ele mostra os dados e as informações dos cabeçalhos dos protocolos contidos nos pacotes capturados.

Existem versões do Wireshark para as plataformas Windows, Linux, MacOs, Solaris, FreeBSD, entre outros. Isso é muito bom, porque evita que você tenha que aprender a utilizar vários programas diferentes para realizar a mesma tarefa.

O programa Wireshark é muito usado por profissionais de redes, como:

- **Administradores**, com o objetivo de solucionar problemas na rede.
- **Engenheiros de segurança**, com o objetivo de verificar possíveis problemas na segurança.
- **Desenvolvedores**, que objetiva encontrar possíveis bugs na implementação do protocolo.

- **Estudantes**, com o objetivo de melhorar o entendimento acerca dos protocolos utilizados em redes de computadores.

Vale salientar que o Wireshark também é muito usado pelos hackers para fins ilícitos, ou seja, para fins que contrariam a lei, pois com ele é possível encontrar falhas nas implementações dos protocolos, capturar senhas e obter outras informações.

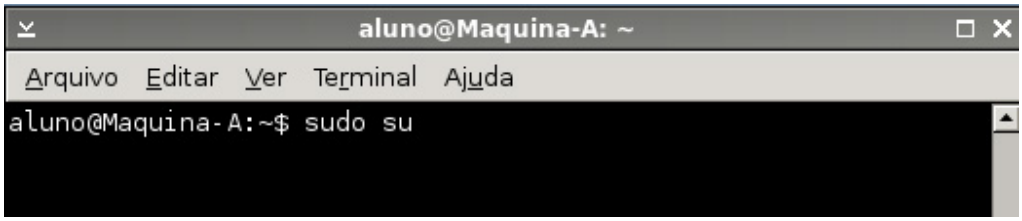
Vídeo 03 - Wireshark

Instalação do Wireshark

Nesta seção, mostraremos como realizar a instalação do Wireshark no Linux. Há versão do Wireshark também para o Windows, mas preferimos o Linux porque é um ambiente mais flexível e propício a depuração de softwares e protocolos.

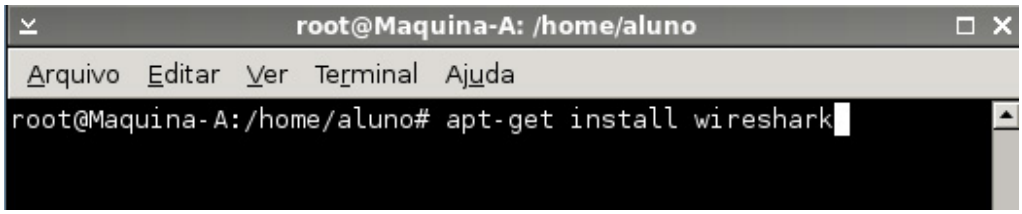
Para iniciar a instalação do pacote do Wireshark no Linux, o usuário deverá ter a permissão máxima, ou seja, ser o superusuário (**root**). O comando Linux que transforma o usuário em superusuário é o **sudo su**, conforme mostrado na **Figura 1** . Lembramos que as operações usadas nesta aula são aceitas em qualquer distribuição Linux baseada na distribuição Debian, como o Ubuntu.

Figura 01 - Acesso como superusuário



Na **Figura 2** é apresentado o comando usado **apt-get install** para instalar o Wireshark.

Figura 02 - Instalação a partir do apt-get



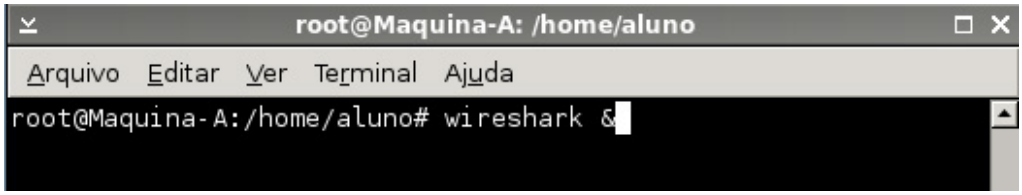
Como dissemos, também é possível instalar o Wireshark a partir do modo gráfico, mas como os detalhes desse processo dependem muito da versão do Linux que você estiver usando, não vamos abordá-lo aqui.

Utilizando o Wireshark

Após a instalação do Wireshark, vamos agora executá-lo e aprender a utilizá-lo. Lembramos que a execução do Wireshark deverá ser como superusuário (root), de modo que você deve digitar o comando *sudo su* antes de executar o Wireshark. Para iniciar o programa usamos o comando *wireshark*.

Porém, desta forma o programa ficará atrelado ao terminal, ou seja, se o terminal for encerrado o programa também o será. Outra forma de executar o programa é acrescentar o caractere & após seu nome, fazendo com que o programa execute em *background* (segundo plano), liberando, assim, o terminal para que você possa utilizá-lo. O comando ficaria assim: *wireshark* &, como mostrado na **Figura 3** .

Figura 03 - Execução do Wireshark



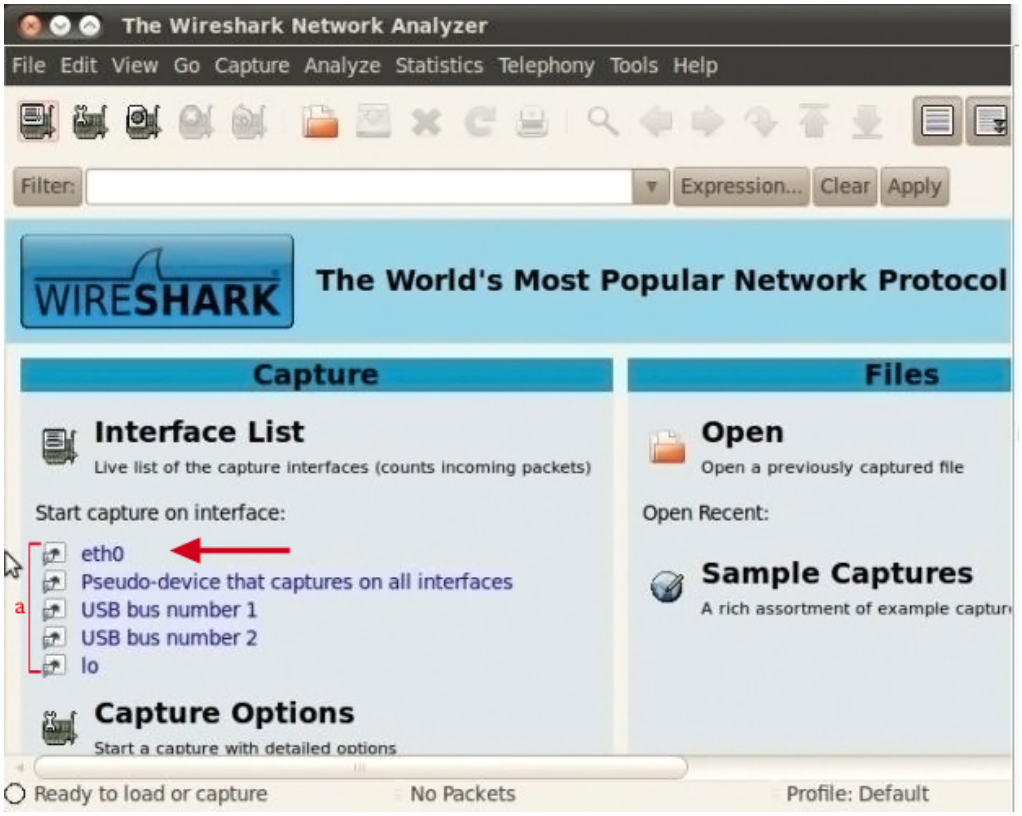
Ao executar o comando mostrado na **Figura 3** , o Wireshark irá iniciar e teremos a tela mostrada na **Figura 4**. A chave apontada pela letra a nessa figura mostra as interfaces presentes no computador que podem ser usadas para capturar os pacotes. Basta clicar sobre o nome da interface para que a captura de pacotes seja iniciada. Saiba que os nomes das interfaces podem mudar de computador para computador, pois elas dependem do *hardware* presente em sua máquina e como o sistema operacional o chama. A interface deve ser a usada para ter acesso à rede.

No Linux, o nome das placas de rede Ethernet são *eth0*, *eth1*, e assim sucessivamente, e as placas de rede sem fio (*wireless*) são chamadas de *wlan0*, *wlan1*, e assim sucessivamente.

Vale lembrar que você pode escolher capturar os pacotes em modo promíscuo (*promiscuous mode*) ou não promíscuo.

No modo promíscuo, a interface aceitará todos os pacotes da rede que chegarem nela, mesmo os que não são destinados a ela. O modo desejado é informado no filtro, que será explicado posteriormente.

Figura 04 - Tela inicial do Wireshark



Atividade 01

---

1. Para que serve colocar o caractere & após o nome do programa Wireshark quando vamos executar esse programa através do terminal?

2. O que significa modo promíscuo?

Resposta

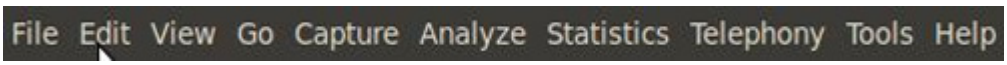
Resposta

1. Para colocar o programa para executar em segundo plano (background) liberando o terminal para que possa continuar a ser utilizado.
2. É um modo onde a placa de rede captura todos os pacotes, inclusive os que não são destinados a ela (ou seja, ao seu endereço MAC).

Menus

Agora que você já sabe instalar, executar e escolher a interface de rede de onde capturar os pacotes, vamos mostrar em detalhes as opções disponíveis no menu principal do Wireshark, que é mostrado na **Figura 5**. Clicar em qualquer uma das palavras desse menu irá levar a um outro menu mais específico, que são descritos a seguir.

Figura 05 - Menu do Wireshark



File (*Arquivo*): É possível salvar as informações capturadas em um arquivo para que possam ser analisadas posteriormente. Esse menu contém itens que possibilitam salvar as informações em um arquivo, abrir um arquivo salvo anteriormente, imprimir, exportar (salvar em um formato padrão, como txt). Além disso, existe a opção *sair*, que fecha o programa.

Edit (*Editar*): Esse menu contém itens que possibilitam encontrar um pacote específico dentre todos os pacotes capturados. Para isso aplicamos um filtro, que consiste em especificar alguma informação sobre o pacote desejado. Após aplicar o *filtro* apenas os pacotes que “casam” com ele são exibidos. Outra opção muito importante desse menu é Preferências, que permite configurar várias coisas sobre a aparência e o modo de funcionamento do programa.

View (*Visualizar*): Esse menu contém itens que controlam a exibição dos dados capturados, como por exemplo: a coloração de pacotes, a ampliação da fonte, a exibição do pacote em uma janela separada, entre outros.

Go (*Ir*): Esse menu contém itens que permitem ir para um pacote específico.

Capture (*Captura*): Esse menu contém itens que lhe permitem iniciar e parar a captura de pacotes, além de definir filtros a serem aplicados na captura. Enquanto os filtros utilizados no menu Edit (*Submenu Find Packet* – Localizar pacotes) controlam os pacotes que são exibidos na tela, os filtros especificados no submenu Options desse menu controlam os pacotes que serão capturados. O submenu "*Capture Filters*" (Filtros de captura) desse menu permite que você crie novos filtros. Na sessão "Filtros" desta aula, daremos mais detalhes sobre eles.

Analyze (*Analisar*): Esse menu contém itens que permitem manipular filtros de tela, ativar ou desativar a dissecção de protocolos. Você não precisa se preocupar muito com esse menu, porque, normalmente, ele não é muito utilizado.

Statistic (*Estatística*): Esse menu contém itens que exibem janelas com estatísticas diversas, incluindo um resumo dos pacotes que foram capturados, hierarquia de protocolos e muito mais.

Telephony (*Telefonia*): Esse menu contém itens que exibem várias estatísticas relacionadas com janelas de telefonia, incluindo uma análise de mídia, diagramas de fluxo e muito mais. Você não precisa se preocupar muito com esse menu, porque, normalmente, ele não é muito utilizado.

Tools (*Ferramentas*): Esse menu contém várias ferramentas disponíveis no Wireshark, como a criação de Regras de ACL do Firewall.

Help (*Ajuda*): Esse menu contém itens para ajudar o usuário, por exemplo, no acesso a uma lista dos protocolos suportados, páginas de manual, o acesso online às páginas relacionadas.

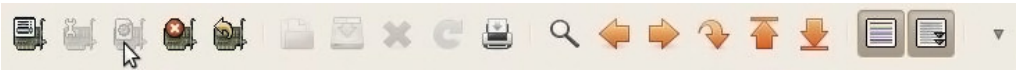
Talvez depois de olhar tantas opções de menu você esteja achando que utilizar o Wireshark é muito complicado. Não se preocupe, pois com o tempo você verá que na prática a maioria dessas opções não é usada frequentemente. Muitas vezes você apenas clica no nome da interface para iniciar a captura e clica nos pacotes que

desejar ver mais detalhes sobre ele. Realmente a utilização do Wireshark é bem simples! Olhe a descrição dos botões apresentada a seguir que você já verá um número bem menor de opções, mas que fornecem a maioria das operações que você vai precisar.

Botões de Atalho

Algumas das operações discutidas acima, referentes aos menus da **Figura 5**, podem ser executadas a partir de um atalho, como mostrado na **Figura 6**.

Figura 06 - Aba de botões de atalho



Captura de interfaces.

Mostra a opção de captura.

Inicia a captura.

Para a captura.

Reinicia a captura.

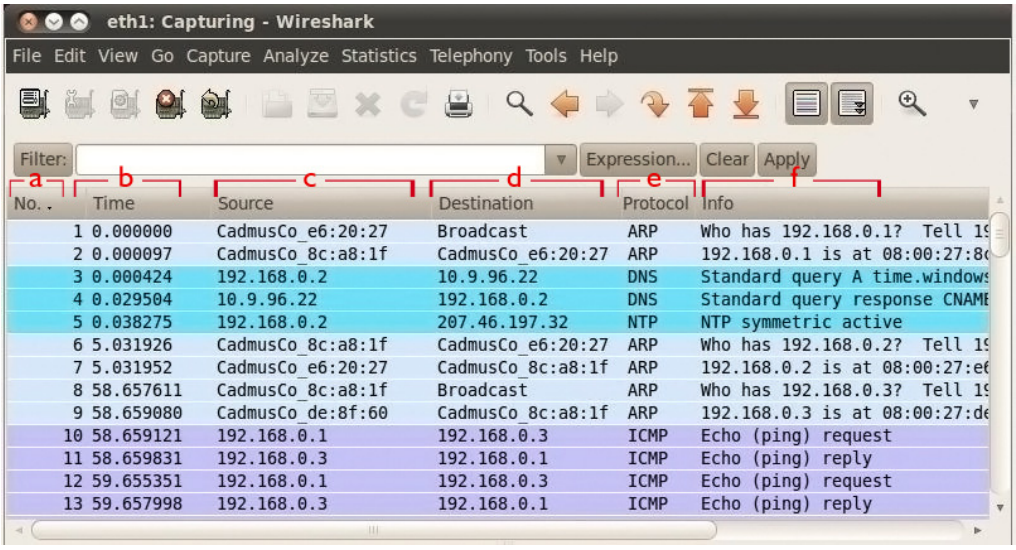
Listagem dos Pacotes Capturados

A parte mais importante que você precisa saber é analisar os pacotes.

Portanto, vamos agora começar a ver como o Wireshark mostra os pacotes para você. Nesta seção, você verá como o Wireshark lhe apresenta a lista dos pacotes capturados, e na próxima seção aprenderá como ver os detalhes de cada pacote.

A listagem dos pacotes capturados é mostrada na **Figura 7**. Essa tela irá aparecer após você iniciar a captura dos pacotes, que pode ser feita, como já dissemos, clicando no nome da interface, selecionando uma opção do menu, ou clicando no botão.

Figura 07 - Lista dos pacotes capturados



A seguir, discutiremos o que significam os campos identificados com as letras de “a” até “f”, na **Figura 7** .

a. No. : Número do pacote na lista dos pacotes capturados. Como você pode perceber, esse número é sequencial, e é incrementado cada vez que um pacote é capturado.

b. Time: Tempo de *timestamp* do pacote. Informa o instante decorrido desde o início da captura até o momento em que o referido pacote foi capturado. É útil, por exemplo, para você analisar o tempo decorrido entre dois pacotes quaisquer, como

o tempo decorrido entre o envio de um pacote que solicitou uma página web a um servidor, e o pacote de resposta, contendo a página, enviado pelo servidor.

c. Source: Endereço IP de origem, ou seja, qual máquina transmitiu o pacote.

d. Destination: Endereço de destino, ou seja, para qual máquina o pacote deve ser entregue.

e. Protocol: Protocolo usado na transferência do pacote.

f. Info: Informações adicionais sobre o conteúdo do pacote. As informações que são mostradas dependem do tipo de protocolo sendo utilizado (informado no item anterior).

> É muito comum salvar os pacotes capturados em um arquivo e depois analisá-los. Normalmente se captura os pacotes com o tcpdump (que estudaremos em breve) e depois se utiliza o Wireshark para analisar os pacotes.

Veja aqui a explicação, em vídeo, sobre a interface do Wireshark.

Vídeo 04 - Wireshark - pt.2

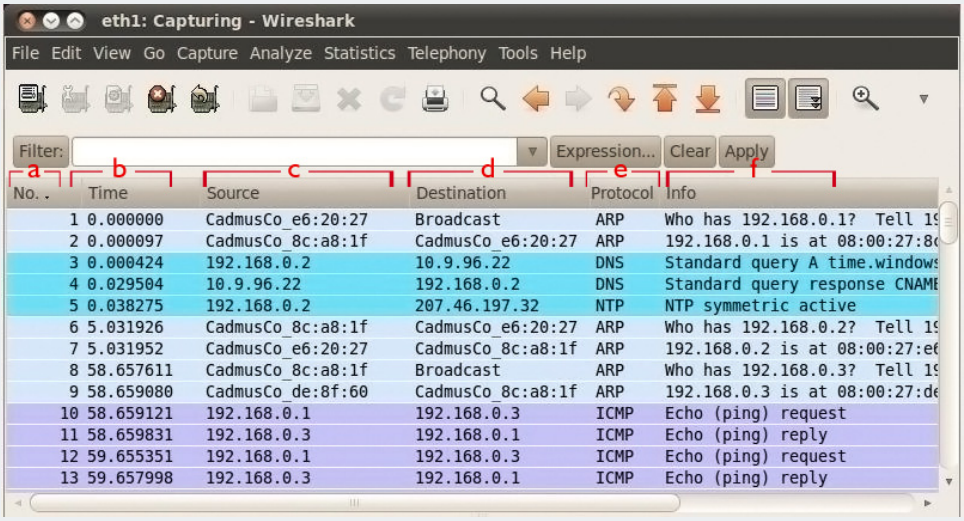
Atividade 02

1. O que significa o campo “Source” na Figura 7?

2. O que significa o campo “Destination” na Figura 7?

Resposta

Figura 07 - Lista dos pacotes capturados



Resposta

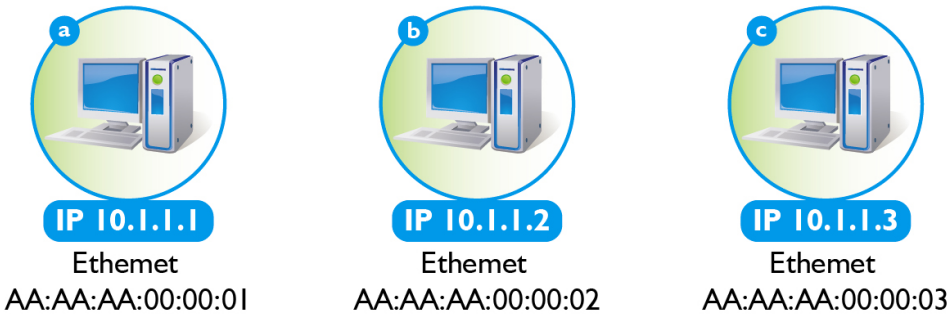
1. O endereço da máquina que transmitiu o “pacote”. Se for um pacote IP (ex: ICMP, UDP, TCP) será mostrado o endereço IP, mas se for um pacote ARP será mostrado o endereço MAC.

2. O endereço da máquina para quem o “pacote” é destinado. Se for um pacote IP (ex: ICMP, UDP, TCP) será mostrado o endereço IP, mas se for um pacote ARP será mostrado o endereço MAC.

Analisando um Pacote Detalhadamente

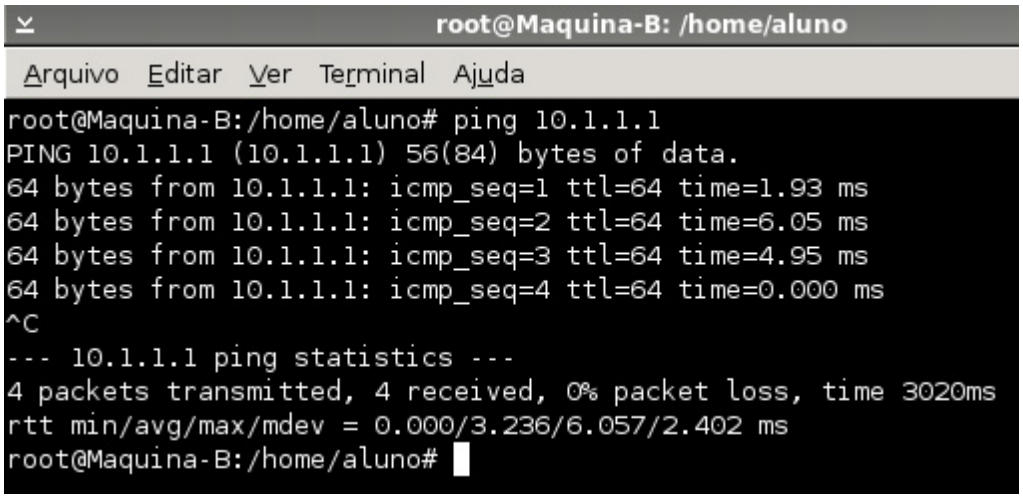
A listagem mostrada na **Figura 7** é importante para lhe dar uma visão mais geral da comunicação, pois ela permite identificar quais pacotes foram transmitidos, e em que ordem. Mas, muitas vezes, é necessário analisar detalhes de alguns pacotes. Para mostrar como fazer isso, vamos realizar um *ping* entre duas máquinas e capturar os pacotes. Assuma uma rede composta pelas três máquinas mostradas na **Figura 8**.

Figura 08 - Três máquinas em uma rede



Imagine que a máquina B realizou um *ping* para a máquina A, conforme mostrado na **Figura 9**. O*ping* usa as mensagens "Echo Request" e a "Echo Reply" do protocolo ICMP. Desse modo, a máquina que realiza *ping* envia para a outra a mensagem "*Echo Request*", e a máquina que recebe esta mensagem envia de volta a mensagem "*Echo Reply*".

Figura 09 - Maquina B realizou um *ping* para a máquina A

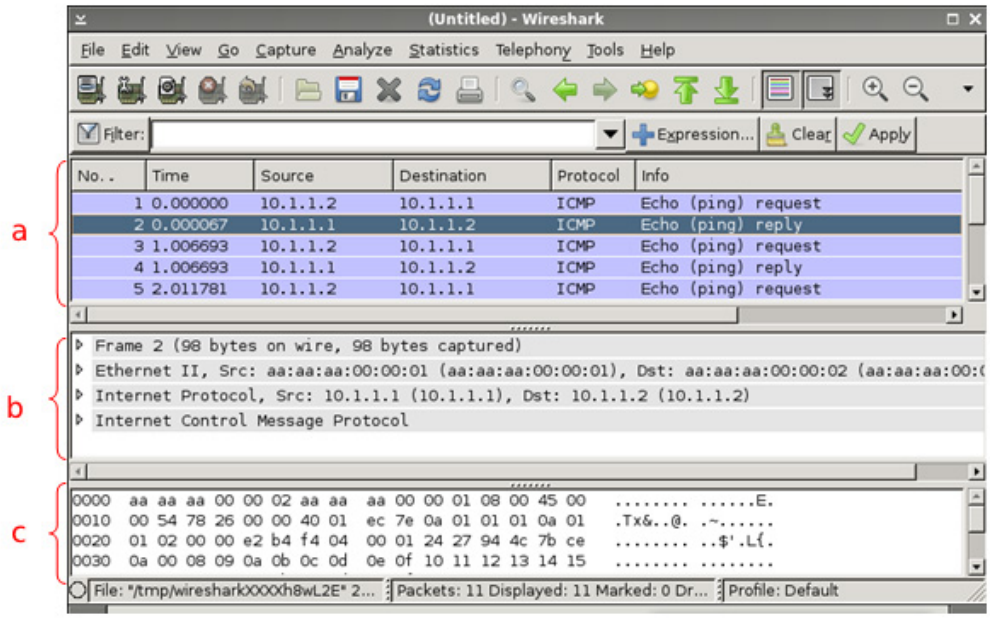


Como você já sabe, o *ping* é útil para verificarmos se conseguimos nos comunicar com uma determinada máquina e para termos uma ideia do tempo em que a comunicação demora.

Por isso, em uma saída típica do comando *ping* , como a que pode ser vista na **Figura 9**, ele informa se recebeu o "*Echo Reply*" para cada pacote transmitido (icmp_seq=), e quanto tempo depois do envio do "Echo Request" essa resposta foi recebida (time=). Desse modo, vemos que para os quatro primeiros pacotes transmitidos, as respostas foram recebidas.

Após essa pequena “pausa” para falarmos do *ping* , vamos agora voltar ao Wireshark e à questão de como podemos ver os detalhes de cada pacote. Lembrando que cada pacote ICMP é enviado dentro de um pacote IP, veja na **Figura 10** a tela completa do Wireshark após a captura dos pacotes de *ping* realizado de B para A.

Figura 10 - Pacotes capturados de um *ping* de B para A com o Wireshark



Na Figura 10 você pode ver que a tela onde os pacotes capturados são exibidos é formada por três partes, **a, b** e **c**. Em a, vemos a mesma parte da tela do Wireshark que será detalhada na **Figura 11**, sendo que agora reduzimos seu tamanho para vermos também as informações mostradas em **b** e **c**.

Em**a**, podemos ver que o primeiro pacote capturado foi um pacote ICMP (campo *Protocol*) do tipo *Echo Request* (campo *Info*) enviado pela máquina com IP 10.1.1.2 (campo *source*) para a máquina 10.1.1.1 (campo *destination*). A segunda linha em **a** mostra que 10.1.1.1 enviou a resposta quando haviam passado 0.000067 segundos (campo *Time*) desde que a captura de pacotes tinha sido iniciada.

A parte identificada por **b** mostra informações individuais do pacote, separando essas informações de acordo com os cabeçalhos de cada protocolo existente no pacote. Cada linha mostra informações sobre um protocolo. Desse modo, a segunda

linha de **b** mostra informações sobre o quadro Ethernet, a terceira linha sobre o protocolo IP, e a quarta linha sobre o protocolo ICMP. A primeira linha mostra informações relacionadas principalmente ao tempo em que o pacote foi capturado.

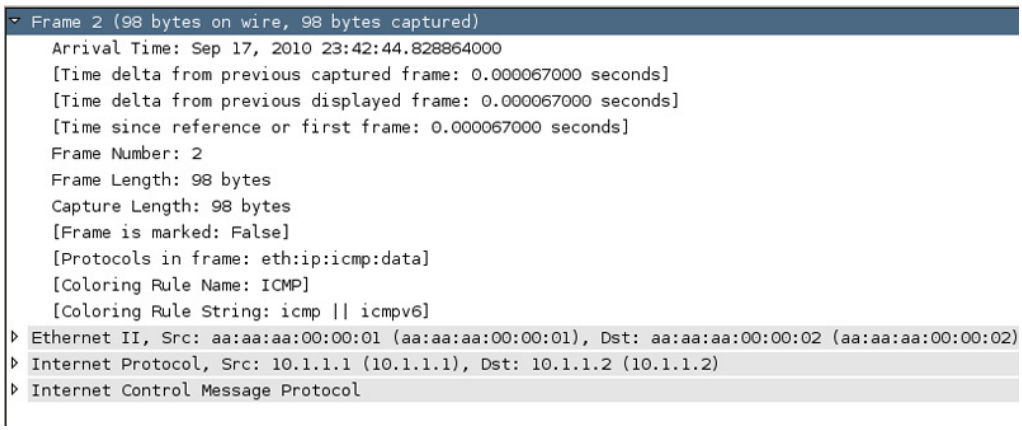
Em **c**, o conteúdo do pacote é mostrado exatamente como ele é recebido, ou seja, como uma sequência de bytes. Essas informações são utilizadas para realizar análises mais profundas nos protocolos. Normalmente, você vai analisar apenas as informações mostradas em **a** e **b**.

As informações mostradas em **b** e **c**são referentes ao pacote que estiver selecionado em **a**. Para selecionar um pacote em **a**, basta clicar nele uma vez. Na **Figura 10**, o pacote selecionado em **a** é o de numero 2. Se você clicar duas vezes sobre um pacote em **a** as informações contidas em **b** e **c** também serão mostradas, só que em uma outra janela.

Podemos expandir as informações mostradas em **b** para cada protocolo, ou seja, ver as informações de cada protocolo em detalhes. Nas próximas quatro figuras, expandimos um item de **b** por vez. Todas as informações mostradas são referentes ao pacote de número 2, que é um pacote ICMP de *Echo Reply* enviado de 10.1.1.1 para 10.1.1.2.

Na **Figura 11** expandimos a primeira linha em b, de modo que você pode ver que os detalhes mostrados são referentes, principalmente, à hora em que o pacote foi capturado, seu tamanho, e quais protocolos estão contidos no pacote.

Figura 11 - Informações básicas sobre o pacote capturado

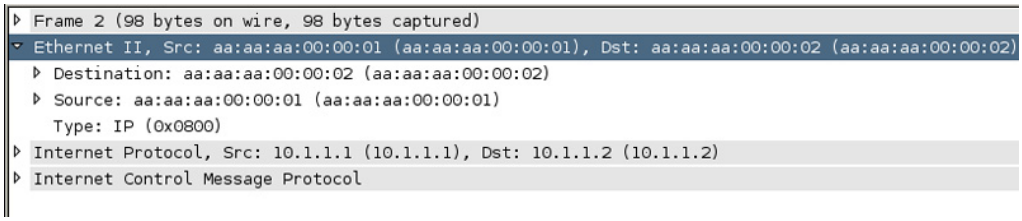


Na **Figura 12** , expandimos a segunda linha em **b**, que mostra informações sobre a camada de enlace utilizada para transmitir o quadro. Você pode ver que se tratava de um quadro Ethernet.

Desse modo, são mostrados os valores contidos nos campos endereço de destino (*Destination*) e endereço de origem (*Source*) do quadro, além do valor do campo de tipo (*Type*). Pelos valores mostrados você vê que o quadro foi enviado da máquina cujo endereço Ethernet da placa de rede é AA:AA:AA:00:00:01 para a máquina com endereço Ethernet AA:AA:AA:00:00:02.

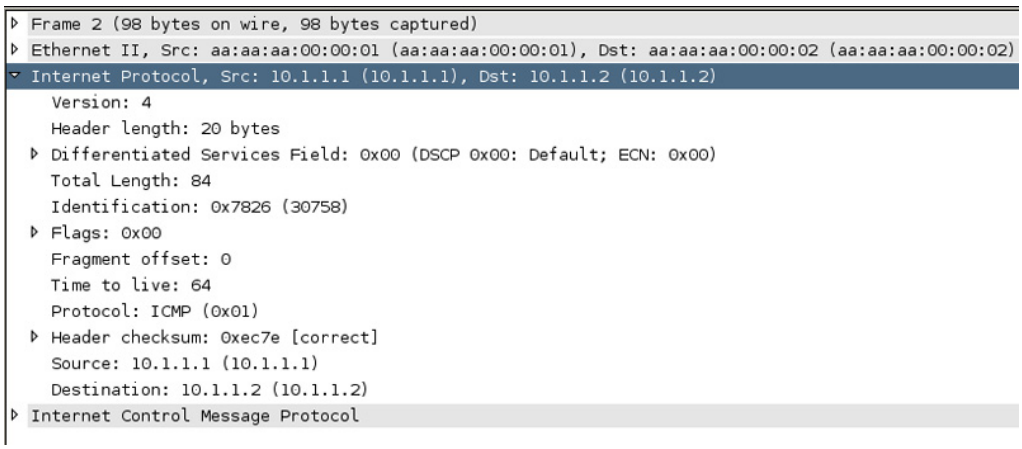
Se você olhar a **Figura 8** , verá que esses são os endereços das máquinas A e B, respectivamente. Ou seja, da máquina que recebeu o *ping* para a que o enviou. O valor 0x0800 contido no campo de tipo (*Type*) mostra que o conteúdo do quadro Ethernet é um pacote IP. 0x0800 é o valor em hexadecimal do código do protocolo IP.

Figura 12 - Informações da camada de enlace



Ao expandimos a terceira linha em**b**, vemos as informações referentes ao cabeçalho IP, conforme mostrado na **Figura 13** . Não vamos descrever todos os campos, mas você pode observar que o pacote foi enviado a partir do endereço IP (*Source*) 10.1.1.1 para a máquina com o endereço 10.1.1.2.

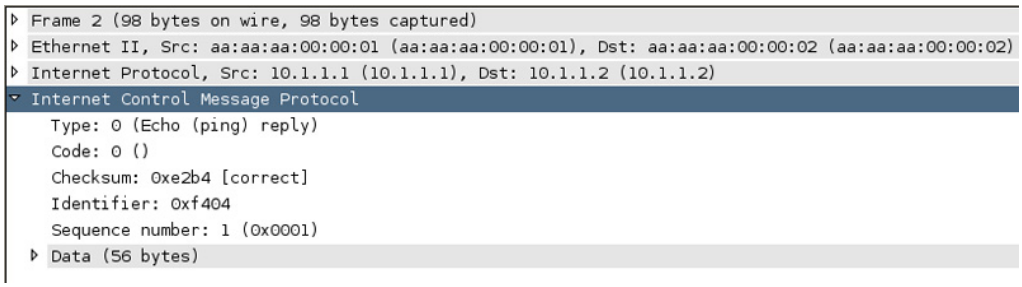
Figura 13 - Informações da camada de rede.



Além disso, você pode observar que o campo *Protocol* contém o código do protocolo ICMP, que é o número 0x01 (em hexadecimal). Lembre-se que esse campo indica qual é o protocolo contido na parte de dados dopacote IP!

Ao expandimos a quarta linha em **b**, vemos as informações referentes ao protocolo ICMP (*Internet Control Message Protocol*), conforme mostrado na Figura 14. O primeiro campo mostra o tipo de mensagem ICMP, nesse caso, é uma mensagem de echo reply (tipo 0). Observe que há também um campo de Checksum que, relembrando, serve para checar a integridade do pacote.

Figura 14 - Informações do protocolo ICMP



No vídeo abaixo, a continuação das explicações sobre a captura de pacotes no Wireshark.

Vídeo 05 - Wireshark - pt.3

Utilizando Filtros

Como explicamos anteriormente, o Wireshark permite que você utilize filtros para selecionar quais pacotes serão capturados, ou para selecionar entre os pacotes já capturados quais serão exibidos na tela. Para o primeiro caso, utilize o menu "*Edit->FindPacket*", e depois clique no botão "*Filter*". Para o segundo caso, clique no menu "*Capture->Options*" e depois clique no botão "*Filter*".

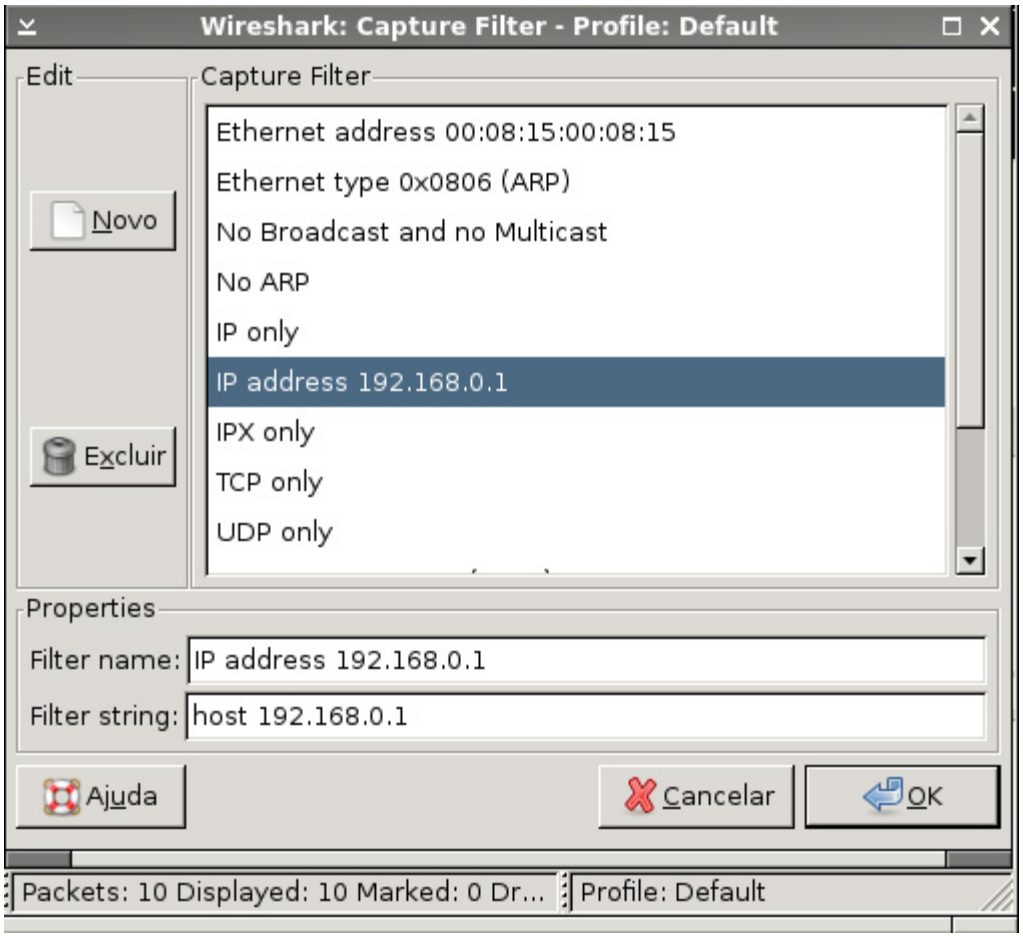
O Wireshark já vem com uma série de filtros e você ainda pode criar seus próprios filtros através do menu "*Capture-> Capture Filters*".

Um filtro nada mais é do que uma regra contendo nomes de campos dos protocolos contidos nos pacotes e os valores que cada campo deve conter. Se um pacote contém o valor especificado no filtro para o referido campo, ele “casa” com o filtro.

Um exemplo de filtro simples pode ser que o endereço IP do pacote seja X.X.X.X (substituindo X.X.X.X pelo endereço desejado). Outro filtro simples pode ser que o protocolo seja TCP e a porta seja 80.

A tela onde se escolhe um filtro para aplicar ou se permite a criação de um novo filtro é a mesma, e está mostrada na **Figura 15**.

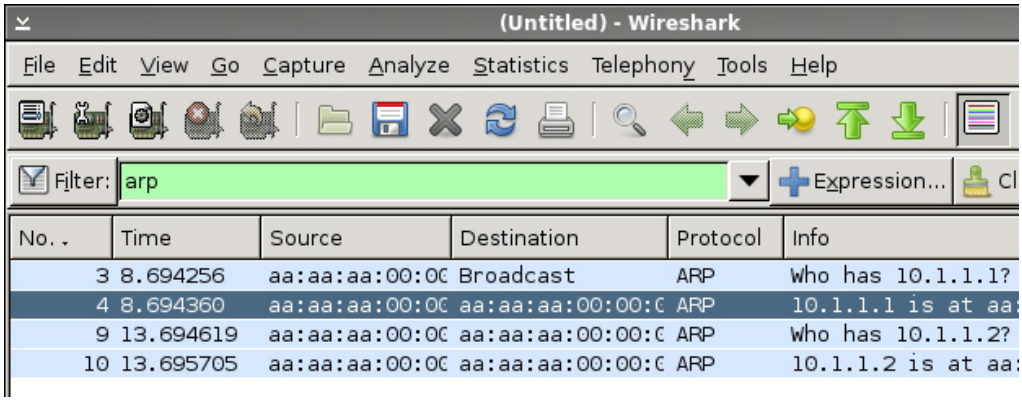
Figura 15 - Filtros no Wireshark



Veja ainda na **Figura 15** que ao clicarmos sobre um filtro na parte inferior da tela aparece o texto (string) que equivale a esse filtro.

Com o tempo, você pode começar a aprender esta “linguagem” e preferir escrever as regras do filtro ao invés de selecioná-las na tela mostrada na **Figura 15**. Conforme mostrado na **Figura 16**, existe um campo na interface do Wireshark que lhe permite escrever os filtros usados para selecionar os pacotes exibidos. No exemplo, o filtro faz com que sejam exibidos apenas os pacotes do protocolo ARP.

Figura 16 - Especificando um filtro através de texto



Nos vídeos abaixo, mais explicações sobre o Wireshark.

Vídeo 06 - Wireshark - pt.4

Vídeo 07 - Wireshark - pt.5

Recursos Disponíveis por Plataforma

Saiba que nem todas as interfaces (Bluetooth, Ethernet, USB, WLAN, Loopback etc.) são suportadasnas versões do Wireshark para todos os sistemas operacionais. Verifique se a interface que pretende utilizar é suportada no seu sistema operacional no link: http://wiki.wireshark.org/CaptureSetup/NetworkMedia. Adiantamos que Ethernet e WLAN são suportadas no Windows, Linux e Mac.

Tcpdump

Uma das ferramentas de captura de pacotes mais tradicionais no Linux é o tcpdump. Apesar de não possuir uma interface gráfica, ele ainda é até hoje muito utilizado.

Lembra-se da linguagem de filtros do Wireshark? Pois é, digamos que no tcpdump essa é a única forma de definir os filtros.

Caso esteja pensando porque você iria deixar de usar uma ferramenta que possui uma ótima interface gráfica para usar uma em modo texto, temos duas respostas.

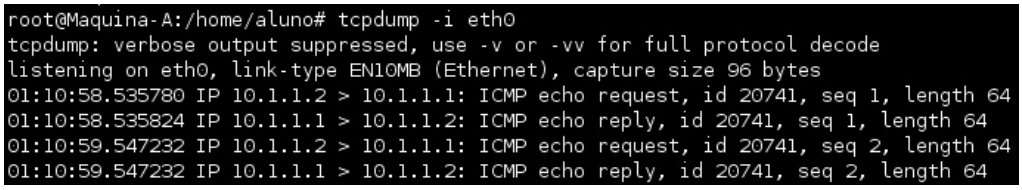
A primeira é agilidade. É bem mais rápido digitar um comando do que abrir um programa gráfico e utilizar seus menus.

A segunda é que, muitas vezes, você precisa capturar pacotes em servidores que não têm uma interface gráfica instalada. Mas não se preocupe. Normalmente, nesses casos os filtros que você vai precisar usar são bem simples.

Você, normalmente, vai estar mais interessado em verificar se determinados pacotes chegaram até a máquina onde você está executando a ferramenta, do que propriamente em analisar detalhadamente os campos do pacote. Nesses casos a interface gráfica realmente não faz tanta diferença.

Vamos capturar os mesmos pacotes do exemplo mostrado na **Figura 9**, onde a máquina B, que possui o endereço IP 10.1.1.2, enviou um *ping* para a máquina A que possui o endereço IP 10.1.1.1. A **Figura 17** mostra o comando e a saída do tcpdump.

Figura 17 - Pacotes capturados com tcpdump



Veja que a sintaxe é bem simples. O “-i eth0” é para indicar de qual placa de rede os pacotes devem ser capturados. Caso não fosse informado, o padrão é capturar da eth0. Portanto, nesse caso, bastava termos digitado “tcpdump”.

É muito comum utilizarmos também a opção “-n” para dizer ao tcpdump para mostrar as informações (endereços IP, números de porta) como números e não como nomes.

Além disso, você pode observar que as informações exibidas são praticamente as mesmas do Wireshark (só que mais resumidas), com a diferença que não aparecem os nomes dos campos. O **Quadro 1** mostra outros exemplos do uso do tcpdump.

Comando	Significado
tcpdump src 10.1.1.1 and icmp	Pacotes ICMP cujo IP de origem e 10.1.1.1.
tcpdump dst 10.1.1.2 and tcp and dst port 80	Pacotes cujo endereço IP de destino seja 10.1.1.2 e o protocolo de transporte seja TCP e a porta de destino seja 80.
tcpdump host 10.1.1.1	Pacotes cujo endereço IP de origem ou o de destino seja 10.1.1.1.
tcpdump –i eth0 –n –w /dados/ping.pcap	Captura pacotes recebidos e transmitidos pela eth0 e salva os pacotes no arquivo /dados/ping.pcap (sem tentar traduzir os endereços IP, nem os nomes das portas, para os nomes equivalentes).

Quando 1 - Exemplos do tcpdump

Veja aqui a explicação, em vídeo, sobre o tcpdump

Vídeo 08 - O que é Tcpdump?

Vídeo 09 - Captura de Pacotes Tcpdump

Vídeo 10 - Tcpdump

Resumo

Nesta aula, você aprendeu um novo tipo de ferramenta, que são as ferramentas de captura de pacotes. Estudou qual a finalidade delas e aprendeu a utilizar dois programas diferentes: o Wireshark e o tcpdump. Aprendeu que essas ferramentas mostram as informações dos cabeçalhos de cada protocolo existente no pacote e possuem uma linguagem de filtro para se especificar quais pacotes devem ser capturados.

Autoavaliação

1. Através de qual menu do Wireshark é possível especificar um filtro para selecionar entre todos os pacotes capturados quais devem ser mostrados na tela?
2. Qual a expressão de texto no Wireshark para um filtro que deve capturar apenas pacotes IP transmitidos ou recebidos do IP 192.168.1.2 (Dica: olhe os filtros que já existem.)?
3. O que significa o campo "*Time*" na lista de pacotes capturados?
4. É possível utilizar o Wireshark para capturar pacotes transmitidos entre quaisquer duas máquinas da sua rede sem executá-lo como root (administrador)?
5. Qual a principal diferença entre o Wireshark e o tcpdump?

Resposta

Resposta

1. Menu “Capture”, submenu “Capture Filters”.
2. ip.src == 192.168.1.2
3. O tempo em que o pacote foi capturado, referente ao início da captura.
4. Não. De fato existe uma maneira de fazer isso, mas é muito avançada para vermos nessa disciplina. Por isso, a resposta dos alunos deve ser mesmo NÃO.
5. A diferença principal é que o Wireshark possui uma interface gráfica que facilita muito a utilização. Não sendo necessário, por exemplo, decorar a sintaxe para definição de filtros.

Referências

KUROSE, James; ROSS, K. **Redes de Computadores** e a Internet:uma abordagem top-down. 5. ed. São Paulo: Addison Wesley, 2010. p. 640.

TCPDUMP & LIBPCAP. Disponível em: http://www.tcpdump.org/#documentation Acesso em: 12 maio 2012.

WIKIPÉDIA.Promiscuousmode. Disponível em: http://en.wikipedia.org/wiki/Promiscuous_mode.Acesso em: 12 maio 2012.

WIRESHARK.Disponível em: http://www.wireshark.org/docs/. Acesso em: 12 maio 2012.