

PRACTICAL PACKET ANALYSIS

USING WIRESHARK TO SOLVE REAL-WORLD
NETWORK PROBLEMS

CHRIS SANDERS



Tradução revisada em 09.08.2010

AGRADECIMENTOS

Meus sinceros agradecimentos a três amigos queridos com os quais tenho a honra de trabalhar: Ednilson Silva, Lênio Edberg semeadores do conhecimento em comunicação de dados e Christian Barnard companheiro e iniciante nessa longa jornada: de desafios, pesquisas e disseminação do conhecimento adquirido. Esta tradução não poderia ter sido feita sem o auxílio dessas pessoas que compartilharam comigo seus conhecimentos e acima de tudo a amizade. Esse trabalho é fruto dessa sinergia e esperamos que venha ajudar àqueles que buscam uma literatura em nossa língua portuguesa, que trate da análise de pacotes do mundo IP; Tão presente no nosso dia a dia. Por se tratar da tradução de uma obra publicada, não vislumbramos em hipótese alguma o cunho comercial, portanto fica aqui por parte do tradutor expressamente o desejo de não se fazer o comércio de qualquer tipo ou forma de publicação.

“Ao Grande Arquiteto do Universo, toda honra e toda glória”
José VILOBALDO Soares
@Badu

CONTEÚDO

CAPÍTULO 1

| | |
|---|----------|
| BÁSICO DE REDES E ANÁLISE DE PACOTES..... | 6 |
| O que é Análise de Pacotes?..... | 7 |
| Evolução de um Packet Sniffer (Farejador de Pacotes)..... | 7 |
| Como o Farejador de Pacotes (Packet Sniffer) Funciona | 7 |
| Como os Computadores se Comunicam | 8 |
| O Modelo OSI de Sete Camadas | 8 |
| Interação entre Protocolos..... | 10 |
| Encapsulamento de Dados | 10 |
| Hardware da Rede | 11 |
| Classificação de Tráfego | 14 |

CAPÍTULO 2

| | |
|-----------------------------------|-----------|
| CHACOALHANDO OS FIOS | 15 |
| Modo Promíscuo | 16 |
| Espelhamento de Porta | 18 |
| Hubbing | 18 |
| Envenenamento do Cache ARP | 19 |
| Usando o Cain & Abel | 20 |

CAPÍTULO 3

| | |
|--|-----------|
| INTRODUÇÃO AO WIRESHARK | 23 |
| Uma Breve História sobre o Wireshark | 23 |
| Os Benefícios do Wireshark..... | 23 |
| Protocolos Suportados | 24 |
| Instalando o Wireshark..... | 24 |
| Fundamentos do Wireshark | 26 |
| A Janela Principal..... | 27 |
| A caixa de Diálogo Preferências | 28 |
| Código de Cores de um Pacote..... | 29 |

CAPÍTULO 4

| | |
|--|-----------|
| TRABALHANDO COM PACOTES CAPTURADOS | 31 |
| Encontrando e Marcando Pacotes | 31 |
| Salvando e Exportando Arquivos Capturados | 32 |
| Formatos de Exibição de Tempo e Referências..... | 35 |
| Filtros de Captura e Exibição | 36 |
| A Sintaxe de Construção de Filtros..... | 37 |

CAPITULO 5

| | |
|---|-----------|
| CARACTERISTICAS AVANÇADAS DO WIRESHARK | 40 |
| Resolução de Nomes | 40 |
| Dissecação de Protocolo..... | 41 |
| Seguindo os fluxos TCP | 43 |
| Janela Estatística Hierárquica de Protocolo | 43 |
| Visualizando os Dispositivos Finais..... | 44 |
| Conversações..... | 45 |
| A Janela IO Graphs..... | 46 |

CAPITULO 6

| | |
|--|-----------|
| PROTOCOLOS MAIS COMUNS..... | 47 |
| Protocolo de Resolução de Endereços – ARP | 48 |
| Protocolo de Configuração Dinâmica – DHCP | 48 |
| TCP/IP e HTTP | 49 |
| TCP/IP | 49 |
| Sistema de Nomes de Domínio – DNS..... | 52 |
| Protocolo de Transferência de Arquivos – FTP..... | 53 |
| Protocolo Telnet | 54 |
| Serviço de Mensagens MSN | 55 |
| Protocolo Internet de Controle de Mensagens – ICMP | 57 |
| Considerações Finais | 57 |

CAPITULO 7

| | |
|------------------------------------|-----------|
| CENÁRIOS BÁSICOS | 58 |
| Perda da Conexão TCP..... | 58 |
| Sem Conectividade | 63 |
| Fantasma no Internet Explorer..... | 64 |
| Problemas com o FTP | 66 |
| Falha no Servidor Remoto | 67 |
| Um Programa Mal Intencionado | 69 |

CAPITULO 8

| | |
|---|-----------|
| LUTANDO CONTRA A LENTIDÃO DA REDE..... | 74 |
| Anatomia do Download Lento..... | 75 |
| Uma Rota Lenta | 78 |
| Visão Dupla | 80 |
| Será que o Servidor Emperrou?..... | 82 |
| Uma falha de Análise | 84 |
| O Servidor POP de Email..... | 85 |
| Acessando o Gnutella | 87 |

CAPITULO 9

| | |
|--|-----------|
| ANÁLISE BASEADA EM SEGURANÇA | 91 |
| Sistema Operacional Fingerprinting | 91 |
| Um Simples Escaneamento (varredura) de Porta | 92 |
| Uma Impressora Abarrotada de Impressão | 93 |
| Um FTP Interrompido | 94 |
| O Vírus (Worm) Blaster..... | 96 |
| Informações Confidenciais | 97 |
| O Ponto de Vista de um Hacker..... | 98 |

CAPITULO 10

| | |
|---|------------|
| FAREJANDO PELO AR | 101 |
| Farejando um Canal de Cada Vez | 101 |
| Interferência do Sinal em uma Rede Sem Fio | 102 |
| Modos de Funcionamento de uma Placa de Rede Sem Fio | 102 |
| Farejando via Rede Sem Fios no Windows..... | 103 |
| Configurando o AirPcap..... | 103 |
| Capturando Tráfego com o AirPcap | 104 |
| Pacotes Extras 802.11..... | 105 |
| Colunas Wireless Específicas | 107 |
| Filtros Wireless Específicos..... | 108 |
| Uma Tentativa Ruim de Conexão | 110 |
| Considerações Finais | 112 |

1

BÁSICO DE REDES E ANÁLISE DE PACOTES

Um milhão de coisas diferentes pode dar errado com uma rede de computadores em um dado dia a partir de uma simples infecção por um spyware ou um erro de uma complexa configuração de um roteador, e será impossível resolver todos os problemas imediatamente. O melhor que podemos esperar fazer é estar plenamente preparado com o conhecimento e as ferramentas para resolver a esses tipos de problemas. Todos os problemas em uma rede decorrem a nível dos pacotes, até mesmo as futurísticas aplicações podem revelar implementações ruins e aparentemente protocolos confiáveis podem ser usados para fins maliciosos. Para melhor compreender e resolver problemas em uma rede, nós vamos ao nível de pacote onde nada está escondido de nós, onde nada é obscurecido pelas estruturas menos enganosas, gráficos atraentes, ou funcionários não confiáveis. Aqui não há segredos, e o que mais pudermos fazer no nível da análise de pacotes, poderemos controlar melhor a nossa rede e resolver os problemas nela encontrados. Este é o mundo da análise de pacotes.

Este livro mergulha no mundo da análise de pacotes de cabeça. Você vai aprender o que é análise de pacotes, antes de mergulhar em rede de comunicação, para que possa ganhar alguns conhecimentos básicos que você precisa para analisar diferentes cenários. Você aprenderá como usar os recursos da ferramenta de análise Wireshark para abordar a comunicação de rede lenta, identificar gargalos na aplicação e até mesmo acompanhar hackers através de alguns cenários do mundo real. Antes de você terminar de ler este livro, você deve ser capaz de implementar técnicas avançadas de análise de pacotes que irão ajudá-lo a resolver os problemas mais difíceis em sua própria rede.

O que é Análise de Pacotes?

Análise de pacotes, muitas vezes referida como **packet sniffing** (**farejamento de pacotes**) ou análise de protocolo, descreve o processo de capturar e interpretar dados em tempo real à medida que flui através de uma rede a fim de entender melhor o que está acontecendo na rede. Análise de pacotes é normalmente realizada por um **packet sniffer** (farejador de pacotes), ferramenta utilizada para capturar dados brutos atravessando os fios de uma rede. A análise de pacotes podem nos ajudar a entender características da rede, saber quem está em uma rede, ou determinar qual é a utilização da largura de banda disponível, identificar as épocas de pico de uso da rede, identificar possíveis ataques ou atividades maliciosas, e encontrar aplicações inseguras.

Existem vários tipos de programas farejadores de pacotes (Packets sniffing), incluindo tanto o software livre como o comercial. Cada programa foi concebido com objetivos diferentes em mente. Alguns dos mais populares programas de análise de pacotes são tcpdump (Um programa de linha de comando), OmniPeek e Wireshark (ambos baseados em GUI sniffers).

Evolução de um Packet Sniffer (Farejador de Pacotes)

Existem vários tipos de sniffers. Ao selecionar o que você vai usar, você deve considerar as seguintes variáveis:

- Protocolos suportados
- Interface amigável
- Custo
- Suporte ao Programa
- Suporte aos Sistemas Operacionais

Protocolos Suportados

Todos os sniffers podem interpretar vários protocolos. A maioria dos sniffers pode interpretar os protocolos mais comuns, tais como DHCP, IP e ARP, mas nem todos podem interpretar alguns dos protocolos não tanto tradicionais. Ao escolher um sniffer, certifique-se que ele suporta os protocolos que você vai usar.

Interface Amigável

Considere o layout do programa, a facilidade de instalação e, em geral o fluxo padrão de operação. O programa que você escolher deve se ajustar ao seu nível de perícia. Se você tem muito pouca experiência em análise de pacotes, você pode querer evitar o mais avançado programa de linha de comandos como o tcpdump. Pelo contrário, se você tem uma experiência muita rica, aí sim você poderá fazer dele a sua melhor escolha.

Custo

A grande coisa a respeito dos programas analisadores de pacotes é que existem muitos softwares livres que qualquer produto rival comercializado. Você nunca poderá ter que pagar por um programa analisador de pacotes.

Suporte ao Programa

Mesmo depois de ter dominado o básico de um programa analisador de pacotes, você ainda vai precisar de apoio ocasional para resolver novos problemas que possam surgir. Ao avaliar os suportes disponíveis, procurar coisas como documentação de desenvolvimento, fóruns públicos, e listas de discussão. Embora possa haver uma falta de suporte ao desenvolvedor para programas de análise de pacotes gratuitos como o Wireshark, as comunidades que utilizam estas aplicações, muitas vezes compensa essa falta, através de fóruns de debates, wikis, blogs desenvolvidos para ajudá-lo a obter mais do seu programa.

Suporte ao Sistema Operacional

Infelizmente, nem todos os programas analisadores de pacotes tem suporte a cada sistema operacional. Tenha certeza de que o que você escolher para aprender funcione em todos os sistemas operacionais que você precisa de suporte.

Como o Farejador de Pacotes (Packet Sniffer) Funciona

O processo de farejamento (sniffer) de pacotes pode ser dividido em três etapas: coleta, conversão e análise.

Coleta

Na primeira etapa, o farejador (sniffer) de pacotes coloca a interface de rede selecionada, em modo promiscuo. Neste modo a placa de rede pode escutar todo o tráfego em seu segmento de rede. O farejador (sniffer) usa este

modo juntamente com o acesso de baixo nível da interface para capturar os dados em formato binário que passam pelos fios.

Conversão

Nesta etapa, os dados binários capturados são convertidos em uma forma legível. Este é o ponto onde os mais avançados analisadores de pacotes param. Neste ponto, a rede de dados está em uma forma que só pode ser interpretada em um nível muito básico, deixando a maioria da análise para o usuário final.

Análise

A terceira e última etapa consiste na análise real dos dados capturados e convertidos. Nesta etapa, o analisador de pacotes pega os dados capturados, verifica o seu protocolo com base nas informações extraídas, e começa a sua análise a partir das características específicas do protocolo. Uma análise mais aprofundada é realizada comparando vários pacotes, bem como vários outros elementos da rede.

Como os Computadores se Comunicam

A fim de compreender totalmente a análise de pacotes, você precisa entender exatamente como os computadores se comunicam uns com os outros. Nesta seção examinaremos os conceitos básicos dos protocolos de rede, o modelo OSI, os quadros de dados de rede, e o hardware que suporta tudo.

Protocolos de rede

As redes modernas são compostas de uma variedade de diferentes sistemas em execução e muitas plataformas diferentes. Para auxiliar nesta comunicação, usamos um conjunto comum de regras chamado protocolos de rede, protocolos esses que regem toda a comunicação. Protocolos de rede comuns incluem TCP, IP, ARP e DHCP. Uma pilha de protocolo é um agrupamento lógico de protocolos que trabalham juntos.

Um protocolo de rede pode ser extremamente simples ou extremamente complexo, dependendo de sua função. Embora os vários protocolos de rede são muitas vezes radicalmente diferentes, a maioria tem que abordam as seguintes questões:

Controle de fluxo - É a geração de mensagens pelo sistema de recepção para instruir o sistema de transmissão para acelerar ou retardar a transmissão de dados

Confirmação de pacotes - É a transmissão de uma mensagem de retorno do sistema de recepção para o sistema de transmissão confirmando a recepção dos dados

Erro de detecção - É o uso de códigos pelo sistema de transmissão para verificar se os dados enviados não foram danificados durante a transmissão

A correção de erros - É a retransmissão de dados que foram perdidos ou danificados durante a transmissão inicial

Segmentação - É a divisão de longos fluxos de dados em fluxos menores para uma transferência mais eficiente

A criptografia de dados - É uma função que usa chaves de criptografia para proteger os dados transmitidos através de uma rede

Compressão de dados - É um método para reduzir o tamanho dos dados transmitidos através de uma rede, eliminando informações redundantes

O Modelo OSI de Sete Camadas

Os protocolos são separados com base em suas funções através de um padrão de referência, modelo de referência chamado de **Open Systems Interconnection** (OSI). Este modelo foi originalmente publicado em 1983 pela **Organização Internacional for Standardization** (ISO) como um documento chamado ISO 7498.

O modelo OSI divide o processo de comunicação de rede em sete camadas distintas:

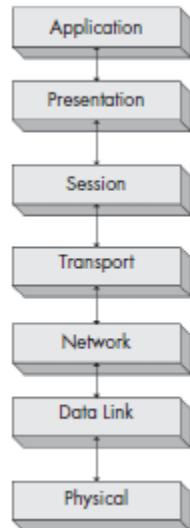
| | |
|-----------------|------------|
| Aplicação | - Camada 7 |
| Apresentação | - Camada 6 |
| Seção | - Camada 5 |
| Transporte | - Camada 4 |
| Rede | - Camada 3 |
| Enlace de Dados | - Camada 2 |
| Física | - Camada 1 |

A divisão em sete camadas do modelo hierárquico OSI (figura ao lado) torna mais fácil a compreensão da comunicação em uma rede. A camada de aplicação, na parte superior representa os programas utilizados para acessar os recursos existentes em uma rede. A camada inferior é a camada física, através da qual a rede envia e recebe os dados. Os protocolos em cada camada trabalham em conjunto com as camadas superiores e inferiores.

Nota:

O modelo OSI não é mais do que um padrão de recomendação, os desenvolvedores de protocolos não são obrigados a segui-lo exatamente. Por uma questão de fato, o modelo OSI não é o modelo de rede única que existe, por exemplo algumas pessoas preferem o modelo do Departamento de Defesa (DoD). Vamos trabalhar em torno dos conceitos do modelo OSI, neste livro.

Vamos olhar amplamente, as funções de cada uma das camadas do modelo OSI, bem como alguns exemplos de protocolos utilizados em cada uma.



A Camada de Aplicação

A camada de aplicação, a camada superior do modelo OSI, fornece um meio para que os usuários realmente acessem os recursos de rede. Esta é a única camada tipicamente vista pelos usuários finais, uma vez que fornece a interface que é a base para todas as suas atividades de rede.

A Camada de Apresentação

A camada de apresentação transforma os dados que recebe em um formato que possa ser lido pela camada de aplicação. Os dados de codificação e decodificação feitos aqui dependem do protocolo da camada de aplicação que está enviando ou recebendo os dados. Esta camada também controla várias formas de criptografia/decriptografia utilizadas para prover a segurança dos dados enviados ou recebidos.

A Camada de Seção

A camada de sessão controla o diálogo, ou sessão entre dois computadores, estabelece, gerencia e termina, a comunicação entre todos os dispositivos. A camada de sessão também é responsável por determinar se uma conexão é duplex ou half-duplex e graciosamente fechar uma conexão entre os dispositivos, ao invés de deixá-la cair abruptamente.

A Camada de Transporte

O objetivo principal da camada de transporte é fornecer o serviço de transporte confiável dos dados para as camadas inferiores. Através de recursos, incluindo o controle de fluxo, a segmentação e controle de erro, a camada de transporte garante o transporte de dados de um ponto a outro sem erros. Garantir o transporte de dados confiável pode ser extremamente complicado, o modelo OSI dedica toda uma camada a ele. A camada de transporte fornece os seus serviços tanto a protocolos orientados à conexão ou não. **Firewalls** e **Servidores Proxy** operam nesta camada.

A Camada de Rede

A camada de rede é responsável pelo roteamento de dados entre redes físicas, e é uma das camadas OSI mais complexas. É responsável pela lógica de endereçamento de hosts da rede (por exemplo, através de um endereço IP), e também lida com a segmentação de pacotes, a identificação do protocolo e, em alguns casos, detecção de erros. Os **Roteadores** operam nesta camada.

A Camada de Enlace de Dados

A camada de enlace de dados permite transporte de dados através do meio físico da rede. Seu objetivo principal é proporcionar um esquema de endereçamento que pode ser usado para identificar os dispositivos físicos e fornecer recursos de verificação de erros para garantir a integridade dos dados. **Bridges** e **Switches** são dispositivos físicos que operam nesta camada.

A Camada Física

A camada física, na parte inferior do modelo OSI é o meio físico através do qual os dados são transferidos na rede. Esta camada define a natureza física e elétrica de todo o hardware utilizado, incluindo as tensões, os nós da

rede, placas, repetidores, e especificações de cabeamento. A camada física estabelece e termina conexões, fornece um meio de compartilhamento dos recursos de comunicação, e converte sinais digitais em analógicos e vice-versa.

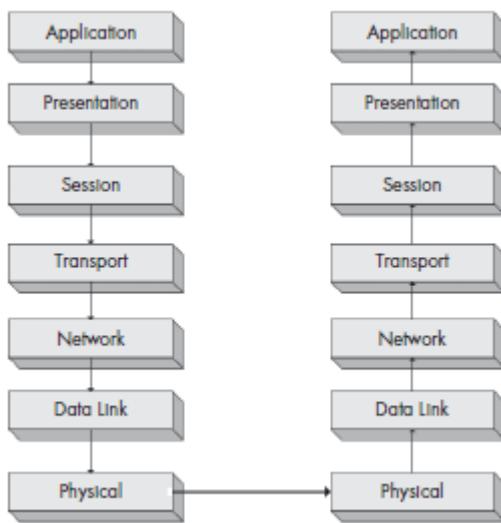
A tabela abaixo lista alguns dos protocolos mais comuns utilizados em cada camada do modelo OSI.

| Layer | Protocol |
|--------------|---------------------------------------|
| Application | HTTP, SMTP, FTP, Telnet |
| Presentation | ASCII, MPEG, JPEG, MIDI |
| Session | NetBIOS, SAP, SDP, NWLink |
| Transport | TCP, UDP, SPX |
| Network | IP, ICMP, ARP, RIP, IPX |
| Data Link | Ethernet, Token Ring, FDDI, AppleTalk |

Interação entre Protocolos

Como é o fluxo de dados através das camadas do modelo OSI? A transferência inicial dos dados em uma rede começa na camada de aplicação do sistema de transmissão. Os dados são passados as camadas inferiores do modelo OSI até atingir a camada física, neste ponto a camada física do sistema de transmissão envia os dados para o sistema de recepção. O sistema receptor captura os dados em sua camada física, e os repassa as camadas superiores até chegar a camada de aplicação.

Serviços prestados por diferentes protocolos em qualquer nível do modelo OSI não são redundantes. Por exemplo, se um protocolo em uma camada fornece um serviço particular, então nenhum outro protocolo em qualquer outra camada irá fornecer este mesmo serviço. Protocolos em camadas correspondentes, no envio e na recepção dos dados são complementares. Se na transmissão um protocolo na camada sete é responsável por criptografar os dados a serem transmitidos, o protocolo correspondente na camada sete da máquina receptora deverá ser responsável por decifrar os dados. A figura abaixo mostra uma representação gráfica do modelo OSI na comunicação entre dois computadores. Aqui você pode ver a comunicação que vai de cima para baixo, de um computador e então inverter quando ela atinge o outro computador.



Cada camada do modelo OSI só é capaz de se comunicar com as camadas imediatamente acima e abaixo dela. Por exemplo, a camada 2 só pode enviar e receber dados da camada 1 e da camada 3.

Encapsulamento de Dados

Os protocolos em diferentes camadas se comunicam com o auxílio do encapsulamento de dados . Cada camada na pilha é responsável por adicionar um cabeçalho ou rodapé aos dados serem transmitidos, é este bit de informação extra que permite a comunicação entre as camadas. Por exemplo, quando a camada de transporte recebe dados da camada de sessão, ele adiciona seu próprio cabeçalho de informação aos os dados antes de passá-lo para a próxima camada.

As Unidades de Dados dos Protocolos

O processo de encapsulamento cria uma unidade de dados dos protocolos (PDU), que inclui os dados enviados e todas as informações de cabeçalho ou rodapé adicionadas a ele.

Como os dados se movem de cima para baixo no modelo OSI, a PDU adicionará o cabeçalho e o rodapé com as informações do protocolo envolvido entre as camadas. A PDU estará em sua forma final quando atingir a camada física, nesse ponto ela será enviada ao computador de destino. O computador de destino retira o cabeçalho e o rodapé do protocolo envolvido entre as camadas do modelo OSI. Uma vez que a PDU atinge a camada superior do modelo OSI, restarão apenas os dados enviados.

Nota:

Pacote é o termo associado à Protocol Data Unit (PDU). Quando eu uso a palavra pacote, refiro-me a uma PDU completa que inclui as informações de cabeçalho e rodapé de todas as camadas do modelo OSI.

Hardware da Rede

Agora é hora de olhar para o hardware da rede, onde todo o trabalho sujo é feito. Vamos focar especificamente algumas das peças de hardware mais comuns em uma rede, hubs, switches e roteadores.

Hubs

Um hub é geralmente não mais que uma caixa com várias portas RJ-45, como mostrado na figura abaixo. Hubs variam de muito pequeno de quatro portas, até os maiores de 48 portas, projetados para serem montados em rack no ambiente empresarial. Hubs são projetados para conectar dispositivos de rede para que eles possam se comunicar.



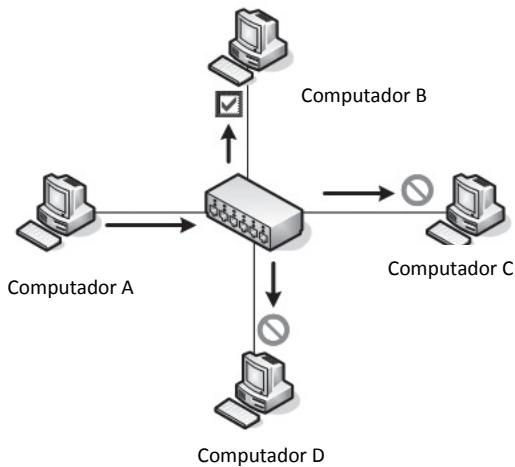
Um hub nada mais é do que um dispositivo de repetição que opera na camada física do modelo OSI. Um dispositivo de repetição simplesmente pega os pacotes enviados a partir de uma porta e transmite (repete) a outro dispositivo conectado a outra porta. Por exemplo, se um computador conectado a porta 1 de um hub de quatro portas tem que enviar dados a um computador conectado a porta 2, o hub envia os pacotes para todas as portas. Os clientes conectados às portas 3 e 4 ignoram os dados, porque não é para eles, descartando os pacotes. Nesse tipo de conexão o resultado é uma grande quantidade de tráfego desnecessário na rede.

Imagine que você está enviando um email aos empregados da empresa. O e-mail tem no campo assunto Relativo a todo o pessoal de marketing, mas em vez de enviá-lo apenas as pessoas que trabalham no departamento de marketing, você o envia para todos os funcionários da empresa. Os empregados que trabalham no marketing sabe que é pra eles e vão lê-lo. Os outros empregados, no entanto, verão que não é para eles, e vão descartá-lo. Você pode ver como isso é resultado de um monte de comunicação desnecessária e um desperdício de tempo, é exatamente assim como funciona um hub.

A figura abaixo fornece uma exibição gráfica do que está acontecendo aqui. Nesta figura, Um computador A está transmitindo dados para o computador B. No entanto, quando o computador A envia esses dados, todos os computadores conectados ao hub os recebem. Apenas o computador B realmente os aceita, os outros computadores vai descartá-lo.

Uma última nota sobre hubs é que eles só são capazes de operar em modo half-duplex, ou seja, eles não podem enviar e receber dados ao mesmo tempo. Isso os diferencia dos switches, que são dispositivos full-duplex que podem enviar e receber dados de forma síncrona.

Enquanto você não vai ver hubs sendo usado nas mais modernas ou rede de alto tráfego (switches são usados em vez disso, como discutido abaixo), você deve saber como os hubs trabalham, uma vez que eles serão muito importantes para a análise de pacotes.



Switches

A melhor alternativa ao hub em uma rede de alto tráfego é o dispositivo chamado switch. Como um hub, um switch é projetado para repetir os pacotes, mas faz isso de maneira muito diferente, também como um hub, um switch proporciona um caminho de comunicação entre dispositivos, mas o faz com mais eficiência. Ao invés de enviar um broadcast de dados para cada porta individualmente, somente envia os dados para o computador para quais os mesmos são destinados. Fisicamente falando, um switch parece idêntico a um hub. Por uma questão de fato, se os dispositivos não tiverem uma identificação visível, você terá dificuldade em diferenciá-los.

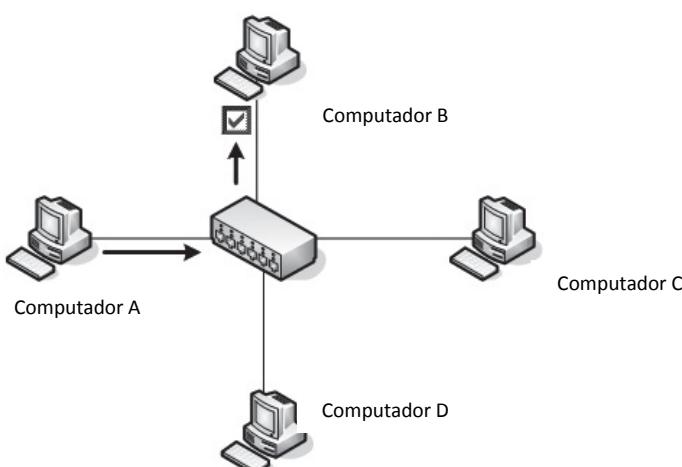
Muitos dos switches do mercado são gerenciáveis através de software específico ou interface web. Estes switches são referidos como switches gerenciados e oferecem vários recursos que podem ser úteis na gestão da rede. Isto inclui a capacidade de visualizar, habilitar ou desabilitar portas específicas, fazer mudanças de configuração e ser reiniciado remotamente.



Switches têm funcionalidades avançadas de manipulação dos pacotes transmitidos. É capaz de se comunicar diretamente com dispositivos específicos, switches são capazes de identificar dispositivos baseados em seus endereços. Tudo isso significa que devem operar na camada de enlace do modelo OSI.

Os Switches armazenam os endereços de camada 2 de cada dispositivo conectado em uma tabela CAM, que funciona como uma espécie de guarda de trânsito. Quando um pacote é transmitido, o switch lê as informações do cabeçalho da camada 2 do pacote e usando a tabela CAM como referência, determina para qual porta será enviado o pacote. Os switches apenas enviam pacotes para as portas envolvidas em uma comunicação, o que reduz consideravelmente o tráfego na rede.

A figura abaixo mostra uma representação gráfica do fluxo de tráfego através de um switch. Nesta figura, um computador A mais uma vez envia os dados para o computador B. Neste caso, os computadores estão conectados através de um switch que permite que o computador A envie os dados diretamente para o computador B sem outros dispositivos na rede estar ciente da presente comunicação. Além disso, várias conversas podem acontecer ao mesmo tempo.



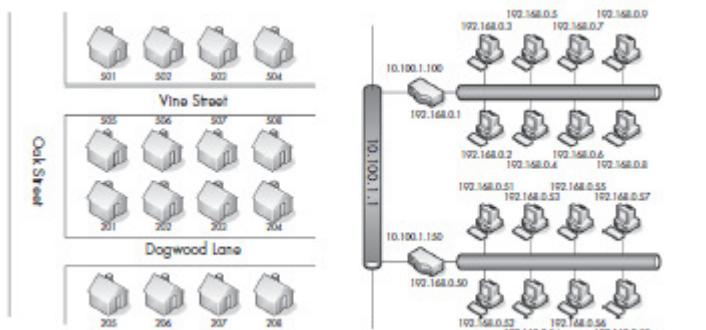
Roteadores

Um roteador é um dispositivo de rede avançada com um nível muito mais elevado de funcionalidade do que qualquer switch ou um hub. Um roteador pode ter várias formas, mas a maioria tem vários leds na frente e portas na parte traseira, dependendo do tamanho da rede. Roteadores operam na camada 3 do modelo OSI. Eles são responsáveis pelo envio de pacotes entre duas redes distintas. O processo que os roteadores usam para direcionar o fluxo de tráfego entre redes é chamado de roteamento.

Existem vários tipos de protocolos de roteamento que ditam como diferentes tipos de pacotes são encaminhados para outras redes. Roteadores normalmente usam endereços da camada 3 (tais como endereços IP) com exclusividade para identificar dispositivos em uma rede.

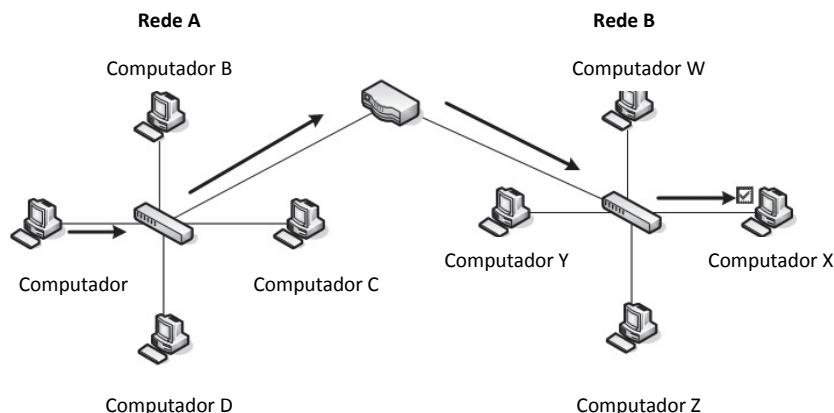
Uma maneira fácil de ilustrar o conceito de roteamento é pensar em um bairro com uma rede de ruas, cada rua tem casas, e cada casa tem seu próprio endereço. Você mora em uma rua, assim você pode mover-se entre todas as casas da rua. Isto é muito semelhante ao funcionamento de um switch que permite a comunicação entre todos os computadores em um segmento de rede. Para se comunicar com um vizinho em outra rua, no entanto, uma pessoa deve seguir o fluxo dos sinais da rua até a casa do vizinho.

Vamos trabalhar com um exemplo de comunicação através das ruas, digamos que eu estou sentado na 503 Vine Street, e eu preciso ir até a 202 Lane Dogwood. Para fazer isso, eu preciso atravessar para Oak Street, e em seguida, para Dogwood Lane. Pense nisso como uma passagem entre segmentos de rede. Se o dispositivo em 192.168.0.3 precisa se comunicar com o dispositivo em 192.168.0.54, ele deve atravessar um roteador para chegar à rede 10.100.1.1, em seguida, atravessar o segmento do roteador de destino antes que ele possa chegar ao segmento da rede de destino.



O tamanho e o número de roteadores em uma rede dependem do tamanho e função dessa rede. As redes domésticas ou de pequenos escritórios consistem de um pequeno roteador localizado no centro da rede, enquanto uma grande rede corporativa pode ter vários roteadores espalhados por vários departamentos, todos se conectando a um roteador central ou switch de camada 3. Um switch de camada 3 é um tipo avançado de switch que também tem uma funcionalidade interna para atuar como um roteador.

Começando a olhar para diagramas de rede cada vez mais, você vai compreender como os dados são encaminhados a diversos pontos. A figura abaixo mostra o layout de uma forma muito comum de rede roteada. Neste exemplo, duas redes separadas estão ligadas através de um único roteador. Se um computador na rede A desejar se comunicar com um computador na rede B, os dados transmitidos devem passar pelo roteador.



Classificação de Tráfego

Ao considerar o tráfego em uma rede, vamos classificá-lo em três classes principais: **broadcast**, **multicast** e **unicast**. Cada classificação tem uma característica distinta que determina como os pacotes de uma determinada classe são tratados pelo hardware de rede.

Tráfego Broadcast

Um pacote de broadcast é enviado a todos as portas de um segmento de rede, independentemente de saber se essa porta é um hub, switch ou roteador. Lembre-se que hub só utiliza tráfego de broadcast.

Tráfego Multicast

O Multicast é um meio de transmitir um pacote a partir de uma única fonte para múltiplos destinos simultaneamente. O objetivo do multicast é fazer com que este processo seja o mais simples possível, utilizando a largura de banda disponível. A otimização deste tráfego reside no número de vezes que um fluxo de dados é replicado para alcançar o seu destino. A manipulação exata do tráfego multicast é altamente dependente da implementação do protocolo usado. O principal método de execução multicast é usar um esquema de endereçamento especial que une os destinatários de pacotes em um grupo multicast, isto é como o IP multicast funciona. Esse esquema de endereçamento garante que os pacotes não serão transmitidos a computadores que não façam parte do grupo de endereçamento.

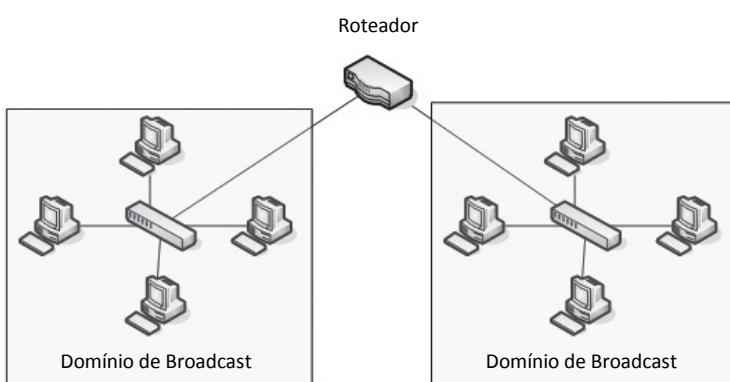
Tráfego Unicast

Um pacote unicast é transmitido diretamente de um computador para outro. Os detalhes de como funciona o unicast dependem do protocolo usado.

Domínio de Broadcast

Lembre-se que um pacote de broadcast é aquele que é enviado para cada dispositivo em segmento particular. Em redes maiores, com vários hubs ou switches conectados via diferentes meios, os pacotes transmitidos de um switch através de todo o caminho até outras portas de outros switches na rede, são repetidos de switch a switch.

A extensão que os pacotes percorrem é chamada de domínio de broadcast - esse é o segmento de rede onde qualquer computador pode transmitir diretamente para outro computador sem passar por um roteador. A figura abaixo mostra um exemplo de dois domínios de broadcast em uma rede pequena. Porque cada domínio de broadcast estende-se até que ele encontre o roteador, os pacotes de broadcast circularão apenas dentro deste domínio de broadcast especificado.



Nosso exemplo anterior descrevendo como o roteamento refere-se a um bairro também fornece uma boa visão sobre como funcionam os domínios de broadcast. Você pode pensar em um domínio de broadcast como sendo uma rua do bairro. Se você ficar na varanda e gritar, apenas as pessoas da sua rua vão ser capazes de ouvir você. Se você quiser falar com alguém em uma rua diferente, você tem que encontrar uma maneira de falar com essa pessoa diretamente, ao invés de usar o broadcast a partir de sua varanda.

As coisas que você aprendeu até aqui são os princípios básicos da análise de pacotes. Você deve entender o que está acontecendo neste nível de comunicação de rede antes de começar a solucionar os seus problemas. No próximo capítulo aprimoraremos mais estes conceitos e discutiremos sobre os princípios avançados de comunicações de rede.

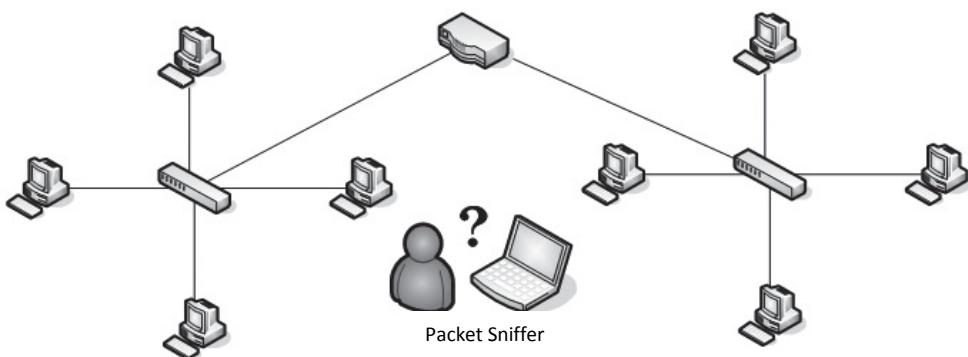
2

CHACOALHANDO OS FIOS

Podemos agora passar para a etapa final da preparação, antes de começarmos a capturar online os pacotes na rede. Esta última etapa é descobrir o local mais adequado para colocar um sniffer no sistema de cabeamento da rede. Isso é muitas vezes referido pelos analistas como a obtenção de pacotes no fio, batendo na rede, ou batendo no fio. Basta colocar, este é o processo de colocação de um farejador (sniffer) de pacote em uma rede no local físico correto.

Infelizmente, os pacotes de sniffing não é tão simples como ligar um laptop em uma porta de rede e ir capturando o tráfego (Figura abaixo). Na verdade, às vezes é mais difícil colocar um sniffer no sistema de cabeamento de uma rede que realmente analisar os pacotes capturados.

O desafio com a colocação do sniffer é que existe uma grande variedade de hardware de rede que é usado para conectar dispositivos. Porque os três principais dispositivos em uma moderna rede (hubs, switches e roteadores) todos lidam diferentemente com o tráfego, você deve estar muito consciente da instalação física da rede que você está analisando.



O objetivo deste capítulo é ajudá-lo a desenvolver uma compreensão da colocação de um farejador (sniffer) de pacotes em uma variedade de topologias de rede diferentes. Nós vamos olhar várias configurações de rede e determinar a melhor maneira para capturar pacotes em ambientes baseado em um hub, switch, roteador. Como um precursor para a compreensão da colocação do farejador (sniffer), vamos também obter uma forma mais aprofundada de olhar para as placas de rede em modo promíscuo, como elas funcionam, e por que elas são uma necessidade para a análise de um pacote.

Modo Promíscuo

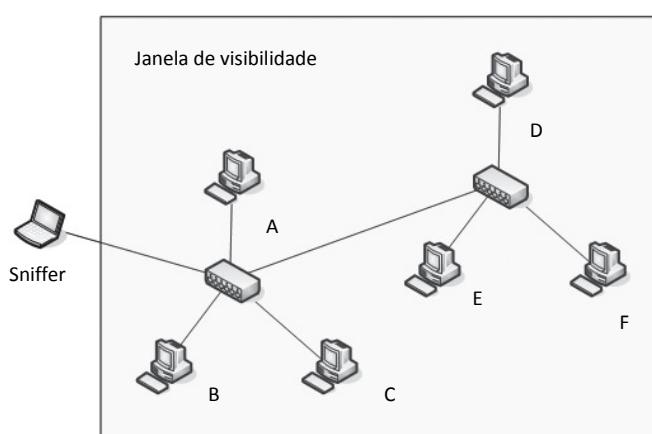
Antes que você possa capturar os pacotes em uma rede, você precisará de um cartão de interface de rede (NIC) que suporte um driver de modo promíscuo. É o modo promíscuo que permite uma NIC ver todos os pacotes que atravessam o sistema de cabeamento. Quando uma placa de rede não está no modo promíscuo, ela geralmente vê uma grande quantidade broadcast e outros tráfegos que não é dirigida a ela, que serão descartados. Quando está no modo promíscuo, ela captura tudo e passa todo o tráfego que recebe para a CPU, basicamente ignorando as informações que se encontram na camada 2. Seu farejador (sniffing) de pacotes agarra todos os pacotes para dar-lhe um relato completo e preciso de todos os pacotes no sistema.

Nota:

A maioria dos sistemas operacionais (incluindo o Windows) não vai deixar você usar uma placa de rede em modo promíscuo a menos que você tenha privilégios de administrador. Se você não pode obter esses privilégios em um sistema, é possível que você não realize qualquer tipo de farejamento (sniffer) de pacotes em uma rede.

Farejando através de um Hub

Farejar (sniffing) em uma rede que tem hubs instalados é um sonho para qualquer analista de pacotes. Como você aprendeu anteriormente, o tráfego através de um hub é enviado para todas as portas conectadas ao hub. Portanto, para analisar com um computador ligado a um hub, tudo que você precisa fazer é conectar o farejador (sniffer) de pacotes em uma porta vazia do hub, e você poderá ver todas as comunicações de/e para todos os computadores conectados ao hub. Conforme ilustrado na figura abaixo, a sua janela de visibilidade é ilimitada quando o farejador (sniffer) está conectado a uma rede através de um hub.



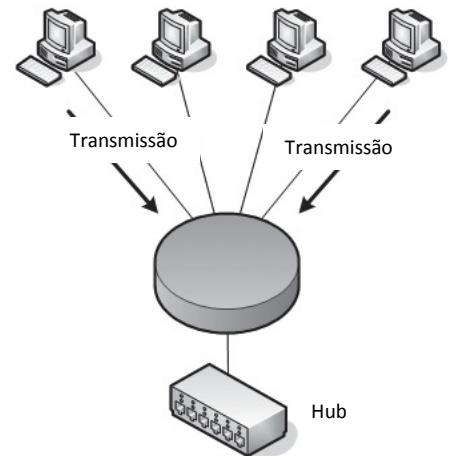
Nota:

A janela de visibilidade, como mostrada nos diagramas ao longo deste livro, mostra os dispositivos de uma rede, cujo tráfego será capaz de ser visto com um farejador (sniffer) de pacotes.

Infelizmente para nós, redes baseadas em hubs são muito raras, devido à dor de cabeça que causam aos seus administradores. Hubs tendem a deixar o tráfego da rede lento, pois apenas um dispositivo pode usar o hub por vez, portanto, um dispositivo conectado através de um hub tem de competir por largura de banda com os outros dispositivos também tentando se comunicar através dela. Quando dois ou mais dispositivos tentam se comunicar ao mesmo tempo, colidem pacotes (como mostrado na figura abaixo) e pacotes transmitidos são perdidos e devem ser retransmitidos.

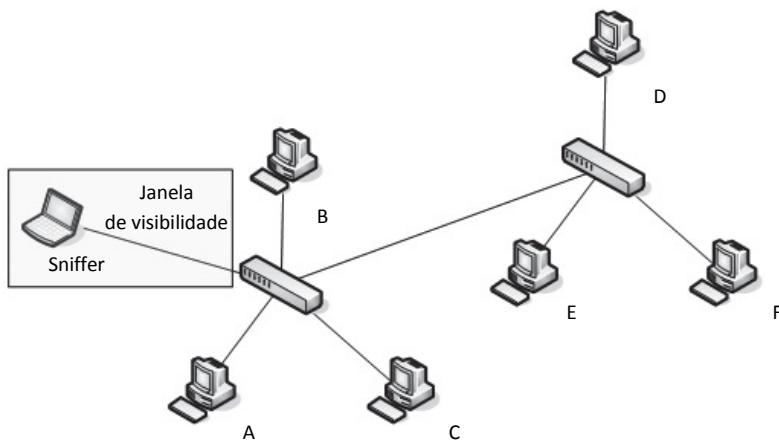
Com o aumento de colisões, o desempenho da rede pode diminuir drasticamente. Como o nível de tráfego aumenta as colisões, dispositivos podem tentar transmitir um pacote três ou quatro vezes. É por isso que as redes mais modernas de qualquer tamanho usam switches.

A única preocupação que temos que considerar quando estamos farejando (sniffing) pacotes em um computador individual em uma rede utilizando um hub, é a captura do grande volume de tráfego. Uma vez que uma placa de rede em modo promiscuo vê todo o tráfego indo e vindo de todos os dispositivos em um hub, você vai ter uma quantidade muito grande de dados para pesquisar, a maioria dos quais são irrelevantes. No próximo capítulo você vai aprender como aproveitar o poder de capturar e exibir esses dados utilizando filtros, a fim de realizar sua análise de forma mais eficiente.



Farejando através de um Switch

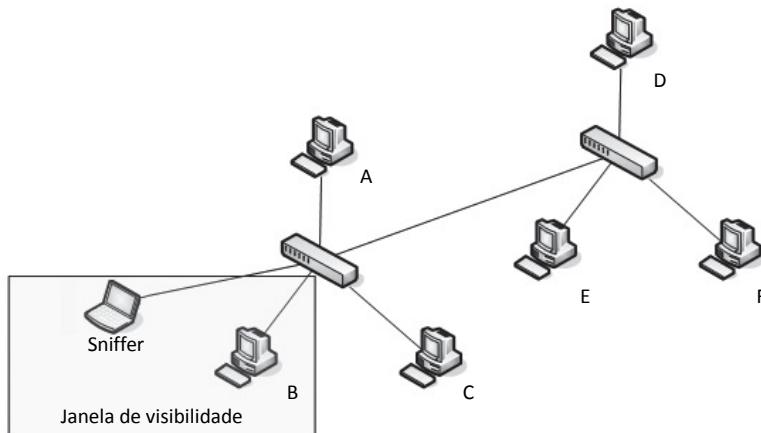
Um ambiente com switch é o tipo mais comum de rede que você estará trabalhando. Switches proporcionam um meio eficaz de transporte de dados via broadcast, o tráfego unicast e multicast. (Para mais informações sobre estes temas ver Capítulo 1). Como bônus, switches permitem a comunicação full-duplex, significando que as máquinas podem enviar e receber dados simultaneamente através do mesmo. Infelizmente para analistas de pacotes, switches adicionam um novo nível de complexidade ao seu trabalho. Quando você conecta um farejador (sniffer) de pacotes a uma porta de um switch, você só pode ver o tráfego broadcast e o tráfego transmitidos e recebidos por sua máquina, como mostrado na figura baixo.



Existem três formas principais de capturar o tráfego de um dispositivo em uma rede com switch: espelhamento de porta, o envenenamento de cache ARP, e hubbing.

Espelhamento de Porta

Espelhamento de porta (Port mirroring), ou a medição de porta (Port Spanning) como muitas vezes é chamado, é talvez a maneira mais fácil de capturar o tráfego de um dispositivo em uma rede através de um switch. Neste tipo de instalação, você deve ter acesso à interface de linha de comando do switch em que o computador está localizado. Além disso, o switch terá que suportar o espelhamento de porta. Para que você possa através de linha de comando forçar a opção de cópia de todo o tráfego em uma determinada porta para outra porta (ver figura baixo). Por exemplo, para capturar o tráfego de um dispositivo na porta três de um switch, você pode simplesmente ligar o farejador (sniffer) na porta quatro e fazer o espelhamento da porta três. Isso permitirá que você veja todo o tráfego transmitido e recebido pelo seu dispositivo ligado na porta três. O comando exato que você irá digitar para configurar o espelhamento de porta variará dependendo do fabricante do switch que você está usando. Você vai encontrar uma lista de comandos de espelhamento de portas mais comuns na tabela baixo.



Quando estiver utilizando o espelhamento de porta, tenha cuidado com a quantidade de portas que estiverem sendo espelhadas. Alguns fabricantes permitem que você faça o espelhamento de múltiplas portas em uma única, isso pode ser muito útil quando se analisa a comunicação entre dois ou mais dispositivos em um único switch. No entanto, consideremos o que vai acontecer com um pouco de matemática básica. Por exemplo, se você tiver um switch de 24 portas e você espelha 23 portas de 100Mbps full-duplex para uma porta, você pode ter, potencialmente, 4600Mbps fluindo para essa porta. Isto obviamente vai muito além do limite físico de uma única porta e pode causar perda de pacotes ou de lentidão da rede se tráfego atingir um determinado nível. Nestas situações os switches descartarão o excesso de pacotes, impedindo a comunicação. Certifique-se que este tipo de situação não ocorra quando você tentar executar a sua captura.

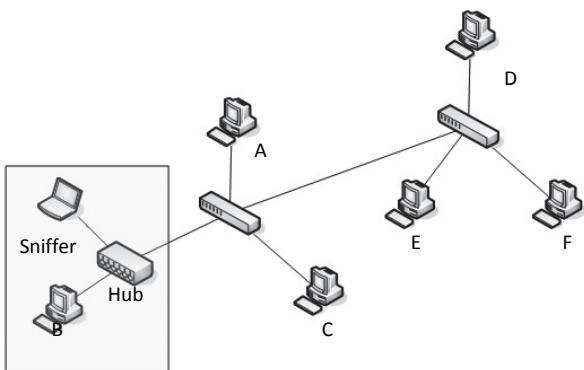
Table 2-1: Commands Used to Enable Port Mirroring for Different Manufacturers' Switches

| Manufacturer | Port Mirroring Command |
|--------------|---|
| Cisco | set span <source port> <destination port> |
| Enterasys | set port mirroring create <source port> <destination port> |
| Nortel | port-mirroring mode mirror-port <source port> monitor-port <destination port> |

Hubbing

Outra maneira muito simples de capturar o tráfego através de um dispositivo em uma rede com switch é o hubbing. O Hubbing é uma técnica na qual você coloca o dispositivo desejado e seu sistema farejador (sniffer) no mesmo segmento de rede, ligando-os diretamente a um hub. Muitas pessoas pensam que o uso de hub nos dia está em desuso, mas realmente a sua utilização é uma perfeita solução em situações onde você não pode executar o espelhamento de porta, mas possui um hub para ligar o dispositivo desejado e o farejador (sniffer) na rede.

Para utilizar o Hubbing, tudo o que você precisa é de um próprio hub e alguns cabos de rede. Depois de ter tudo á mãos, vá até onde o dispositivo desejado está plugado e desconecte-o. Em seguida, conecte-o ao hub junto com o seu dispositivo farejador (sniffing). Em seguida, ligue o hub à rede, conectando-o ao switch. Agora você tem que basicamente colocar o dispositivo desejado e o farejador (sniffing) no mesmo domínio de broadcast, e todo o tráfego enviado ao dispositivo desejado será transmitido também ao farejador (sniffing) de forma que o mesmo poderá capturar os pacotes como mostrado na figura abaixo.



Na maioria das situações, a utilização da técnica de Hubbing reduz a forma de transmissão duplex, a simples half-duplex. Enquanto este método não é o melhor caminho para farejar os fios, às vezes será a sua única opção quando um switch não suportar o espelhamento de porta.

Nota:

Como um lembrete, geralmente é um gesto simpático alertar ao usuário do dispositivo desejado, que você irá desconectá-lo e monitorado, especialmente se o usuário for o seu chefe!

Quando estiver utilizando a técnica Hubbing, certifique-se que você está usando um hub de verdade e não um falso switch. Vários fornecedores de hardware de rede têm o mau hábito de marketing de venda de um dispositivo hub quando ele realmente funciona como um switch. Se você não está trabalhando com um hub, comprovadamente testado, você só vai ver seu próprio tráfego, e não do dispositivo de desejo. Quando você encontrar um hub, teste-o para certificar-se que realmente é um hub! A melhor maneira de determinar ou não se o dispositivo que você está usando é um verdadeiro hub é só ligar um par de computadores nele e ver se você pode farejar o tráfego dos dois. Se conseguir, você tem um verdadeiro hub em sua posse.

Envenenamento do Cache ARP

Lembre-se do capítulo 1, que os dois principais tipos de pacotes de endereçamento estão nas Camadas 2 e 3 do modelo OSI. Esta camada 2 de endereçamento, ou endereço MAC, é usada em conjunto com qualquer sistema de endereçamento da camada 3 que você está usando.

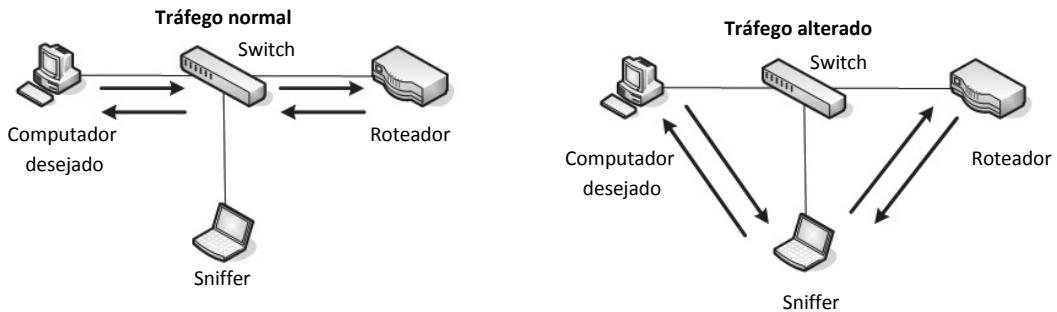
No caso deste livro (e do padrão da indústria), refiro-me ao Sistema de endereçamento de camada 3, o Protocolo IP.

Todos os dispositivos em uma rede se comunicam entre si na camada 3 usando endereços IP. Como switches operam na camada 2 do modelo OSI, eles devem ser capazes de traduzir endereços da camada 2 (MAC) em endereços da camada 3 (IP) e vice-versa, a fim de ser capaz de encaminhar o tráfego para o dispositivo apropriado. Este processo de tradução é feito através de um protocolo de camada 3 conhecido como o Protocolo de Resolução de Endereços o ARP.

Quando um computador precisa enviar dados para outro, ele envia um ARP pedindo ao switch onde está ligado. O switch envia um pacote broadcast ARP para todos os computadores conectados a ele, pedindo que cada computador cheque o seu endereço IP. Quando o computador de destino vê este pacote, ele se identifica com o endereço IP e envia o seu endereço MAC. O switch tem agora uma rota estabelecida até computador de destino, e qualquer dispositivo que desejar se comunicar com este computador de destino pode utilizar essa rota. Estas informações recém obtidas são armazenadas no cache ARP do switch, evitando assim o novo envio de um pacote broadcast ARP, cada vez que precisar enviar dados para esse computador.

O envenenamento de cache ARP é uma forma mais avançada de farejar (sniffer) os fios em uma rede comutada. É comumente utilizado por hackers para enviar falsamente pacotes endereçados aos sistemas do cliente, a fim de interceptar os tráfegos da negação de serviço (DoS), mas o envenenamento de cache ARP pode ainda servir como uma forma legítima de capturar os pacotes de uma máquina de destino em uma rede que utiliza switch.

O envenenamento de cache ARP, por vezes referido como **ARP spoofing**, é o processo de envio de mensagens ARP para um switch ou roteador Ethernet com MAC falso, a fim de interceptar o tráfego de outro computador, como mostrado abaixo.

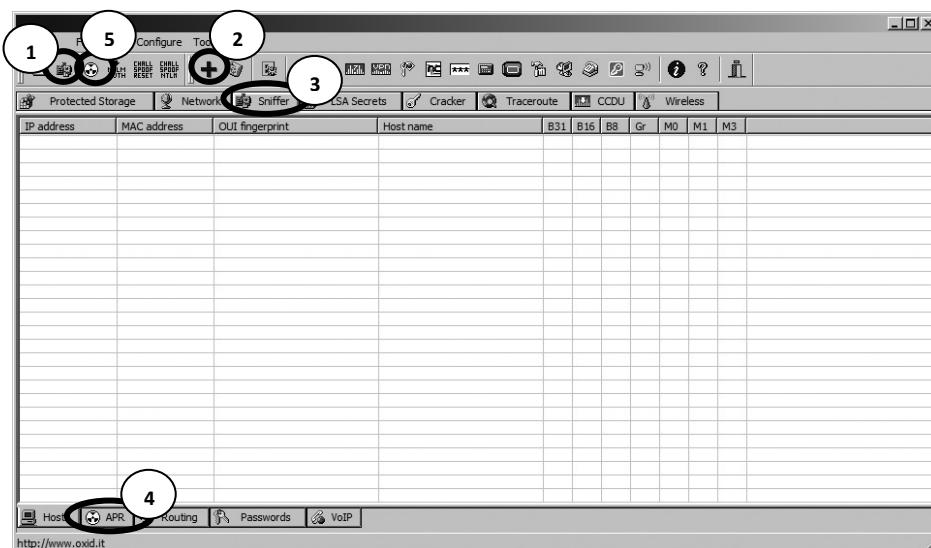


Usando o Cain & Abel

Ao tentar envenenar o cache ARP, o primeiro passo é baixar as ferramentas necessárias e recolher algumas informações necessárias. Nós vamos usar a popular ferramenta de segurança Cain & Abel da Oxdit (<http://www.oxid.it>). Vá em frente e instale-o agora.

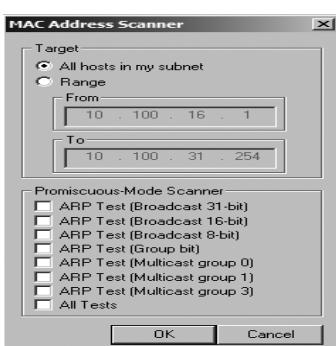
Uma vez que você tenha instalado o software Cain & Abel, você precisa coletar algumas informações adicionais, incluindo os endereços IP de seu sistema analisador, o sistema (dispositivo) remoto que você deseja capturar o tráfego, e o roteador ao qual o mesmo está ligado.

Quando você abre o Cain & Abel, você vai notar uma série de guias perto da janela superior. O envenenamento de cache ARP é apenas uma das várias opções do Cain & Abel. Para os nossos propósitos, vamos estar trabalhando com a aba **Sniffer**. Quando clicar nesta guia, você verá uma tabela vazia, como a mostrada abaixo.



Para preencher esta tabela você terá primeiro que ativar a opção **sniffer** do Cain & Abel e scanear sua rede a procura dos dispositivos conectados a ela. Para fazer isso, siga estes passos:

1. Clique no segundo ícone (1) na barra de ferramentas, que se assemelha a uma placa de rede. A primeira vez que você fizer isso você será solicitado a selecionar a interface que deseja farejar (sniffing). Esta deve a interface que está conectada na rede, dessa forma você estará realizando o envenenamento de cache ARP.
2. Uma vez que você selecionou esta interface, clique em **OK** para ativar os farejadores Cain & Abel.
3. Para criar uma lista de hosts disponíveis em sua rede, clique no ícone (+) (2) e clique em **OK** como mostrado abaixo.



A grade vazia agora deverá ser preenchida com uma lista de todos os hosts encontradas em sua rede, juntamente com os respectivos endereços MAC, endereços IP, e identificação do fabricante. Esta é a lista que você irá trabalhar quando configurar o envenenamento de cache ARP.

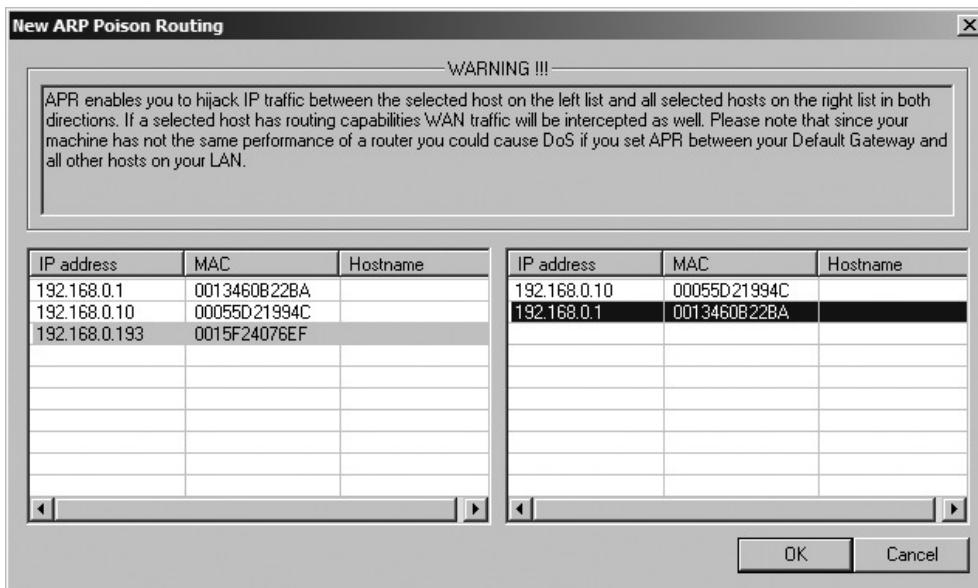
Na parte inferior da janela do programa, você verá um conjunto de guias que faz parte da opção Sniffer (3). Agora que você construiu sua lista de host, você irá trabalhar a partir da guia **APR** (4). Alterne para a janela **APR** clicando na guia.

Uma vez na guia **APR**, serão apresentadas duas tabelas vazias: uma superior e outra inferior. Depois de configurá-las, a tabela superior mostrará os dispositivos envolvidos no seu envenenamento de cache ARP, e a tabela inferior mostrará toda a comunicação entre as máquinas envenenadas.

Para configurar o seu envenenamento, siga estes passos:

1. Clique no ícone (+) na barra de ferramentas padrão. A janela que aparece tem duas colunas laterais de seleção lado a lado.
2. No lado esquerdo, você verá uma lista de todos os hosts disponíveis na rede. Clique no endereço IP do dispositivo desejado, cujo tráfego que você deseja farejar (sniffing). Isso resultará na janela da direita, uma lista de todos os outros hosts existentes na rede, sendo omitido o endereço IP do dispositivo já escolhido.
3. Na janela da direita, clique no endereço IP do roteador que o dispositivo escolhido está conectado e clique em **OK**, ver figura abaixo. Os Endereços IPs de ambos os dispositivos devem agora ser listados na tabela superior na janela principal.
4. Para concluir o processo, clique no símbolo de radiação (5) amarelo e preto na barra de ferramentas padrão. Isso irá ativar os recursos de envenenamento de cache ARP do Cain & Abel, que permitirá que o seu sistema de análise intermedie as comunicações entre o dispositivo escolhido e o roteador.

Agora você pode carregar seu analisador de pacotes (Wireshark) e iniciar o processo de análise. Quando você terminar de capturar o tráfego, basta clicar no símbolo de radiação (5) amarelo e preto para parar o envenenamento de cache ARP.



Como nota final sobre o envenenamento de cache ARP, você deve estar muito consciente das regras do sistema em análise para implementar este processo. Por exemplo, não usar esta técnica quando for muita alta a utilização do dispositivo desejado, como um servidor de arquivos que está ligado a rede por link 1Gbps (especialmente se link do seu sistema analisador for de apenas 100Mbps). Quando você executar este reencaminhamento do tráfego, todo o tráfego transmitido e recebido pelo sistema desejado deve primeiro passar por seu analisador, tornando seu analisador o gargalo no processo de comunicação. Isso pode criar um efeito do tipo DoS na máquina que você está analisando, o que resultará no desempenho da rede degradado, consequentemente falha na análise dos dados.

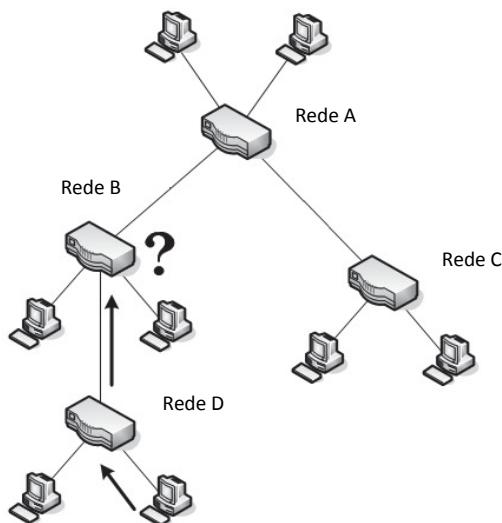
Farejando através de um Roteador

Todas as técnicas para farejar os fios em uma rede comutada estão disponíveis em redes roteadas também. A única consideração importante quando se lida com ambientes roteados é a importância da colocação (localização) do farejador (sniffer) quando você está solucionando um problema que abrange vários segmentos de rede.

Como você aprendeu anteriormente, o domínio de um dispositivo de transmissão se estende até um roteador. Neste ponto, o tráfego é entregue para o próximo roteador e você perde a comunicação com os pacotes

que estão sendo transmitidos, até que você receba uma confirmação de seu recebimento. Em situações como esta, onde dados devem atravessar vários roteadores, é importante analisar o tráfego em todos os lados do roteador.

Por exemplo, considere o problema de comunicação que você pode encontrar em uma rede com vários segmentos de redes ligados através de uma variedade de roteadores. Nesta rede, cada segmento se comunica com outro segmento para armazenar e recuperar dados. O problema que estamos tentando resolver é que um dispositivo na rede D, não pode se comunicar com um dispositivo que se encontra na rede A, como mostrado abaixo.



Seu instinto poderia dizer-lhe para capturar o tráfego de um dispositivo no segmento D. Quando você fizer isso, você pode ver claramente que os dados estão sendo transmitidos ao segmento A, mas sem o recebimento de confirmação. Quando farejar (sniffing) o próximo segmento de rede para encontrar a fonte do problema, você verá que o tráfego é descartado pelo roteador da rede B. Eventualmente, isto leva a um problema de configuração no roteador, quando corrigido, resolve o seu maior dilema. Este é um excelente exemplo de que muitas vezes é necessário capturar o tráfego de vários dispositivos em vários segmentos, a fim de identificar um problema.

Mapas de Rede

Em nossa breve discussão sobre o posicionamento da rede, já olhamos vários mapas diferentes de rede. Um mapa de rede ou diagrama de rede, é um diagrama que mostra todos os recursos técnicos na rede e como eles estão conectados.

Não há melhor maneira de determinar a colocação de seu analisador de pacotes, além de ser capaz de visualizar a rede de forma clara. Se você tem um mapa de rede disponível para você, eu recomendo mantê-lo acessível, como se tornará um recurso valioso na solução de problemas e processo de análise. Você pode mesmo fazer um mapa detalhado de sua própria rede. Lembre-se: às vezes meio caminho andado na solução de um problema é sua identificação.

3

INTRODUÇÃO AO WIRESHARK

Existem diferentes farejadores (sniffing) de pacotes disponíveis para análise de desempenho de uma rede, mas nós vamos estar usando o Wireshark ao longo deste livro. Este capítulo discute a história do Wireshark, bem como seus benefícios, a instalação, e o seu uso básico.

Uma Breve História sobre o Wireshark

O Wireshark tem uma história muito rica. Gerald Combs, um pós-graduado de ciência da computação da Universidade de Missouri, em Kansas City, originalmente o desenvolveu fora de suas necessidades. A primeira versão do aplicativo Combs, chamado Ethereal, foi lançado em 1998 sob a GNU Public License (GPL).

Oito anos depois de lançar o Ethereal, Combs deixou seu trabalho para perseguir outras oportunidades na carreira. Infelizmente, o seu empregador na época tinha plenos direitos sobre a marca Ethereal, e Combs não conseguiu chegar a um acordo que lhe permitiria controlar a marca Ethereal. Em vez disso Combs e o resto da equipe de desenvolvimento, rebatizaram o projeto como Wireshark, em meados de 2006.

O Wireshark tem crescido dramaticamente em popularidade, e sua equipe de desenvolvimento já possui mais de 500 colaboradores. O programa que existe sob o nome Ethereal não está mais sendo desenvolvido.

Os Benefícios do Wireshark

O Wireshark oferece vários benefícios que o tornam atraente para o uso diário. Ele Destina-se tanto ao curioso e ao especialista em análise de pacotes e oferece uma variedade de recursos para seduzir cada um. Vamos examinar o Wireshark de acordo com os critérios definidos no Capítulo 1 para selecionar um a ferramenta farejadora (sniffing) de pacotes.

Protocolos Suportados

O Wireshark excede o número de protocolos que ele suporta, mais de 850, como está escrito. Estes protocolos como o IP e o DHCP são os mais comuns que os mais avançados protocolos proprietários, como o AppleTalk e BitTorrent. É porque o Wireshark é desenvolvido no âmbito de um modelo de fonte aberta, o suporte de um novo protocolo é adicionado a cada atualização. Se existe um protocolo que o Wireshark não suporta, você pode codificá-lo e submeter seu código aos desenvolvedores do Wireshark para inclusão na aplicação (se o seu código é aceito, é claro). Dito isto, não há realmente nenhum protocolo que o Wireshark não seja capaz de suportar.

Uso Amigável

A interface do Wireshark é um das mais fáceis de entender qualquer outra ferramenta farejadora (sniffing) de pacotes. O Wireshark é um aplicativo baseado em GUI com muita clareza escrito com menus de contexto e um layout simples. Ela também fornece vários recursos projetados para melhorar seu uso, como a cor do protocolo baseado em codificação e detalhada representações gráficas de dados brutos. Ao contrário de algumas das mais complicadas linhas de comandos alternativas como o tcpdump, o Wireshark GUI é melhor para aqueles que estão apenas entrando no mundo da análise de protocolo.

Custo

Uma vez que é um open source, o preço do Wireshark não pode ser batido. O Wireshark é liberado como software livre sob a GPL. Você pode baixar e usar o Wireshark para qualquer finalidade, seja pessoal ou comercial.

Suporte ao Programa

Pode ser feito ou não a nível do software. Quando se tratar de software distribuído livremente como o Wireshark, muitas vezes não há apoio formal, razão pela qual a comunidade de código aberto, muitas vezes depende de seu usuário para oferecer esse suporte. Felizmente para nós, a comunidade Wireshark é uma das melhores e mais ativa de que qualquer outro projeto de código aberto. A página do Wireshark na internet permite ligações diretas a várias formas de suporte, incluindo documentação on-line, o suporte e desenvolvimento wiki, FAQs, é um lugar para se inscrever na lista de distribuição de email do Wireshark, que é acompanhada pela maioria dos programas desenvolvedores de topo. Estes desenvolvedores, juntamente com a comunidade de usuários do Wireshark, provêem suporte que não deixa nenhuma pergunta sem resposta.

Suporte aos Sistemas Operacionais

O Wireshark suporta todos os principais sistemas operacionais modernos, incluindo Windows, Mac OS X e Linux. Você pode ver uma lista completa dos sistemas operacionais suportados na home page do Wireshark.

Instalando o Wireshark

O processo de instalação do Wireshark é surpreendentemente simples. Nesta secção nós olharemos os requisitos do sistema e, em seguida, percorrer as etapas envolvidas na instalação do Wireshark no Windows e Linux.

Requerimentos do Sistema

Antes de instalar o Wireshark, você deve se certificar de que o sistema atende aos seguintes requisitos:

- Um processador de 400 MHz ou mais rápido;
- Pelo menos 60MB de espaço de armazenamento disponível;
- Uma NIC (placa de rede) que suporte o modo promiscuo;
- WinPcap capture driver;

O drive de captura WinPcap é a implementação do Windows do Pcap pacote de interface de captura de interface de programação (API). Simplificando, este driver interage com seu sistema operacional para capturar pacotes de dados, aplicar filtros, e mudar a placa de rede dentro e fora do modo promiscuo. Você pode encontrar o pacote de instalação para o driver em <http://www.winpcap.org>.

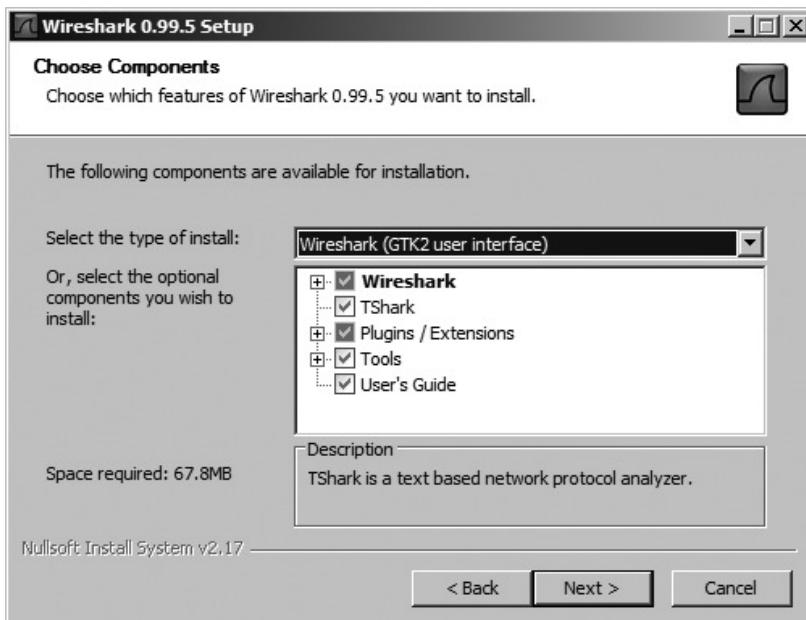
Nota:

Embora você possa baixar separadamente o WinPcap. É melhor instalá-lo a partir da instalação do Wireshark, porque a versão incluída ao instalar o Wireshark já vem testada para trabalhar em conjunto com o mesmo.

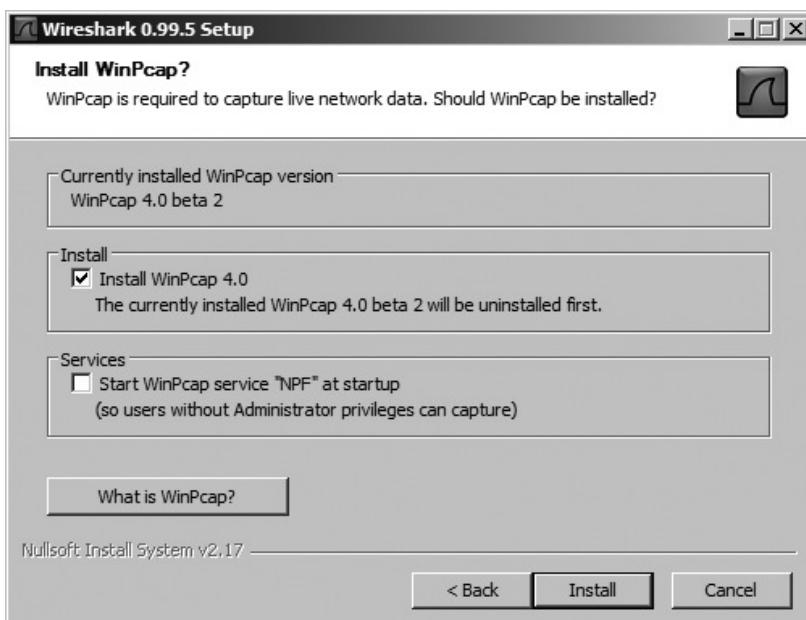
Instalando no Windows

O primeiro passo ao instalar o Wireshark no Windows é obter o instalação compilada mais recente na página web oficial Wireshark, <http://www.wireshark.org>. Navegue até a seção Downloads, e escolha um site para fazer o download. Uma vez que você tenha baixado o pacote, siga estes passos:

1. Duplo clique no arquivo executável. para iniciar a instalação e clique em Avançar na janela introdutória.
2. Leia o contrato de licenciamento e clique em Concordo se você concordar.
3. Selecione os componentes do Wireshark que você deseja instalar. Para nossos propósitos, Você pode aceitar o padrão clicando em Next (como mostrado na figura abaixo).



4. Clique em Avançar na janela de tarefas adicionais.
5. Selecione o local onde você deseja instalar o Wireshark e clique em Avançar.
6. Certifique-se que a opção instalar o WinPcap esteja marcada, e clique em Install. O processo de instalação deve começar.



7. No meio da instalação do Wireshark, a instalação WinPcap deverá começar. Quando isso acontecer, clique em Avançar na janela de introdução. Então leia o contrato de licença e clique em Concordo.
8. O WinPcap será instalado em seu computador. Assim que tiver terminado, clique Concluir.
9. O Wireshark deverá concluir a sua instalação. Uma vez feito isso, clique em Avançar.
10. Uma vez que a janela de confirmação de instalação aparecer, clique em Concluir.

Instalando no Linux

O primeiro passo na instalação Wireshark em um sistema Linux é baixar o pacote de instalação apropriado. Nem todas as versões do Linux são suportadas, não se surpreenda se a sua distribuição específica não tiver o seu próprio pacote de instalação.

Sistemas Baseados em RPM

Para instalar o Wireshark em distribuições baseadas em RPM, como Red Hat, faça o seguinte:

1. Baixe o pacote de instalação apropriado do Wireshark na página web.
2. Abra uma janela do console e digite rpm-ivh wireshark 0.99.3.i386.rpm, substituindo o nome do seu pacote baixado, conforme apropriado.
3. Se todas as dependências estão faltando, instalá-los e repetir a etapa anterior.

Sistemas Baseados em DEB

Para instalar o Wireshark em uma distribuição baseada em DEB, como Debian ou Ubuntu, faça o seguinte:

1. Baixe o pacote de instalação apropriado do Wireshark página web.
2. Abra uma janela do console e digite apt-get install wireshark.

Fundamentos do Wireshark

Uma vez que você instalou com sucesso o Wireshark em seu sistema, você pode começar a se familiarizar com ele. Agora você finalmente poderá conseguir o completo funcionamento do seu farejador (sniffing) de pacotes. O fato é que o Wireshark não é muito interessante quando você abri-lo pela primeira vez. E realmente para que as coisas se tornem emocionantes, você tem que obter (capturar) alguns dados (pacotes).

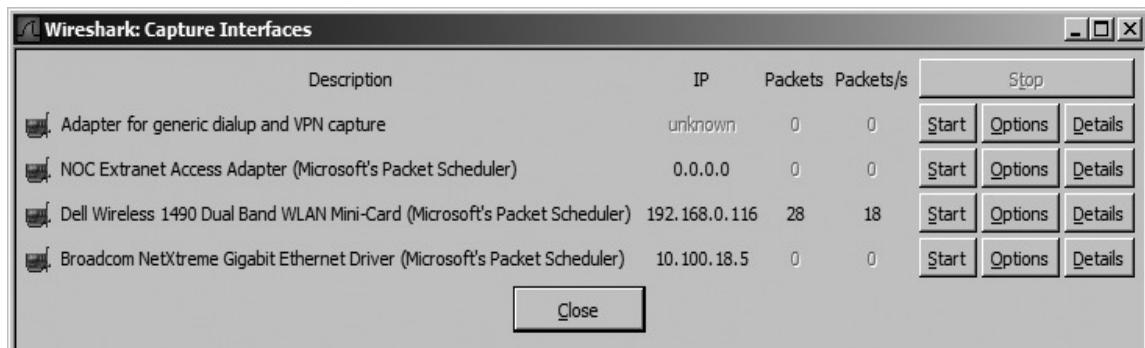
Sua Primeira Captura de Pacotes

Iniciando o uso do Wireshark, você vai realizar a sua primeira captura de pacote. Você pode estar pensando: "Como eu vou capturar pacotes, quando nada há de errado na rede? "Existem duas coisas erradas com esta declaração. A primeira coisa é que sempre há algo de errado na rede. Se você não acredita em mim, então vá em frente e envie um email para todos os seus empregados e informe a eles que tudo está funcionando perfeitamente.

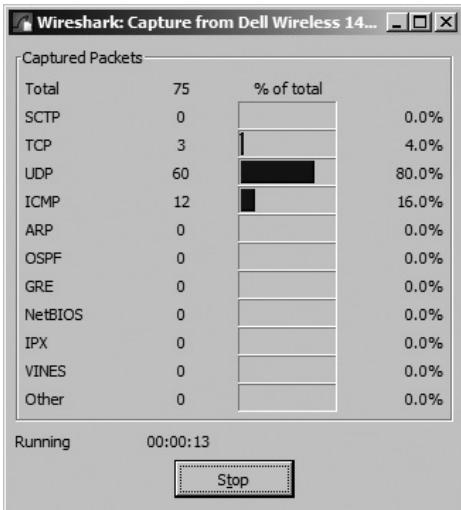
Em segundo lugar, não tem que haver algo de errado para que você possa realizar a análise de pacotes. Na verdade, a maioria dos analistas de rede passa mais tempo analisando o tráfego sem problemas, do que o tráfego com problemas. Você precisa de uma base para comparar a fim de ser capaz efetivamente de solucionar problemas de tráfego em uma rede. Por exemplo, se você sempre esperar para resolver um problema de DHCP analisando seu tráfego, você deve compreender com o que, o fluxo de tráfego DHCP se parece. Mais amplamente, a fim de detectar anomalias no tráfego diário da rede, você necessita conhecer a atividade normal diária da sua rede. Quando sua rede está funcionando corretamente, você poderá levar em consideração para análise de um estado normal.

Nós cobrimos o básico. Agora vamos capturar alguns pacotes!

1. Abra o Wireshark.
2. No menu principal drop-down, selecione Capture e, em seguida, Interfaces. Você deverá ver uma caixa de diálogo listando as várias interfaces que podem ser usados para captura de pacotes, juntamente com os respectivos endereços IP. Escolha a interface que deseja usar e clique em Capture (como mostrado abaixo).



3. Sua captura de pacotes deve começar e o Wireshark deverá mostrar a janela de captura de pacotes. Esta janela exibe um resumo breve do tipo de tráfego que está sendo capturado, assim como a duração da captura atual (como mostrado abaixo).



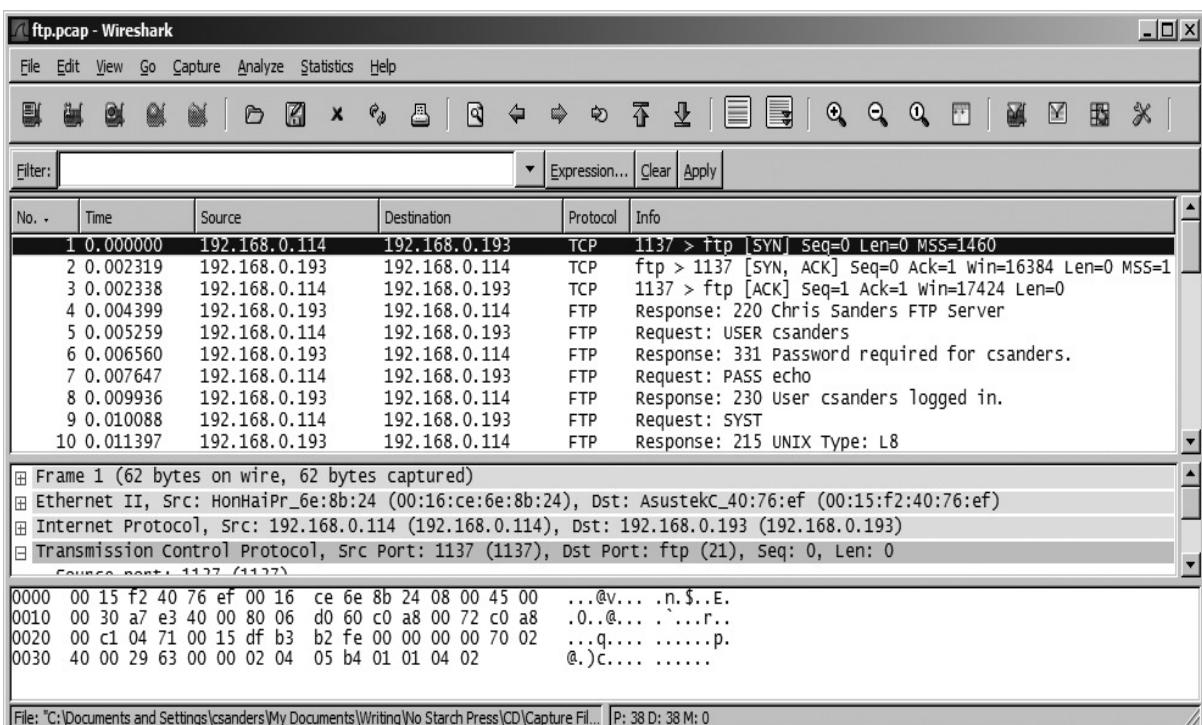
4. Espere cerca de mais ou menos um minuto, e quando você estiver pronto para parar a captura e visualizar os dados, clique em Parar.

Depois de ter concluído as etapas e terminado o processo de captura, a janela principal do Wireshark aparecerá com os dados. Por uma questão de fato, você poderá se assustar com quantidade de dados que aparece, mas tudo vai começar a fazer sentido mais rapidamente, quando quebrarmos (analisarmos) a janela principal do Wireshark uma peça de cada vez.

A Janela Principal

Você vai passar a maior parte de seu tempo usando a janela principal do Wireshark. Este é o lugar onde todos os pacotes que você capturou serão apresentados e divididos em um formato mais compreensível. Usando o pacote de captura que acabou de fazer, vamos dar uma olhada na janela principal do Wireshark (como mostrado abaixo), que contém três painéis.

Os três painéis na janela principal dependem um do outro. Dispostos em ordem para visualizarmos os detalhes de um pacote individualmente, devemos primeiro selecionar o pacote clicando nele no painel **Packet List**. Depois de selecionar o pacote, poderemos ver os bytes que correspondem exatamente ao pacote selecionado, no painel **Packet Bytes** quando clicarmos na parte desejada do pacote no painel **Packet Details**.



Painel Packet List

O painel superior, conhecido como **Packet List**, exibe uma tabela contendo todos os pacotes do arquivo de captura atual. Você vai ver colunas contendo o número do pacote, o tempo relativo quando o pacote foi capturado, a origem e o destino do pacote, o protocolo do pacote, e algumas informações gerais encontrados em no pacote.

Painel Packet Details

O painel central, conhecido como o **Packet Details**, contém uma exibição hierarquia de informações sobre um único pacote. Esta exposição pode ser recolhida e expandida para mostrar todas as informações coletadas sobre um pacote individualmente.

Painel Packet Bytes

O painel inferior, e talvez o mais confuso, é o Packet Bytes. Este painel apresenta o pacote em formato de bytes, não transformado da forma que ele é, ele mostra o pacote da forma que ele atravessa o fio. Esta é a informação em estado bruto, com nada quente ou frio para tornar mais fácil a sua compreensão.

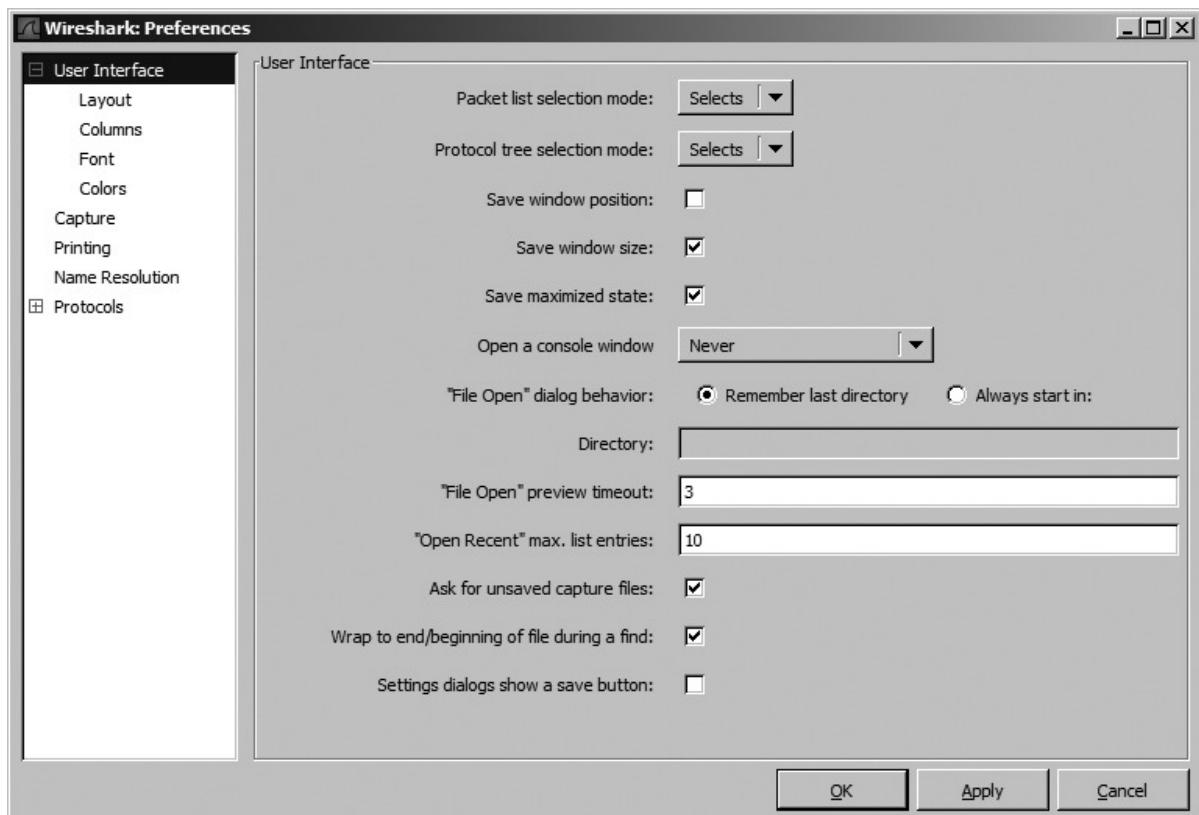
Nota:

É muito importante entender como funcionam esses painéis diferentes entre si, pois você estará gastando mais do seu tempo trabalhando com eles na janela principal.

A caixa de Diálogo Preferências

O Wireshark tem várias preferências que podem ser personalizados para atender às suas necessidades. Vejamos algumas das mais importantes.

Para acessar as preferências do Wireshark, selecione **Edit** do Menu drop-down principal e clique em **Preferences**. Isso deve abrir a caixa de diálogo **Preferences**, que contém várias opções personalizáveis (como mostrado abaixo).



Essas preferências são divididas em cinco secções principais: interface com o usuário, captura, a impressão de resolução de nomes e protocolos.

Interface do usuário

As preferências da interface do usuário determinam como o Wireshark apresenta os dados. Você pode alterar a maioria das opções aqui de acordo com suas preferências pessoais, incluindo ou não salvar as posições da janela, o layout dos três painéis principais, a colocação da barra de rolagem, a colocação da lista de pacotes, as colunas dos painéis, as fontes usadas para exibir os dados capturados, e o fundo e as cores de primeiro plano.

Captura

As preferências de captura permitem que você especifique opções relacionadas à forma como os pacotes são capturados, incluindo a sua interface de captura de padrão, ou não usar o modo promíscuo por padrão, e se deve ou não atualizar o painel **Packet List** em tempo real.

Imprimindo

A seção de preferências de impressão permite que você especifique várias opções relacionadas à maneira como o Wireshark imprimirá seus dados.

Resolução de Nomes

As preferências na seção de resolução de nomes permitem você ativar recursos do Wireshark que lhe permitem resolver endereços para nomes mais reconhecidos (Incluindo o MAC, rede e resolução de nomes de transportes) e especificar o número máximo de solicitações simultâneas de resolução de nome.

Protocolos

As preferências na seção de protocolos permitem que você manipule as opções relacionadas com a captura e exibição dos vários protocolos que o Wireshark é capaz de decodificar. Não tem preferências configuráveis para todos os protocolos, mas alguns têm várias opções que podem ser alteradas. Estas opções são deixadas inalteradas a menos que você tenha um motivo específico para fazê-las.

Código de Cores de um Pacote

Se você se parece comigo, você pode ter uma aversão a objetos brilhantes e cores bonitas. Se for esse o caso, a primeira coisa que você deve ter notado quando você abre o Wireshark, são as diferentes cores dos pacotes no painel **Packet List** (como mostrada abaixo). Pode parecer que as cores sejam distribuídas aleatoriamente para cada pacote individualmente, mas este não é o caso.

Nota:

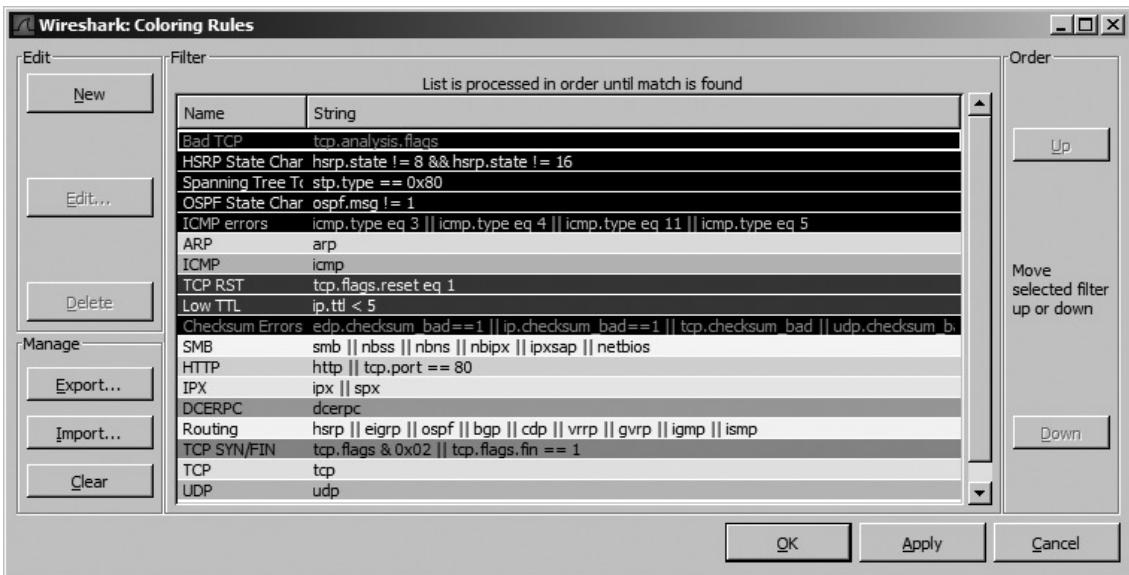
Quando me refiro ao tráfego, você pode supor que eu estou me referindo a todos os pacotes apresentados no painel **Packet List**. Mais especificamente, quando me refiro a ela, no contexto do tráfego DNS, eu estou falando de todos os pacotes do protocolo DNS no painel **Packet List**.

Cada pacote é exibido com uma determinada cor, por uma razão. Por exemplo, você pode perceber que todo o tráfego DNS é azul e todo o tráfego HTTP está verde. Estas cores refletem o protocolo do pacote. O código de cores permite que você rapidamente diferencie entre os vários protocolos de modo que você não precisa ler o campo protocolo no painel **Packet List** para cada pacote individualmente. Você irá achar que isso acelera muito o tempo que leva para percorrer grandes arquivos de captura.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|------------------|---------------------------------|
| 1 | 0.000000 | 10.100.17.47 | 10.100.16.15 | DCERPC Request: | call_id: 95 opnum: 69 ctx_id: 0 |
| 2 | 0.000361 | 10.100.16.15 | 10.100.17.47 | DCERPC Response: | call_id: 95 ctx_id: 0 |
| 3 | 0.001946 | 10.100.17.47 | 10.100.16.15 | DCERPC Request: | call_id: 96 opnum: 26 ctx_id: 0 |
| 4 | 0.002034 | 10.100.16.15 | 10.100.17.47 | DCERPC Response: | call_id: 96 ctx_id: 0 |

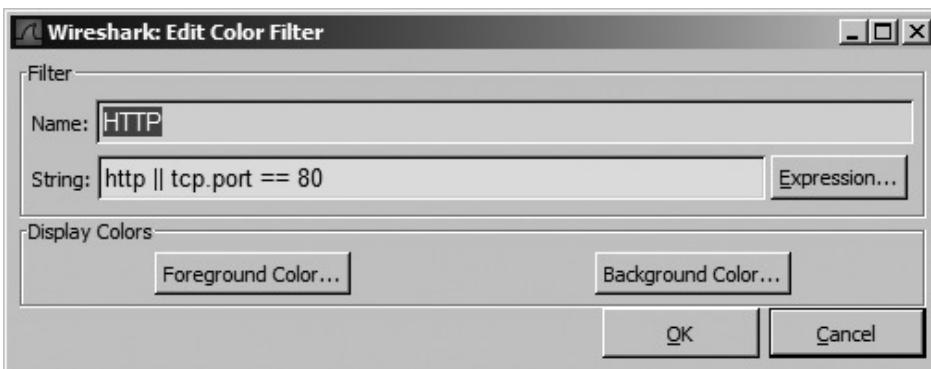
O Wireshark torna fácil de ver as cores que são atribuídas a cada protocolo através da janela **Coloring Rules**. Para abrir essa janela, siga estes passos:

1. Abrir o Wireshark.
2. Selecione a opção **View** no menu drop-down principal.
3. Clique em **Coloring Rules**. A janela **Coloring Rules** aparecerá (como mostrado abaixo), exibindo uma lista completa de todas as regras definidas no Wireshark. Você pode definir suas próprias regras de coloração e modificar as existentes.



Por exemplo, para alterar a cor usada como plano de fundo para o tráfego HTTP a partir do padrão verde lavanda, siga estes passos:

1. Abra o Wireshark e acesse a caixa de diálogo **Coloring Rules** (**View > Coloring Rules**).
2. Encontre a regra de coloração do HTTP na lista **Coloring Rules**, e selecione-a clicando uma vez.
3. Clique no botão **Edit**.
4. Clique no botão **Background Color** (Cor de fundo) como mostrado abaixo.



5. Escolha a cor que deseja usar na roda de cores e clique em **OK**.
6. Clique em **OK** mais duas vezes para aceitar as alterações e voltar à janela principal.
7. A janela principal deverá ser atualizada para refletir a nova cor.

4

TRABALHANDO COM PACOTES CAPTURADOS

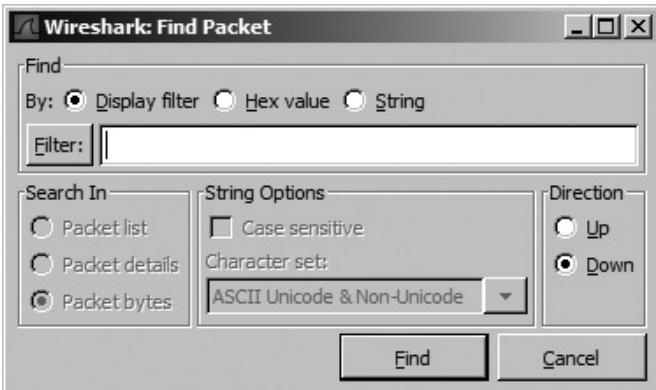
Agora que você já realizou a sua primeira captura de pacotes, vamos nos deter um pouco mais sobre alguns conceitos básicos que você precisa saber sobre o trabalho com os pacotes capturados no Wireshark. Isto inclui a verificação e marcação de pacotes, o salvamento dos arquivos de captura, anexando arquivos de captura, imprimindo os pacotes, e alterando tempo dos formatos exibidos.

Encontrando e Marcando Pacotes

Uma vez que você começa realmente a fazer a análise de pacotes, você acabará por encontrar cenários que envolvem um número muito grande de pacotes. Como o número desses pacotes cresce em milhares e mesmo milhões, você precisa ser capaz de navegar através destes pacotes de forma mais eficiente. Esta é a razão do uso do Wireshark, pois ele lhe permitirá encontrar e marcar os pacotes que correspondam a um determinado critério.

Encontrando Pacotes

Para encontrar os pacotes que correspondam a critérios específicos, abra o pacote de diálogo Localizar (Mostrado na abaixo) selecionando a opção **Edit** do Menu drop-down principal e clicando em **Find Packets** ou pressionando as teclas **CTRL-F**.



Esta caixa de diálogo oferece três opções para encontrar os pacotes: filtro de exibição, valor hexadecimal ou uma string. A opção de filtro de visualização permite que você insira uma expressão que será usada para encontrar somente os pacotes que a satisfaçam (que será detalhada mais tarde). A busca por string ou valor hexadecimal procurará por pacotes com a string ou o valor hexadecimal especificado, você pode ver exemplos de todos estes casos abaixo. Outras opções incluem a capacidade de selecionar a janela que você deseja pesquisar, o conjunto de caracteres a ser usado, e qual a direção que você deseja pesquisar.

Table 4-1: Examples of Various Search Types for Finding Packets

| Search Type | Example |
|----------------|--------------------------------------|
| Display filter | not ip, ip address==192.168.0.1, arp |
| Hex value | 00:ff, ff:ff, 00:AB:B1:f0 |
| String | Workstation1, UserB, domain |

Uma vez que você fez sua escolha, digite a seqüência de pesquisa na caixa de texto, e clique em **Find** para encontrar o primeiro pacote que atenda aos seus critérios. Para encontrar a próxima correspondência, pressione **CTRL-N**, ou para encontrar a correspondência anterior pressione **CTRL-B**.

Marcando Pacotes

Depois de ter encontrado os pacotes que combinam com seus critérios, você pode marcar os de interesse particular. Pacotes marcados destacam-se com um fundo preto e texto em branco, como mostrado abaixo. (Você também poderá ordenar os pacotes marcados quando for salvá-los). Existem várias razões para você querer marca de um pacote, incluindo a possibilidade de salvar os pacotes separadamente, ou ser capaz de encontrá-los rapidamente com base na sua cor.

| No. ▾ | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|--|
| 1 | 0.000000 | 10.100.17.47 | 10.100.16.15 | DCERPC | Request: call_id: 95 opnum: 69 ctx_id: 0 |
| 2 | 0.000361 | 10.100.16.15 | 10.100.17.47 | DCERPC | Response: call_id: 95 ctx_id: 0 |

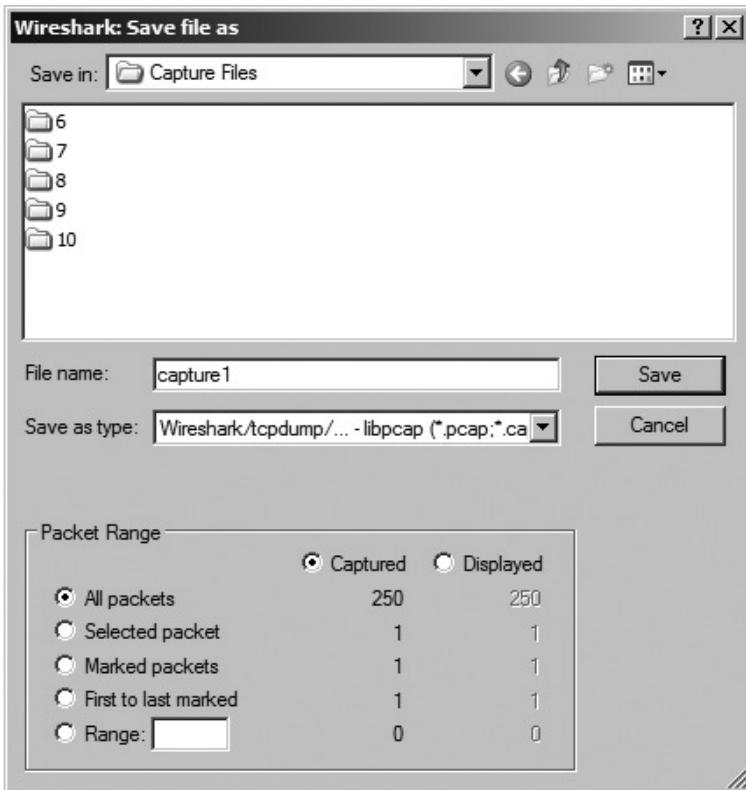
Para marcar um pacote, clique com o botão direito do mouse no painel **Packet List** e escolha **Mark Packet** do Menu pop-up. Ou, simplesmente clique em um pacote no painel **Packet List** e pressione **CTRL-M** para marcá-lo. Para desmarcar um pacote, desfazer esta definição usando **CTRL-M** novamente. Você pode marcar os pacotes que desejar em uma captura. Você pode saltar para frente e para trás entre os pacotes marcados pressionando **SHIFT-CTRL-N** e **SHIFT-CTRL-B**, respectivamente.

Salvando e Exportando Arquivos Capturados

Quando estiver executando a análise do pacote, você vai achar que uma boa parte da análise irá aparecer após a sua captura. Geralmente, você irá realizar várias capturas, e irá salvá-las para analisá-las todas de uma vez. Por isso, o Wireshark permite que você salve a captura de arquivos para serem analisados posteriormente.

Salvando Arquivos Capturados

Para salvar uma captura de pacotes, selecione **File** no menu drop-down e em seguida, clique em **Save As**, ou pressione **CTRL-SHIFT hifen**. Você deverá ver a caixa de dialogo **Save File** como mostrado abaixo. Aqui você será perguntado sobre o local para salvar a sua e captura de pacotes e o formato de arquivo que você deseja usar. Se você não especificar um formato de arquivo, o Wireshark irá usar o formato de arquivo padrão (.pcap).



Uma das características mais poderosas da caixa de diálogo **Save File** é a capacidade de salvar uma série de pacotes específicos. Você pode optar por salvar apenas pacotes em um intervalo específico, pacotes marcados, ou pacotes visíveis como o resultado de um filtro de uma seleção. Esta é uma ótima maneira de reduzir grandes arquivos de captura.

Exportando Dados Capturados

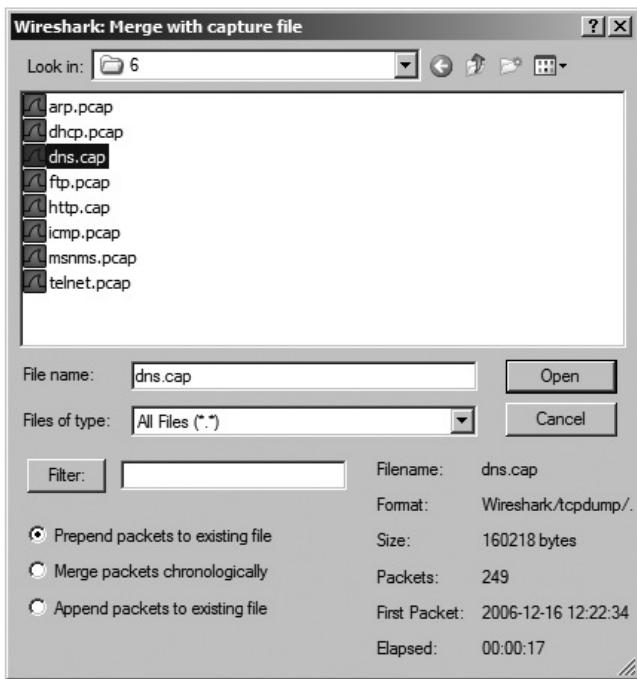
Você pode exportar seus dados capturados pelo Wireshark em diversos formatos para exibição em outras mídias ou importar para outra ferramenta de análise de pacotes. Esses formatos incluem texto simples, Postscript, valores separados por vírgula (CSV), e XML. Para exportar a captura de pacotes, escolha **File > Export..**, e depois selecione o formato que você deseja exportar. Você será perguntado por uma caixa de diálogo **Save As** sobre as opções relacionadas a esse formato específico.

Mesclando Arquivos Capturados

Certos tipos de análise requerem a capacidade de mesclar vários arquivos de captura, e, felizmente o Wireshark fornece dois métodos diferentes para fazer isso.

Para mesclar um arquivo de captura, siga estes passos:

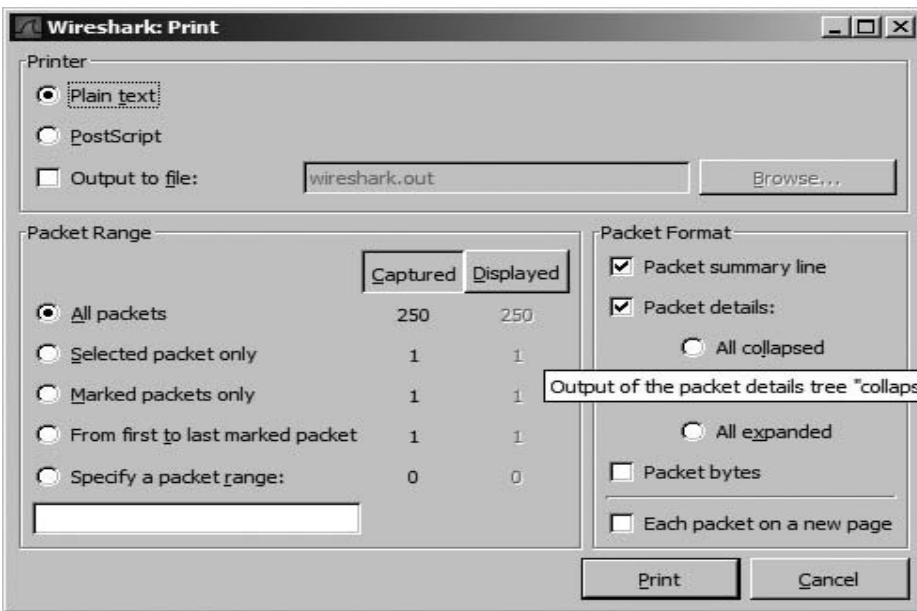
1. Abra um dos arquivos capturados que você deseja mesclar.
2. Escolha **File>Merge** para fazer a mesclagem com a caixa de dialogo Capture File como mostrado abaixo.
3. Selecione o novo arquivo que você deseja mesclar ao arquivo já aberto, e em seguida, selecione o método a utilizar para a mesclagem dos arquivos. Você pode inserir o arquivo selecionado no inicio do aberto, anexá-lo, ou mesclá-lo em ordem cronológica com base na sua data e hora.



Alternativamente, se você desejar mesclar vários arquivos rapidamente em ordem cronológica, considere o uso de arrastar e soltar. Para fazer isso, abra o arquivo com a primeira captura através do Windows Explorer (ou qualquer que seja seu navegador de arquivos preferido). Em seguida vá para o segundo arquivo, clique nele e arraste-o para a janela principal do Wireshark.

Imprimindo Pacotes

Embora a maioria das análises terá lugar na tela do computador, você ainda poderá encontrar a necessidade de imprimir os dados capturados. Para imprimir pacotes capturados, abra a caixa de diálogo **Print** escolhendo **File > Print** a partir do menu principal como mostrado abaixo.



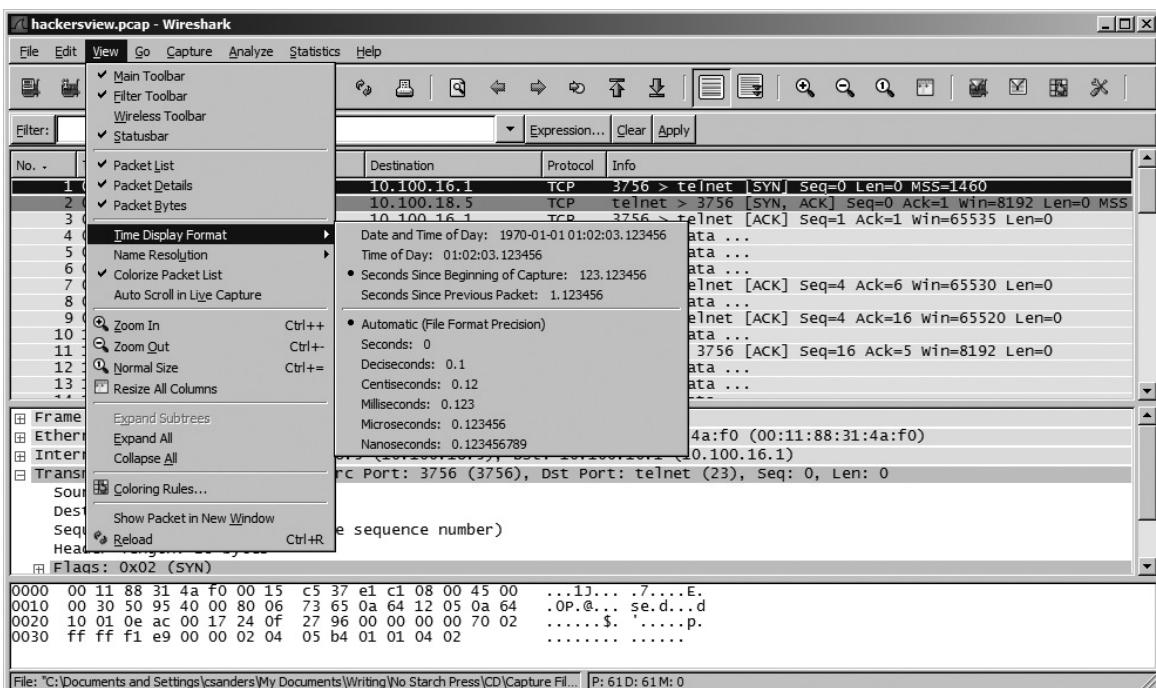
Você pode imprimir os dados selecionados como texto simples, PostScript ou para uma saída de arquivo. Com a caixa de diálogo **Save File As**, você pode especificar uma determinada faixa de pacotes para impressão, apenas os pacotes marcados, ou os pacotes apresentados como o resultado de um filtro. Você também pode escolher qual dos três painéis principais do Wireshark será impresso cada pacote. Depois de ter selecionado as opções desejadas, simplesmente clique em **Print**.

Formatos de Exibição de Tempo e Referências

O tempo é essencial, especialmente na análise de pacotes. Tudo o que acontece em uma rede é sensível ao tempo, e você terá de analisar as tendências e a latência da rede em praticamente todos os arquivos de captura. O Wireshark reconhece a importância do tempo e nos fornece várias opções configuráveis que lhe digam respeito. Aqui vamos dar uma olhada em formatos de tempo de exibição e referências.

Formato de Exibição de Tempo

Cada pacote que é capturado pelo Wireshark é dado um rótulo de tempo, que é aplicada ao pacote pelo sistema operacional. O Wireshark pode mostrar a hora absoluta, e a relação entre tempo em que o último pacote foi capturado e o início e final da captura. As opções relacionadas com a exibição do tempo encontram-se através da visualização do menu principal. O **Time Display Format** (como mostrado abaixo) permite-lhe configurar o formato de apresentação, assim como a precisão do tempo mostrado. A opção de formato de apresentação permite-lhe escolher várias opções para exibição da hora. As opções de precisão permitem que você defina a precisão de exibir o tempo automaticamente ou configurar manualmente, como segundos, milissegundos, microssegundos, e assim por diante. Vamos alterar essas opções, muitas vezes no final do livro, assim você deve se familiarizar com elas agora.



Referências de Tempo de um Pacote

A referência de tempo de um pacote permite que você o configure para que todos os cálculos de tempo subsequentes sejam feitos em relação a esse pacote específico. Esta característica é particularmente útil quando você está examinando várias solicitações de dados em um arquivo de captura e quer ver a referência de tempo de um pacote solicitado. Para definir uma referência de tempo para um determinado pacote, escolha o pacote de referência no painel **Packet List**, em seguida escolha **Edit > Set Time Reference** do menu principal. Ou, selecione o pacote de referência e pressione **CTRL-T**. Para remover uma referência de tempo a partir de um determinado pacote, selecione o pacote e pressione novamente **CTRL-T**. Quando você habilitar uma referência de tempo em um determinado pacote, na coluna de tempo no painel **Packet List** será exibido *REF como mostrado abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-------------|-------------|----------|---|
| 3 | 0.001263 | 10.100.18.5 | 10.100.16.1 | TCP | 3756 > telnet [ACK] Seq=1 Ack=1 win=65535 Len=0 |
| 4 | *REF* | 10.100.16.1 | 10.100.18.5 | TELNET | Telnet Data ... |
| 5 | 0.000058 | 10.100.18.5 | 10.100.16.1 | TELNET | Telnet Data ... |
| 6 | 0.001018 | 10.100.16.1 | 10.100.18.5 | TELNET | Telnet Data ... |

Nota: Definir uma referência de tempo de um pacote só é útil quando o formato de exibição de tempo de uma captura é configurado para exibir o tempo em relação ao início da captura. Qualquer outra configuração não irá produzir resultados úteis e irá criar um conjunto de referências, muitas vezes confusa.

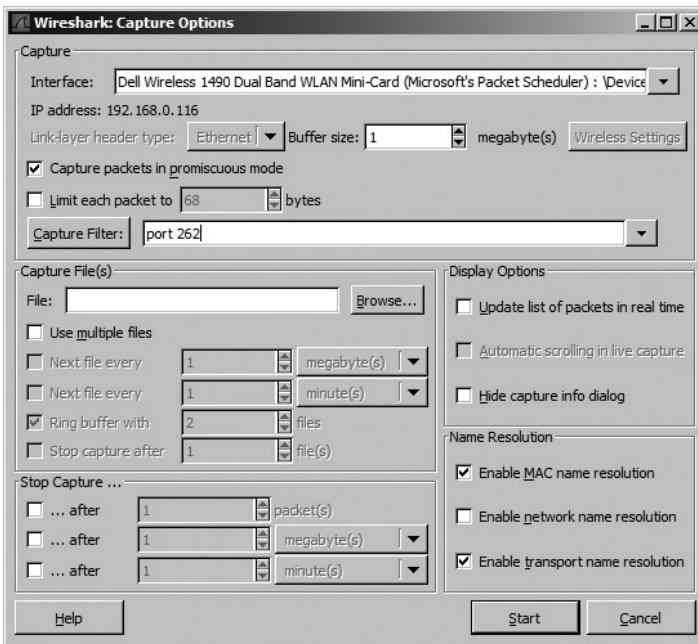
Filtros de Captura e Exibição

Anteriormente discutimos sobre o salvamento de pacotes através do uso de filtros. Os filtros permitem-nos mostrar apenas determinados pacotes em uma captura. Nós podemos criar e usar uma expressão para encontrar exatamente aquilo que queremos, mesmo em um enorme arquivo de captura. Uma expressão não é mais que uma sequência de texto que diz ao Wireshark o que mostrar e o que não mostrar. O Wireshark oferece dois tipos de filtros: os filtros de captura e filtros de exibição.

Filtros de Captura

Filtros de captura são utilizados durante o processo de captura de pacotes em tempo real e são aplicados pelo WinPcap. O conhecimento de sua sintaxe pode ser também útil em outros programas de análise de rede. Você pode configurá-los na caixa de diálogo **Capture Option** onde você pode especificar o tráfego que você quer ou não quer capturar. Uma boa maneira de usar um filtro de captura seria quando queremos capturar o tráfego em um servidor com múltiplas funções. Por exemplo, suponha que você está solucionando um problema com um serviço sendo executado na porta 262. Se o servidor que você está analisando está executando vários serviços diferentes em várias portas, então localizar e analisar apenas o tráfego na porta 262 pode ser completamente trabalhoso. Para capturar somente o tráfego da porta 262, você pode usar um filtro de captura. Basta seguir estes passos:

1. Abra a caixa de diálogo **Capture Options** (como mostrado abaixo), selecione a interface que deseja capturar os pacotes e escolha um filtro de captura.
2. Você pode aplicar o filtro de captura, digitando uma expressão ao lado do botão **Capture Filter** ou clicando no botão **Capture Filter**, que irá iniciar o construtor de expressão de captura que irá ajudar você a criar o seu filtro. Queremos que nosso filtro mostre apenas o tráfego de entrada e saída da porta 262, assim digite apenas **port 262**, como mostrado abaixo.
3. Depois de definir seu filtro, clique em **Start** para começar a captura. Após a coleta de uma amostra adequada, você deve ver agora apenas o tráfego da porta 262 e ser capaz de analisar de forma mais eficiente esses dados em particular.



Filtros de Exibição

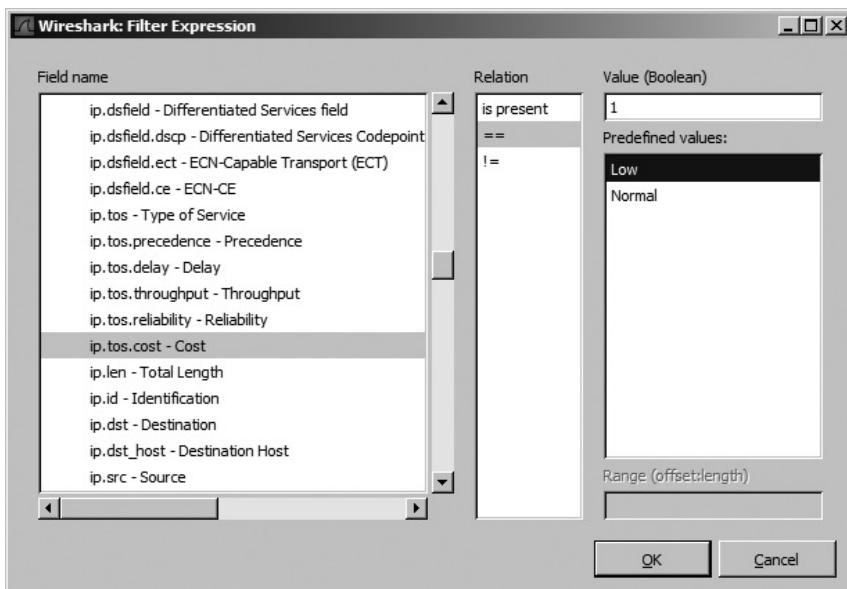
Pode utilizar um filtro para exibir dados em um arquivo de captura, uma vez que o mesmo foi criado, e desta forma mostrar somente os pacotes que correspondam a esse filtro. Você pode inserir um filtro que foi criado na caixa de texto acima do painel **Packet List**. A utilização de filtros é mais comumente usada que a apresentação de toda a captura. Dessa forma, se você precisar voltar para a captura original, você pode simplesmente desmarcar a expressão de filtro. Você pode usar um filtro para limpar o tráfego de broadcast a partir de um arquivo de captura, por exemplo, para limpar broadcasts de ARPs a partir do painel **Packet List** quando estes pacotes não estão relacionados com o problema atual que está sendo analisado. No entanto, devido aos pacotes de broadcasts ARPs poderem ser úteis mais tarde, é melhor filtrá-los temporariamente do que eliminá-los completamente. Para filtrar todos os pacotes ARPs na janela de captura, siga estes passos:

- Navegue até a parte superior do painel **Packet List** e coloque o cursor na caixa de texto **Filter**.
- Digite **!arp** e pressione **ENTER** para remover todos os pacotes ARP do painel **Packet List** como mostrado abaixo. Para remover o filtro, desmarque a caixa de texto e pressione **ENTER** novamente.



O Construtor de Filtros (uma forma simples)

A caixa de diálogo de construção de filtros é um recurso que torna mais fácil a utilização do Wireshark pelos usuários novatos, para capturar e criar filtros de exibição. Para acessar esta caixa de diálogo, clique no botão **Capture Filter** na caixa de diálogo **Capture Options** e, em seguida, clique no botão **Expression**.



A primeira coisa que você vai notar na caixa diálogo **Filter Expression** é uma lista de todos os campos possíveis relacionadas ao protocolo no lado esquerdo da janela. Esses campos especificam todos os critérios de filtro possíveis. Para criar um filtro, siga estes passos:

- Para ver os campos específicos dos critérios associados a um referido protocolo, expanda-o clicando no símbolo mais (+) próximo a ele. Uma vez que você encontrar os critérios que você deseja basear o seu filtro, selecione-o clicando sobre ele.
- Selecione a relação que o campo selecionado terá com os valores dos critérios fornecidos por você. Esta relação é especificada em termos de igual, maior que, menor que, e assim por diante.
- Crie a sua expressão de filtro, especificando o valor dos critérios que dizem respeito ao o campo selecionado. Você pode definir esse valor ou selecione valores pré-programados do Wireshark.
- Depois de ter feito isso, clique em **OK** para ver a versão completa do texto somente do filtro que você acabou de criar.

A Sintaxe de Construção de Filtros

O caixa de diálogo **Filter Expression** é ótimo para os usuários novatos, mas quando você começar a dominar a criação de filtros, você vai achar que digitar manualmente as expressões de filtro aumentará mais ainda a sua eficiência.

A exibição da **Filter expression structure** é muito simples, mas é extremamente poderosa. Esta linguagem é específica do Wireshark. Vejamos como a sintaxe desse filtro funciona e alguns exemplos do que podemos fazer com ele.

Filtrando um Protocolo Específico

Você vai usar na maioria das vezes a captura ou exibição de uma captura filtrada com base em um protocolo específico. Por exemplo, digamos que você está solucionando um problema de **TCP** e você quer ver apenas o tráfego **TCP** em um arquivo de captura. Se assim for, basta usar um filtro de **tcp** quando quiser começar o trabalho.

Agora vamos olhar para as coisas do outro lado da cerca. Imagine que no curso de solucionar seu problema **TCP**, você tem usado bastante o **ping**, gerando uma grande quantidade de tráfego **ICMP**. Você pode remover este tráfego **ICMP** de seu arquivo de captura com a expressão de filtro **!icmp**.

Sinais de Comparação

Os sinais de comparação permitem comparar valores. Por exemplo, quando você está resolvendo problemas em redes TCP/IP, muitas vezes você vai precisar ver todos os pacotes de um endereço IP em particular. Em um caso como este o sinal de comparação de igual (==) permitirá você criar um filtro que mostre todos os pacotes com um endereço IP 192.168.0.1 usando uma expressão de filtro como ip.addr == 192.168.0.1. Ou, considere o exemplo mais avançado de um sinal de comparação. Imagine um cenário em que só precisamos de visualizar os pacotes de menos de 128 bytes de comprimento. Nós podemos usar o sinal menor ou igual a (<=) para realizar este objetivo em uma expressão de filtro como <frame.pkt_len = 128.

Você vai encontrar uma lista completa de sinais de comparação no Wireshark baixo:

Table 4-2: Wireshark Filter Expression Comparison Operators

| Operator | Description |
|----------|--------------------------|
| == | Equal to |
| != | Not equal to |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal to |
| <= | Less than or equal to |

Operadores Lógicos

Os operadores lógicos permitem combinar várias expressões de filtro em um única instrução. Você pode usar operadores lógicos para aumentar dramaticamente a eficácia de seus filtros.

Por exemplo, considere o nosso exemplo anterior de exibir apenas pacotes de um determinado endereço IP e, agora, assumir que estamos interessados em dois endereços IP. Podemos usar o operador **ou** (or) para criar uma expressão que mostre pacotes contendo um endereço IP. A sintaxe da expressão seria ip.addr ser == 192.168.0.1 or 192.168.0.2 == ip.addr. Você vai encontrar uma lista completa dos operadores lógicos do Wireshark na tabela baixo:

Table 4-3: Wireshark Filter Expression Logical Operators

| Operator | Description |
|----------|---|
| and | Both conditions must be true |
| or | Either one of the conditions must be true |
| xor | One and only one condition must be true |
| not | Neither one of the conditions is true |

Exemplos de Expressões de Filtro

Embora os conceitos relacionados à criação de expressões de filtro sejam bastante simples, você vai precisar fazer referência a diversas palavras-chave e operadores específicos ao criar novos filtros para resolução dos vários problemas. Porque este livro não pretende ser um manual do usuário do Wireshark, não vamos cobrir todas as palavras-chave e operadores, mas você vai encontrar informações sobre eles no site do Wireshark. A tabela abaixo dá uma idéia de alguns exemplos de expressões de filtro.

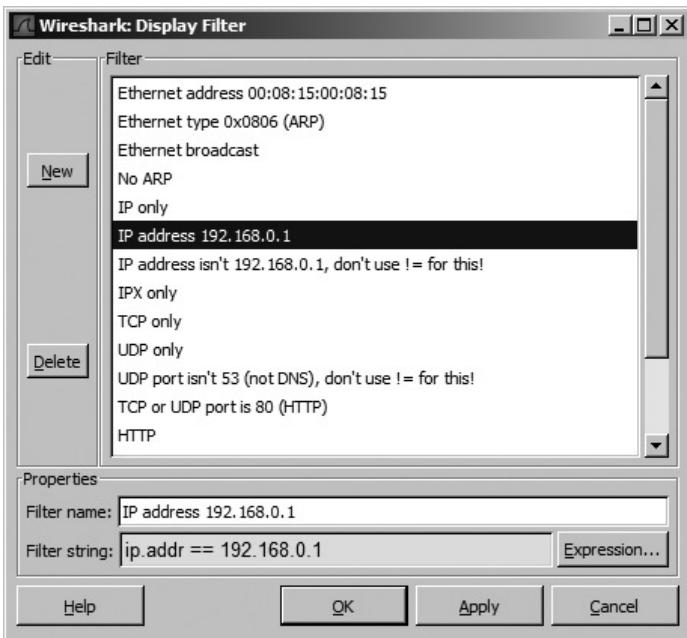
Table 4-4: Sample Capture and Display Filter Expressions

| Expression | Description |
|--|--|
| host www.example.com | Displays all traffic from the host www.example.com |
| host www.example.com and not (port 80) | Displays all non-web traffic from the host www.example.com |
| !dns | Shows everything except DNS traffic |
| not broadcast and not multicast | Only shows unicast traffic |
| ip.dst==192.168.0.1 | Shows all traffic destined for 192.168.0.1 |

Salvando Filtros

Uma vez que você começou a criar lotes de captura e filtros de exibição, você certamente usará determinados filtros com mais freqüência. Felizmente, você não precisará digitá-los toda vez que quiser usá-los; O Wireshark permite que você salve seus filtros para uso posterior. Para salvar o filtro personalizado, siga estes passos:

1. Selecione **Capture > Capture Filters** para abrir a caixa de diálogo **Display Filter**.
2. Crie um novo filtro clicando no botão **New** no lado esquerdo da tela.
3. Digite um nome para o filtro na caixa ao lado das palavras **Filter name**.
4. Digite o filtro atual na caixa ao lado das palavras **Filter string**.
5. Assim que tiver terminado, clique no botão **Save** para salvar o seu filtro na lista.



O Wireshark também inclui vários filtros construídos, mas estes são apenas para dar um exemplo de como um filtro pode se parecer. Você vai querer usá-los quando você estiver criando seus próprios filtros, porque eles são ótimos para fins de referência.

5

CARACTERISTICAS AVANÇADAS DO WIRESHARK

Depois de dominar os conceitos fundamentais do Wireshark, você provavelmente vai querer mergulhar ainda mais em algumas de suas mais avançadas características. Neste capítulo, vamos olhar para alguns desses recursos poderosos, incluindo a resolução de nomes, dissecação de protocolo e remontagem de pacotes.

Resolução de Nomes

Os dados em rede são transportados através de vários sistemas de endereçamento alfanuméricos que muitas vezes são demasiado longos ou complicados de se lembrar, como o endereço físico de hardware 00:16: CE: 6E: 8B: 24. A resolução de nomes (também chamado de pesquisa de nome) é o processo que utiliza um protocolo para converter um endereço em outro. Por exemplo, quando um computador tem um endereço físico 00:16: CE: 6E: 8B: 24, o DNS e protocolos ARP nos permitem ver o seu nome como Marketing-2. Através dessa associação poderemos lembrar com facilidade esse novo endereço.

Podemos usar várias ferramentas de resolução de nomes para tornar a nossa captura de pacotes mais legível. Por exemplo, nós poderemos usar a resolução de DNS para ajudar a identificar facilmente o nome de um computador que estamos tentando identificar como a origem de um pacote específico.

Tipos de Resolução de Nomes do Wireshark

Existem três tipos de resolução de nomes disponíveis no Wireshark: **resolução de endereços MAC**, **resolução de nomes de rede** e **resolução de nomes de transportes (portas)**.

Resolução de Endereço MAC

A resolução de endereço MAC utiliza o protocolo ARP para tentar converter endereços MAC da camada 2 como **00:09:05 B1:02:03**, em endereços IP de camada 3, tais como **10.100.12.1**. Se essas tentativas de conversões falharem, como último recurso o Wireshark converterá os três primeiros bytes do endereço MAC na especificação IEEE com o nome do fabricante do dispositivo, como **Netgear_01: 02:03**.

Resolução do Nome da Rede

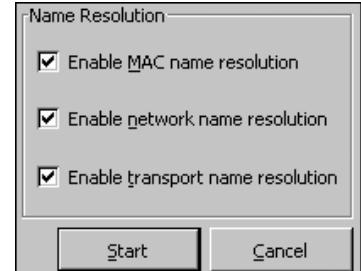
A resolução de nomes de Rede tenta converter um endereço da Camada 3, como o IP endereço **192.168.1.50**, em um nome DNS fácil de leitura e de memorização como **MarketingPC1**.

Resolução de Nome de Transporte

A resolução de nomes de Transporte (portas) tenta converter um número de porta a um nome do protocolo associado a ela. Um exemplo disto seria mostrar a **porta 80** como **http**.

Habilitando a Resolução de Nome

Para habilitar a resolução de nomes, abra a caixa de diálogo **Capture Options** (como mostrado na figura ao lado), escolhendo **Capture > Options** ou pressionando **CTRL-K**.



Desvantagens da Resolução de Nomes

Tendo em conta os seus benefícios, utilizar a resolução de nomes pode nos trazer algumas desvantagens como as seguintes:

- **Às vezes a resolução de nomes falha.** Isso simplesmente pode acontecer porque o nome é desconhecido pelo servidor de consulta.
- **A resolução de Nomes deve ocorrer toda vez que você abrir um arquivo de captura específico.** Porque essa informação não é salva no arquivo. Isto significa que se os servidores que a resolução de nome depende não estão disponíveis a resolução do nome irá falhar.
- O DNS pode adicionar pacotes ao arquivo de captura, silenciosamente e sem aviso. **O tráfego resultante de resolver todos os endereços baseados em DNS tornará o seu arquivo de captura mais volumoso.**
- **A resolução de Nomes exige uma sobrecarga de processamento adicional.** Se você está lidando com uma captura muito grande de arquivos e com pouca memória, você poderá querer renunciar ao recurso de resolução de nomes, a fim de conseguir utilizar mais os recursos do sistema.

Dissecção de Protocolo

Um dissecador de protocolo permite ao Wireshark quebrar um protocolo (ICMP, por exemplo) em várias seções para que ele possa ser analisado. O dissecador do protocolo ICMP permite ao Wireshark pegar os dados brutos e formatá-lo em um pacote ICMP. Você pode pensar em um dissecador como um tradutor entre os dados brutos que fluí através dos fios e o programa Wireshark. Para que um protocolo seja suportado pelo Wireshark, ele deve ter um dissecador incorporado.

O Wireshark utiliza vários dissecadores em conjunto para interpretar cada pacote. Ele determina que dissecador usar, usando sua lógica programada fazendo dessa forma uma busca bem sucedida.

Infelizmente, o Wireshark nem sempre faz as escolhas certas quando seleciona o dissecador correto para uso em um pacote. Isto é especialmente verdadeiro quando ele está usando um protocolo sobre a rede em uma configuração não padrão, como uma porta não-padrão. Felizmente, nós podemos mudar a maneira como Wireshark implementa certos dissecadores.

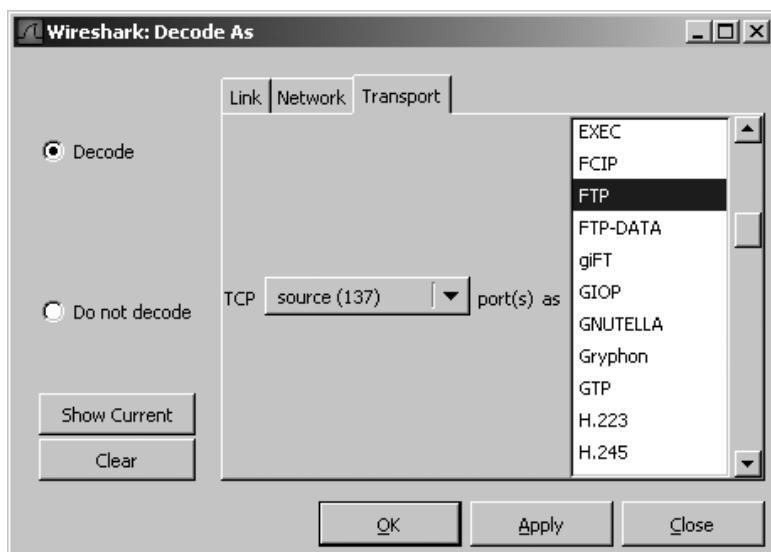
Por exemplo, abra o arquivo de rastreamento wrongdissector.dmp. Observe que esse arquivo contém um monte de comunicação NetBIOS entre dois computadores. No entanto, há algo errado aqui. Se você clicar em alguns dos pacotes, você vai notar que alguns dados no painel de Packet Bytes definitivamente não se parecem com o tráfego NetBIOS. Na verdade, se você olhar para pacotes 6 e 7, pode realmente ver um username e uma senha que está sendo enviado de um computador para o outro.

Após uma pequena investigação, descobrimos que os computadores que estamos analisando estão se comunicando via FTP (note as palavras FTP Server do lado direito da figura abaixo). O Wireshark pensa que este é um tráfego NetBIOS porque o servidor e o cliente estão configurados para usar o FTP na porta 137, a porta padrão de comunicação NetBIOS.

| | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|
| 0000 | 08 | 00 | 46 | 15 | 4c | c0 | 00 | 20 | 78 | e1 | 5a | 80 | 08 | 00 | 45 | 00 | ..F.L.. | x.Z...E. |
| 0010 | 00 | 63 | 6a | eb | 40 | 00 | 80 | 06 | e1 | c7 | cf | 89 | 07 | 67 | cf | 89 | .cj.Q... |g.. |
| 0020 | 07 | 68 | 00 | 89 | 05 | 04 | d6 | 64 | 6e | c9 | ba | 6c | d9 | 1b | 50 | 18 | .h.....d | n..l..P. |
| 0030 | 44 | 2b | d3 | 1b | 00 | 00 | 32 | 31 | 35 | 20 | 4d | 53 | 44 | 4f | 53 | 20 | D+....21 | 5 MSDOS |
| 0040 | 41 | 20 | 4e | 20 | 28 | 46 | 54 | 50 | 53 | 65 | 72 | 76 | 65 | 72 | 20 | 56 | A N (FTP | Server V |
| 0050 | 33 | 2e | 35 | 20 | 62 | 79 | 20 | 42 | 69 | 73 | 6f | 6e | 57 | 61 | 72 | 65 | 3.5 by B | isonware |
| 0060 | 20 | 49 | 6e | 74 | 65 | 72 | 6e | 61 | 74 | 69 | 6f | 6e | 61 | 6c | 29 | 0d | Interna | tional). |
| 0070 | 0a | 00 | 00 | 00 | 00 | 00 | 00 | 00 | | | | | | | | | | |

Para corrigir esse problema, temos que forçar o Wireshark a usar o disseccador do protocolo FTP nestes pacotes, um processo conhecido como decodificação forçada. Para executar este processo, siga estes passos:

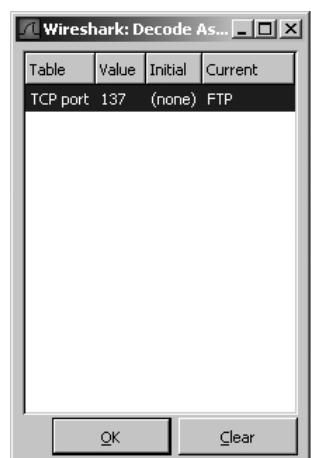
1. Clique com o botão direito do mouse em um dos pacotes e selecione **Decode As**. Isso abrirá uma janela na qual você poderá selecionar o disseccador desejado a ser usado (figura abaixo).
2. Diga ao Wireshark para decodificar todo o tráfego TCP originado na porta 137 usando o disseccador de protocolo FTP, selecionando Source (137) a partir do menu drop-down e selecionando em seguida FTP na guia **Transport**.
3. Depois de ter feito suas seleções, clique em **OK** para ver as alterações imediatamente aplicadas ao processo de captura. Você deverá ver os dados decodificados de modo que você possa analisá-los do painel **PacketList** sem ter que dissecá-los profundamente através de seus bytes individualmente.



Nota:

As mudanças que você faz quando criou uma decodificação forçada não são salvas junto com o arquivo de captura. Você deve recriar sua decodificação forçada toda vez que você abrir o arquivo de captura.

Você pode usar esta funcionalidade múltipla vezes no mesmo arquivo de captura. Porque pode ser difícil manter uma decodificação forçada quando você for usar mais de um arquivo de captura, o Wireshark pode fazer isso para você. A partir da caixa de diálogo **Decode**, você pode clicar no botão **Show Current** e mostrar todas as decodificações forçadas criadas até o momento. Você pode também limpá-las clicando no botão **Clear** (figura ao lado).



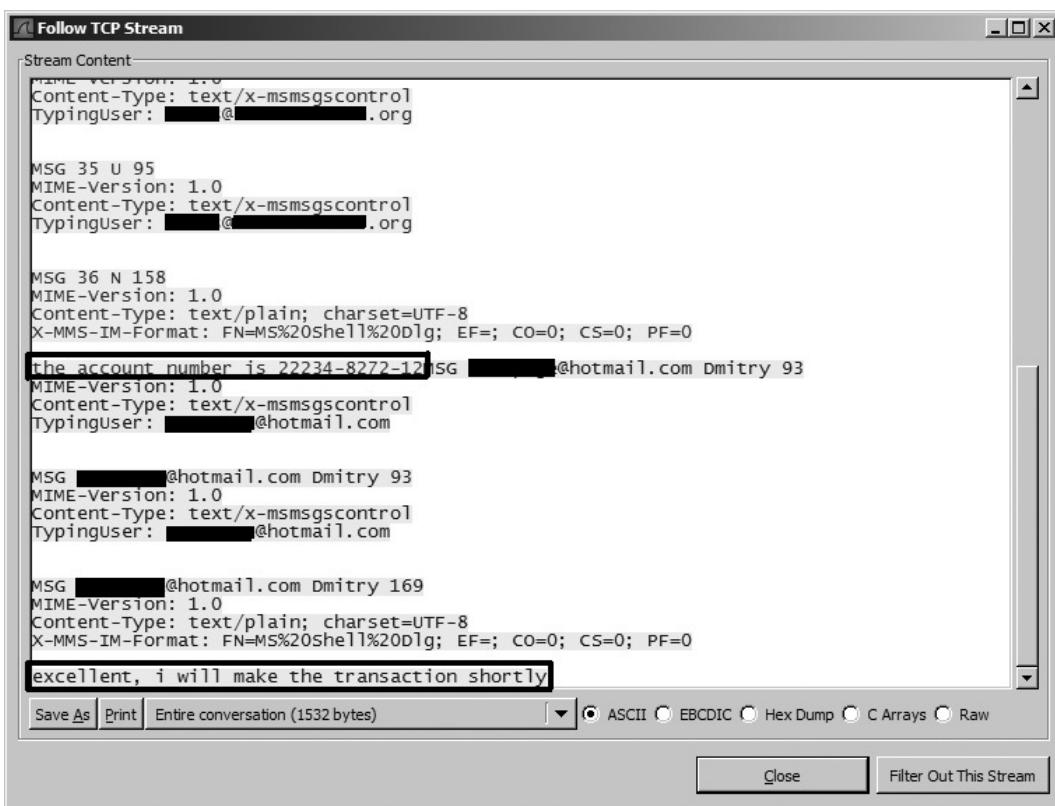
Seguindo os fluxos TCP

Uma das características de análise mais útil do Wireshark é sua capacidade para visualizar os fluxos do TCP na camada de aplicação. Este recurso permite que você combine todas as informações relacionadas aos pacotes e visualizar os dados das aplicações que os pacotes carregam como os usuários finais os vê. Mais que a visualização dos dados sendo enviado do cliente para o servidor em um grupo de pequenos pedaços, o fluxo do TCP ordena os dados para torná-los facilmente visíveis.

Você pode usar esta ferramenta na tentativa de capturar e decifrar mensagens instantâneas enviadas por um funcionário que é suspeito de oferecer informações corporativas. Para ver como isso iria funcionar, abra o arquivo de exemplo suspectemployechat.dmp. Neste arquivo você verá uma grande quantidade de tráfego gerado pelo cliente IM popular MSN Messenger. (Você pode identificar este como o tráfego MSN Messenger pela MSNMS que aparece no campo protocolo no painel Packet List).

Se você examinar os detalhes de cada pacote, você poderá ver pequenos pedaços de texto sendo transmitidos. Poderíamos passar muito tempo escrevendo as informações de cada pacote e ir combinando-as até descobrir o que está sendo dito no chat, mas isso não é muito prático. Em vez disso, vamos usar a Janela **TCP Stream** para obter uma melhor imagem do que está acontecendo.

Para seguir o fluxo TCP de dados, clique com o botão direito e selecione **Follow TCP Stream**. Fazendo isso no arquivo de captura de exemplo obtemos alguns resultados positivo. A janela **TCP Stream** agora mostra o bate-papo completo entre o nosso funcionário suspeito e a pessoa que está se comunicando com ele (figura abaixo).



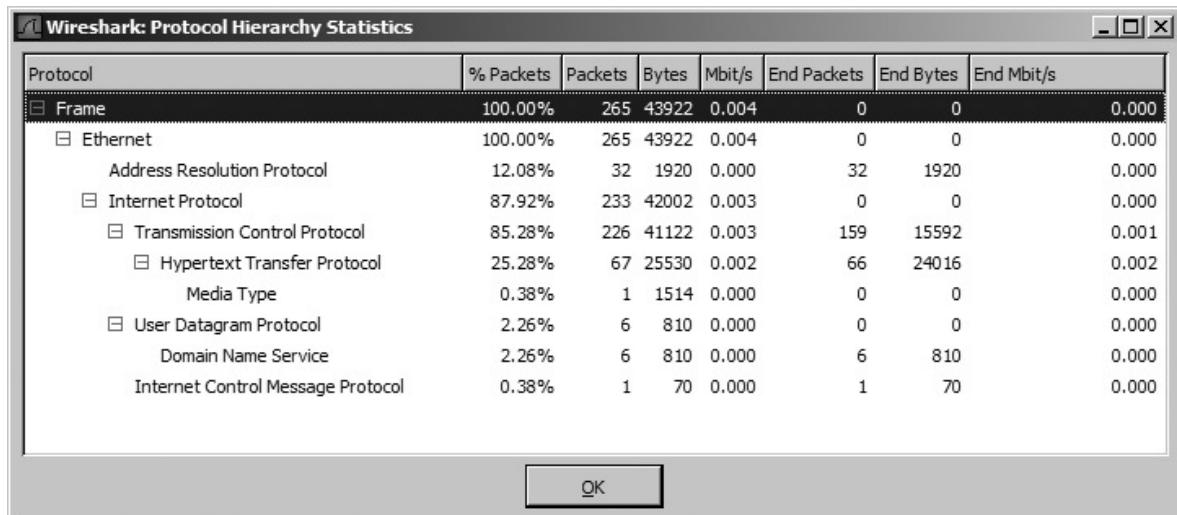
Além de visualizar os dados nesta janela, você também poderá salvá-lo como um arquivo de texto, imprimi-lo, ou optar por exibir os dados em ASCII, EBCDIC, Hex, matrizes, ou formato de dados original.

Janela Estatística e Hierárquica de Protocolo

Ao lidar com arquivos de captura extremamente grande, às vezes precisamos determinar a distribuição dos protocolos no arquivo, ou seja, que percentual da captura é TCP, que percentual é de IP, que percentual é DHCP, e assim por diante. Ao invés de contar cada pacote, totalizando os resultados, podemos usar a janela **Protocol Hierarchy Statistics** do Wireshark. Esta é uma ótima maneira de confrontar o resultado de sua rede. Por exemplo, se você sabe que 10 por cento do tráfego de sua rede é geralmente composto de tráfego ARP, e um dia você ter obtém uma captura que chega a 50 por cento, então você saberá que algo de errado está acontecendo.

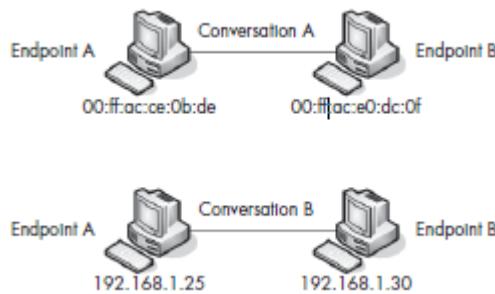
Abra a janela de **Protocol Hierarchy Statistics** (como mostrado abaixo) escolhendo **Statistics > Protocol Hierarchy**.

Observe que nem todos os totais somam exatamente 100 por cento. Porque um monte de pacotes que você vai ver contém vários protocolos de várias camadas, a contagem de cada protocolo em relação a cada pacote pode ser desligada. No entanto, você ainda vai ter uma imagem precisa da distribuição de protocolos nos arquivo de captura.



Visualizando os Dispositivos Finais

Um dispositivo final é o local onde termina a comunicação em um determinado protocolo. Por exemplo, existem dois pontos de comunicação TCP / IP: os endereços IP dos sistemas de envio e recepção de dados, 192.168.1.5 e 192.168.0.8. Um exemplo da Camada 2 seria a comunicação entre duas placas de rede (NIC) e seus endereços MAC. As placas de envio e recepção de dados tem os endereços 01:00:5e:00:00:16 e 01:00:5e:01:01:06, tornando esses endereços os dispositivos finais da comunicação. Você pode ver uma representação gráfica desse conceito na figura abaixo.



Ao analisar o tráfego, você pode descobrir que pode reduzir um problema em um determinado dispositivo final em sua rede. A janela **Endpoints** do Wireshark (**Statistics > Endpoints**) mostra várias estatísticas úteis para cada dispositivo final (figura abaixo), incluindo os endereços, bem como o número de pacotes e bytes transmitidos e recebidos por cada um. A guia na janela superior mostra os dispositivos finais suportados e reconhecidos no arquivo de captura. Clique na guia para visualizar a lista dos dispositivos finais e os respectivos protocolos. Selecione na próxima caixa a opção **Name Resolution** para visualizar a resolução de nomes na comunicação entre os dispositivos finais.

Endpoints: slowrouter.dmp

| Ethernet: 3 | Fibre Channel | FDDI | IPv4: 5 | IPX | JXTA | NCP | TCP: 26 | Token Ring | UDP: 3 | WLAN | RSVP |
|--------------------|---------------|-------|------------|----------|------------|----------|---------|------------|--------|------|------|
| Ethernet Endpoints | | | | | | | | | | | |
| Address | packets | Bytes | Tx Packets | Tx Bytes | Rx Packets | Rx Bytes | | | | | |
| AmbitMic_aa:af:80 | 233 | 42002 | 140 | 25628 | 93 | 16374 | | | | | |
| Intel_bc:c0:b9 | 265 | 43922 | 125 | 18294 | 140 | 25628 | | | | | |
| Broadcast | 32 | 1920 | 0 | 0 | 32 | 1920 | | | | | |

[Copy](#)

Name resolution

[Close](#)

Você pode usar a caixa de diálogo **Endpoints** para filtrar pacotes específicos e exibi-los no painel **Packet List**. Se você clicar com o botão direito do mouse em um determinado dispositivo final, serão mostradas várias opções, incluindo a capacidade de criar um filtro para exibir apenas o tráfego deste dispositivo ou todo o tráfego excluindo o do dispositivo selecionado. Como bônus, você também poderá exportar diretamente o dispositivo final através de uma regra de colorização.

Conversações

A conversação em uma rede, parece uma conversa entre duas pessoas, descrever uma comunicação que ocorre entre dois hosts (endpoints). Por exemplo, a conversa entre Jim e Sally pode consistir em "Ei, como vai você? "Estou ótimo! E você? "Poderia estar melhor!". Uma conversa entre 192.168.1.5 e 192.168.0.8 poderá parecer "SYN", "SYN / ACK", e "ACK". (Nós vamos olhar para o processo de comunicação TCP / IP com mais detalhes no Capítulo 6.)

O Wireshark fornece caixa de diálogo **Conversations** (**Statistics > Conversations**), mostrado na figura abaixo. Você vai ver os endereços dos dispositivos finais envolvidos na conversa listados como endereço A e endereço B, serão exibidas colunas com os pacotes e bytes transmitidos de cada dispositivo. As conversas constantes nesta janela são divididas através do protocolo em uso, e podem ser selecionadas através das guias na parte superior da janela. Clicando com o botão direito do mouse em uma conversa específica lhe permitirá criar filtros que podem ser úteis, tais como mostrando todo o tráfego transmitido pelo dispositivo A, todo o tráfego recebido pelo dispositivo B, ou todo o tráfego de comunicação entre os dispositivos A e B.

Conversations: slowrouter.dmp

| Ethernet: 2 | Fibre Channel | FDDI | IPv4: 4 | IPX | JXTA | SCTP | TCP: 23 | Token Ring | UDP: 2 | WLAN | NCP | RSVP |
|------------------------|-------------------|---------|---------|--------------|------------|--------------|------------|------------|--------|------|-----|------|
| Ethernet Conversations | | | | | | | | | | | | |
| Address A | Address B | packets | Bytes | Packets A->B | Bytes A->B | Packets A<-B | Bytes A<-B | | | | | |
| Intel_bc:c0:b9 | Broadcast | 32 | 1920 | 32 | 1920 | 0 | 0 | | | | | |
| Intel_bc:c0:b9 | AmbitMic_aa:af:80 | 233 | 42002 | 93 | 16374 | 140 | 25628 | | | | | |

[Copy](#)

Name resolution

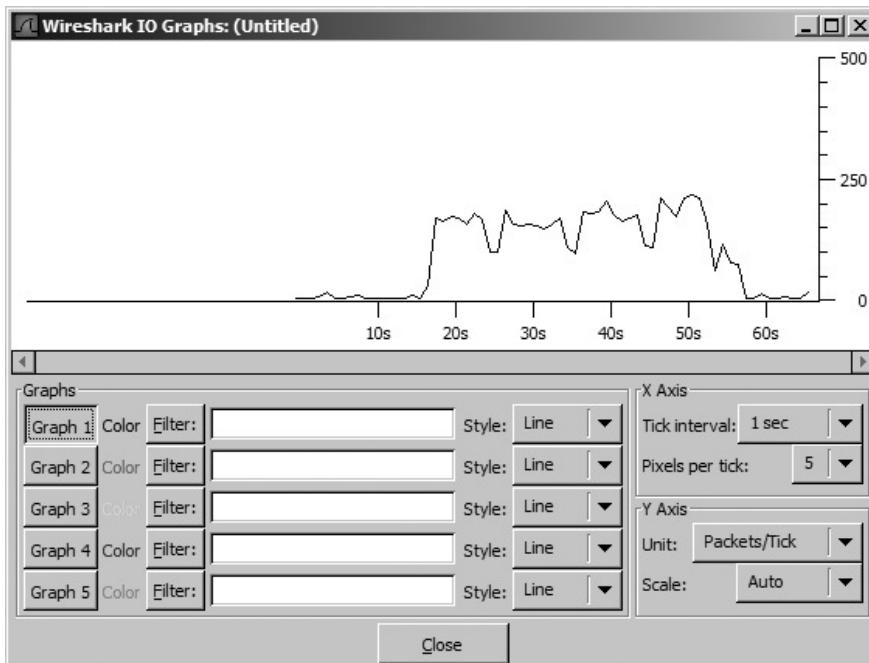
[Close](#)

A Janela IO Graphs

Uma das melhores maneiras de visualizar tendências é visualizá-las graficamente. A janela **IO Graphs** do Wireshark permite mostrar um gráfico da taxa de transferência de dados em uma rede. Você pode usar esse recurso para procurar pontos ou pausas na transferência de um protocolo específico durante todo um dia na sua rede.

Vamos olhar um gráfico de um IO através de um download de um arquivo na internet em um computador específico. Abra o arquivo de rastreamento FileDownload.dmp e, em seguida selecione **Statistics > IO Graphs**. Aqui você pode ver o número pequeno de bytes por segundo no início da captura, até o gráfico mostrar os picos por um período maior de tempo, enquanto o arquivo está sendo baixado (figura abaixo).

Você pode personalizar diversas características deste gráfico. As duas coisas mais importantes que você poderá modificar são as definições para o eixo-x e y do gráfico, que permitirão modificar os intervalos e as unidades usadas para exibir as informações de transferência.



Observe que a maioria das opções configuráveis é composta por uma área onde você pode criar filtros. Você pode criar até cinco filtros originais (usando a mesma sintaxe tanto para visualizar ou capturar) e especificar as cores de exibição para os filtros. Por exemplo, você pode criar filtros para mostrar o tráfego ARP e DHCP e mostrar as linhas do gráfico em vermelho e azul, para que possa mais facilmente diferenciar as tendências de transferência entre estes dois tipos de protocolo.

Embora algumas dessas características nos pareçam serem úteis em situações obscuras, provavelmente você vai se deparar usando-as mais do que possa imaginar. É importante que você se familiarize com essas janelas e opções, porque faremos referências a elas em alguns dos próximos capítulos.

6

PROTOCOLOS MAIS COMUNS

Este capítulo é uma síntese de alguns dos protocolos mais comuns que aparecem no Wireshark. Vamos olhar arquivos de exemplo contendo os vários protocolos e em seguida discutir sobre a forma como cada um funciona. Meu objetivo aqui é ajudar você a entender cada um desses protocolos e dar-lhe uma base para comparação quando os tiver analisando, por suspeita de mau funcionamento . Este capítulo contém algumas informações básicas sobre esses protocolos.

Nota:

Não vou entrar em grandes detalhes sobre o projeto de cada protocolo individualmente, em vez disso, eu fornecerei o número da RFC associada a cada um deles. Uma RFC , ou pedido de comentários, é o documento oficial que define as normas de execução dos protocolos na Pilha TCP/IP. Você pode procurar a documentação RFC na home page do RFC Editor, <http://www.rfc-editor.org>.

Protocolo de Resolução de Endereços – ARP

Vamos começar com o Address Resolution Protocol (ARP), porque ele é um dos protocolos mais simples, necessitando apenas de alguns pacotes para completar a operação inteira. O ARP (RFC 826) é usado para traduzir endereços da camada 3 (IP) em endereços da camada 2 (MAC), permitindo assim que os dispositivos (como switches e roteadores) determinem onde os outros dispositivos estão localizados em suas portas.

A coisa engraçada sobre ARP é que ele realmente presta serviço a duas diferentes camadas do modelo OSI: a camada de rede e a camada de enlace de dados.

Quando um computador quer transmitir dados para outro computador, ele deve primeiro saber onde é que o computador está. Isso é feito com a ajuda do switch ou roteador que conecta os dois computadores e o protocolo ARP. Agora, dê uma olhada no nosso arquivo de captura, como mostrado na figura abaixo.

Note que o primeiro pacote enviado por nosso computador de origem (01:16:ce:6e:8b:24), é um pacote enviado a ff:ff:ff:ff:ff:ff perguntando: Quem é 192.168.0.1?.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-------------------|-------------------|----------|---|
| 213 | *REF* | 00:16:ce:6e:8b:24 | ff:ff:ff:ff:ff:ff | ARP | who has 192.168.0.1? tell 192.168.0.114 |
| 214 | 0.004081 | 00:13:46:0b:22:ba | 00:16:ce:6e:8b:24 | ARP | 192.168.0.1 is at 00:13:46:0b:22:ba |

Como você aprendeu anteriormente, a mudança só funciona na camada 2, não há conhecimento sobre o endereço do computador da camada 3. O que o computador faz, então? Bem, o que você faz quando você não sabe o primeiro nome do Smith ao qual deseja se comunicar? Você pega a lista telefônica e liga para Smith que encontrar até chegar ao Smith desejado!

O ARP fornece a funcionalidade para localizar o endereço do cliente na camada 3 permitindo que o computador de origem envie um broadcast ARP. Este broadcast é enviado para o endereço da camada 2 ff:ff:ff:ff:ff:ff (padrão de endereçamento broadcast), o pacote é então enviado para todos os computadores que se encontram no mesmo domínio de broadcast.

A função desse pacote é perguntar a cada computador que se encontra conectado se possui ou não o endereço IP 192.168.0.1. Computadores com um endereço IP diferente simplesmente descartam o pacote, enquanto o que tem o endereço IP, envia uma resposta contendo seu endereço de Camada 2 ao computador de origem.

O segundo pacote (também mostrado na figura acima) mostra a resposta ARP ao computador que enviou o primeiro pacote. A resposta é muito simples um: 192.168.0.1 é 00:13:46:0b:22:ba. Deste ponto em diante, o computador de origem saberá o endereço da camada 2 do computador de destino e será capaz de enviar dados diretamente para ele.

Protocolo de Configuração Dinâmica – DHCP

O Dynamic Host Configuration Protocol (DHCP) é outro protocolo bastante simples. O DHCP (RFC 2131) fornece automaticamente aos clientes as informações relacionadas às configurações de rede, como um nome de domínio, endereço do servidor NTP, ou um único endereço de camada 3 (IP). O processo de comunicação DHCP é um tipo de comunicação cliente/servidor no qual o computador cliente solicita um endereço IP a um servidor DHCP e o servidor reconhece-o dando-lhe um.

A funcionalidade básica do servidor DHCP é um processo simples de quatro etapas. O processo começa com o pacote 1 quando o computador cliente envia um pacote DHCP Discover para o endereço de broadcast IP 255.255.255.255 (como mostrado na figura abaixo).

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------|-----------------|----------|---------------------------------------|
| 1 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Discover - Transaction ID 0x3d1d |

Quando um cliente deseja obter um endereço IP em uma rede, ele deve primeiro localizar um servidor de DHCP válido. Isso é feito através do envio de um pacote de broadcast tentando localizar um servidor de DHCP válido na rede. Quando um servidor DHCP válido recebe um destes pacotes, ele envia uma resposta ao cliente com um pacote DHCP Offer, como visto no pacote 2 (figura abaixo). Este pacote contém o endereço IP que o servidor DHCP quer atribuir ao cliente e outras informações que o servidor está configurado para oferecer.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-------------|--------------|----------|------------------------------------|
| 2 | 0.000295 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP Offer - Transaction ID 0x3d1d |

Depois que o cliente recebe este pacote, ele solicita as informações de endereçamento a partir do servidor DHCP, enviando um pacote de solicitação, que é o nosso pacote 3 do nosso arquivo de exemplo. Desde que o cliente ainda não tenha se configurado com o endereço IP fornecido, esse pacote é novamente enviado como um broadcast, o que diz ao servidor que o cliente aceitou a sua oferta e notifica a todos os outros servidores DHCP na rede que o cliente não está mais aceitando outras ofertas. Depois que o servidor recebe este pacote, ele atribui esse endereço IP para o cliente e envia um DHCP ACK de volta para o cliente, como pode ser visto em pacote de 4 (figura abaixo), significando o final da transação DHCP.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-------------|--------------|----------|----------------------------------|
| 1 | 0.000000 | 192.168.0.1 | 192.168.0.10 | DHCP | DHCP ACK - Transaction ID 0x3d1e |

Observe que cada transação DHCP tem um ID específico da transação que podem ser visto através do cabeçalho de informação no painel Packet List. Esse ID da transação permitir que o servidor DHCP possa identificar e separar cada transação do cliente. Isto é importante porque permite que você mantenha cada transação separada no processo de análise.

Embora nós discutimos apenas quatro, você pode encontrar até oito diferentes tipos de pacotes DHCP em um arquivo de captura. (Para mais informações sobre estas e outras funções DHCP, leia o RFC DHCP).

TCP/IP e HTTP

O TCP / IP é a base para quase toda a comunicação que discutiremos neste livro. Porque é o protocolo de rede mais amplamente utilizado, vamos nos concentrar nele.

O Hypertext Transfer Protocol (HTTP, RFC 2616) é baseado na comunicação cliente/servidor, é usado para transferir páginas da web através de uma rede. Um HTTP transação simples é um bom exemplo de comunicação TCP / IP. Toda vez que você pesquisar na Internet com o Google, verificar o tempo, ou mesmo verificar a sua equipe de esporte favorita, você estará transferindo dados via TCP / IP usando o HTTP.

TCP/IP

O protocolo TCP/IP é realmente uma pilha de protocolos, composto por diferentes protocolos em ambas as camadas 3 e 4 do modelo OSI. Estes protocolos incluem o TCP, IP, ARP, DHCP, ICMP, e muitos outros.

O Transmission Control Protocol (TCP, RFC 793) é um protocolo da camada 4 que é comumente utilizado porque fornece um método eficiente de transparência, comunicação confiável e bidirecional entre os dispositivos. A comunicação Bi-direcional permite que os dados possam ser transmitidos e recebidos simultaneamente a partir de um único servidor.

Todos os benefícios e as características do TCP são possíveis através de diferentes tipos de pacotes TCP. Nos próximos parágrafos vamos olhar para esses diferentes tipos de pacotes e o que eles fazem.

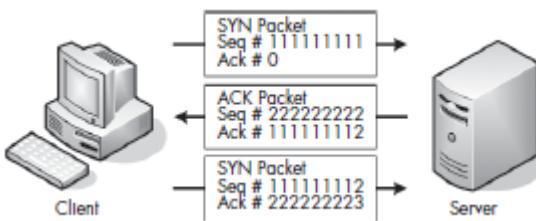
O Internet Protocol (IP, RFC 791) é um protocolo de camada 3 que fornece o sistema de endereçamento que permite a comunicação em uma rede. O IP é um protocolo sem conexão , o que significa que requer a funcionalidade do pacote TCP, para garantir a confiabilidade dos dados transmitidos.

O tráfego no arquivo de captura começa com a criação de uma sessão TCP/IP, seguida pelo pedido e transmissão de dados HTTP e o encerramento da sessão. Passo a passo através desta simples comunicação entre cliente e servidor nos ajudará a compreender como TCP e o IP trabalham.

Estabelecendo uma Sessão

Antes que você possa transferir dados para outro computador, o remetente e o receptor precisam completar um processo de handshake TCP. Um handshake TCP consiste em um processo de três etapas pelo qual o computador de origem (o cliente, neste exemplo) estabelece uma conexão com o computador de destino (o servidor). Você pode ver o processo de handshake nos primeiros três pacotes de nosso arquivo de captura, e é detalhado visualmente na figura abaixo.

Agora é um momento muito bom para ir em frente e estabelecer uma conexão entre os nossos computadores cliente e servidor. O computador cliente é mostrado no primeiro pacote com endereço IP 145.254.160.237. O computador servidor é mostrado no primeiro pacote com Endereço IP 65.208.228.223.



O Pacote SYN

Para iniciar o processo de handshake, o cliente envia um pacote SYN para o servidor; este pacote é usado para estabelecer a sincronização com o servidor, garantindo a ordem da comunicação entre o cliente e o servidor. O pacote SYN carrega com ele um número de seqüência de 32 bits, localizado no cabeçalho de um pacote TCP.

Para visualizar as informações de um pacote TCP, incluindo o seu número de seqüência, expanda a seção TCP no painel **Packet Details** do Wireshark. (Você usará esta seção com freqüência, porque ela contém uma variedade de informações úteis, incluindo a origem e o destino das portas utilizadas, o número de seqüência, o tipo de pacote TCP, e outras opções de TCP específica.) Observe no arquivo de captura que o primeiro pacote SYN tem o número de seqüência é 0, como mostrado na figura abaixo.

```

Frame 1 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
Internet Protocol, Src: 145.254.160.237 (145.254.160.237), Dst: 65.208.228.223 (65.208.228.223)
Transmission Control Protocol, Src Port: 3372 (3372), Dst Port: http (80), seq: 0, Len: 0
    Source port: 3372 (3372)
    Destination port: http (80)
    Sequence number: 0 (relative sequence number)
    Header length: 28 bytes
    Flags: 0x02 (SYN)
        window size: 8760
        checksum: 0xc30c [correct]
    Options: (8 bytes)
        Maximum segment size: 1460 bytes
        NOP
        NOP
        SACK permitted

```

Nota:

No Wireshark, os números da seqüência TCP são tratados como "parente" por padrão. O Wireshark ajusta o primeiro número da seqüência em um fluxo de comunicação a fim de que ele seja 0 independente do seu verdadeiro valor. Isso é feito para que os números de seqüência sejam mais fáceis de serem seguidos.

A resposta do Servidor SYN/ACK

O próximo passo no processo de handshake é a resposta do servidor. Uma vez que o servidor recebe o pacote SYN inicial do cliente, ele lê o número de seqüência e usa esse número no pacote que ele retorna. A resposta é chamada SYN/ACK, e é visto em dois pacotes do exemplo.

Uma parte do pacote ACK reconhece o pacote SYN, em outras palavras, ele diz ao computador cliente que o servidor recebeu o pacote SYN. Ele faz isso incrementando em um o número de seqüência enviado no pacote SYN original e o usa como o número de confirmação do pacote ACK. Quando o cliente recebe o número de confirmação contendo o número de seqüência original SYN , ele sabe que o servidor pode receber sua comunicação, e vice-versa. O objetivo do SYN parte da SYN/ACK é o mesmo do pacote SYN original: Ele é usado para transmitir um número de seqüência que o sistema cliente pode utilizar para confirmar a recepção.

O Pacote ACK Final

Finalmente, o cliente envia um pacote ACK para o servidor. Este pacote informa ao servidor que o cliente recebeu o SYN/ACK. Tal como acontece com as duas etapas do processo, o número de seqüência é incrementado em um e enviado como um número de confirmação para o servidor. Uma vez que esse ultimo pacote ACK é recebido, a comunicação pode começar.

Iniciando o Fluxo de Dados

Uma vez que o handshake foi estabelecido, todos os pacotes enviados nesta sessão particular entre o cliente e o servidor serão usados os números de seqüência para se certificar da ordem de seqüência do envio e recebimento dos pacotes. No entanto, a partir de agora, esses pacotes serão incrementados pelo tamanho do

quadro de dados a serem transmitidos, ao invés de um. (Para saber mais sobre como os pacotes TCP mantêm-se organizados, olhar a RFC 793).

Pedido e Transmissão HTTP

Uma vez que a sessão de comunicação foi estabelecida, é hora do pedido e transmissão da página da web que você está tentando exibir. Isto envolve HTTP e TCP.

O processo começa no pacote 4, o nosso primeiro pacote HTTP, pede ao servidor para transmitir a página web para o cliente. Vá em frente e expanda a seção HTTP deste pacote no painel **Packet Details** para ver as informações específicas do protocolo relacionado a este pedido (conforme mostrado na figura abaixo).

```
□ Hypertext Transfer Protocol
  □ GET /download.html HTTP/1.1\r\n
    Request Method: GET
    Request URI: /download.html
    Request Version: HTTP/1.1
    Host: www.ethereal.com\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113\r\n
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Referer: http://www.ethereal.com/development.html\r\n
  \r\n
```

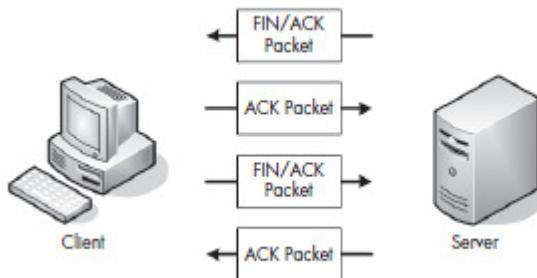
Como você pode ver, este pacote contém um comando GET (Request Método: GET) para a página web download.html no domínio www.ethereal.com domínio (Request URL: / download.html e Host: www.ethereal.com). Você vai notar também outras informações que podem ser úteis, tais como codificação de caracteres (Accept-Charset:ISO-8859-1), e a localização em referência (Referer: <http://www.ethereal.com/Development.html> r \ n).

Depois do HTTP ter feito o pedido inicial através do GET request, o TCP assume o processo de transferência dos dados. Durante o resto do arquivo de captura você vai ver este processo repetido: HTTP faz o pedido de dados ao servidor, e o servidor irá usar TCP para o transporte dos dados de volta ao cliente. O servidor permite que o cliente reconheça o pedido válido enviando uma mensagem HTTP OK antes de transmitir os dados. (Você pode ver o GET correspondente e o pacote OK no arquivo de exemplo, nos pacotes 4 e 38, mostrado na figura abaixo).

| No. | Time | Source | Destination | Protocol | Info |
|---|----------|-----------------|-----------------|----------|-----------------------------|
| 4 | 0.911310 | 145.254.160.237 | 65.208.228.223 | HTTP | GET /download.html HTTP/1.1 |
| □ Transmission Control Protocol, Src Port: 3372 (3372), Dst Port: http (80), Seq: 0, Ack: 0, Len: 479 | | | | | |
| Source port: 3372 (3372) Destination port: http (80) Sequence number: 0 (relative sequence number) [Next sequence number: 479 (relative sequence number)] Acknowledgement number: 0 (relative ack number) Header length: 20 bytes Flags: 0x18 (PSH, ACK) window size: 9660 Checksum: 0xa958 [correct] | | | | | |
| No. | Time | Source | Destination | Protocol | Info |
| 38 | 4.846969 | 65.208.228.223 | 145.254.160.237 | HTTP | HTTP/1.1 200 OK (text/html) |
| □ Transmission Control Protocol, Src Port: http (80), Dst Port: 3372 (3372), Seq: 17941, Ack: 480, Len: 424 | | | | | |
| Source port: http (80) Destination port: 3372 (3372) Sequence number: 17941 (relative sequence number) [Next sequence number: 18365 (relative sequence number)] Acknowledgement number: 480 (relative ack number) Header length: 20 bytes Flags: 0x18 (PSH, ACK) window size: 6432 Checksum: 0x3d97 [correct] [SEQ/ACK analysis] TCP segment data (424 bytes) [Reassembled TCP Segments (18364 bytes): #6(1380), #8(1380), #10(1380), #11(1380), #14(1380), #16(1380), | | | | | |

Terminando uma Sessão

Quando não há mais dados para serem enviados através da conexão estabelecida, a conexão pode ser terminada de uma maneira muito similar ao handshake TCP inicial. Ao invés de usar os pacotes SYN e ACK no entanto, este processo utiliza e pacotes FIN ACK, conforme mostrado na figura abaixo.



Quando o servidor termina a transmissão de dados, ele envia um FIN/ACK para o cliente, como mostrado na figura abaixo. O pacote FIN foi projetado para finalizar a conexão.

| |
|---|
| Transmission Control Protocol, Src Port: http (80), Dst Port: 3372 (3372), Seq: 18365, Ack: 480, Len: 0 |
| Source port: http (80) |
| Destination port: 3372 (3372) |
| Sequence number: 18365 (relative sequence number) |
| Acknowledgement number: 480 (relative ack number) |
| Header length: 20 bytes |
| Flags: 0x11 (FIN, ACK) |
| Window size: 6432 |
| Checksum: 0x3c64 [correct] |
| [SEQ/ACK analysis] |

O cliente responde ao pacote FIN com um pacote ACK que usa os números de seqüência e as regras de incremento que ele encontra no pacote FIN. Isso finaliza a comunicação com o servidor. Embora o servidor possa ainda receber dados do cliente, a partir desse momento ele não mais transmitirá dados.

Para concluir o processo, o cliente deve iniciar o mesmo processo novamente com o servidor. O processo FIN/ACK deve ser iniciado e reconhecido por ambos, cliente e servidor.

Por exemplo, no pacote de 40, o servidor envia um FIN/ACK para o cliente, e o cliente responde com seu pacote ACK no pacote 41. Em seguida o cliente envia o seu próprio FIN/ACK para o servidor, e o servidor fecha a conexão com um pacote ACK pacotes 43, como mostrado na figura abaixo.

| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|-----------------|-----------------|----------|---|
| 40 | 17.905747 | 65.208.228.223 | 145.254.160.237 | TCP | http > 3372 [FIN, ACK] Seq=18365 Ack=480 Win=6432 Len=0 |
| 41 | 17.905747 | 145.254.160.237 | 65.208.228.223 | TCP | 3372 > http [ACK] Seq=480 Ack=18366 Win=9236 Len=0 |
| 42 | 30.063228 | 145.254.160.237 | 65.208.228.223 | TCP | 3372 > http [FIN, ACK] Seq=480 Ack=18366 Win=9236 Len=0 |
| 43 | 30.393704 | 65.208.228.223 | 145.254.160.237 | TCP | http > 3372 [ACK] Seq=18366 Ack=481 Win=6432 Len=0 |

Sistema de Nomes de Domínio – DNS

O Domain Name System (DNS, RFC 1034) traduz uma forma de endereço em outra, especificamente, traduz endereços de DNS, como www.google.com ou Marketing-PC1, em seus endereços IP correspondentes. Alguma forma de tradução de endereços é uma exigência, uma vez camada 3 do modelo OSI só pode localizar um computador se tiver o seu endereço IP.

A tradução DNS é um processo muito simples, e ela consegue na maioria casos terminar o processo utilizando apenas dois pacotes. O primeiro pacote é um pedido para o servidor DNS de sua rede local que pergunta: Qual é o endereço IP do www.google.com? O segundo pacote é a resposta do servidor DNS, dizendo que www.google.com reside em um servidor com o endereço IP de XX.XX.XX.XXX.

Vamos dar uma olhada no DNS em ação (veja a figura abaixo). Observe que no primeiro pacote do arquivo um pacote DNS da origem 192.168.0.114 está solicitando o endereço IP de http://www.chrianders.org do destino 205.152.37.23.

O endereço IP de destino recebe a consulta e responde com o pacote 2, que contém o endereço IP do site solicitado, 208.113.140.24. Quando esse processo estiver concluído, a camada 3 assume o processo e completa o seu handshake TCP para que a transferência de dados possa começar.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|---------------|---------------|----------|--|
| 1 | 0.000000 | 192.168.0.114 | 205.152.37.23 | DNS | Standard query A chrianders.org |
| 2 | 0.112121 | 205.152.37.23 | 192.168.0.114 | DNS | Standard query response A 208.113.140.24 |

Nota:

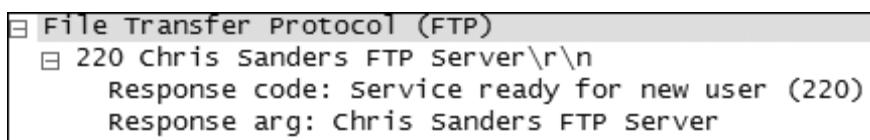
Quando você examinar o arquivo de captura, você verá várias consultas DNS diferentes. Muitas vezes uma simples página web por necessitar de informação poderá gerar uma série de consultas em vários servidores. Tente criar um filtro de exibição para mostrar apenas o tráfego DNS e veja se você pode determinar quantas diferentes consultas DNS se encontram neste arquivo.

Protocolo de Transferência de Arquivos – FTP

O File Transfer Protocol (FTP, RFC 959) é um protocolo da camada 7 que é utilizado para transferir dados entre um servidor e um cliente. Utiliza as portas 20 e 21, o FTP é um dos utilitários de transferência de arquivos mais usados. Porque o FTP é um protocolo cliente/servidor, todas as comunicações presentes no arquivo de captura mostra o tráfego entre um computador cliente e um computador servidor. Tal como acontece com todos os processos TCP, o FTP começa com um handshake TCP padrão, como mostrado no pacote 1 da figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|--|
| 1 | 0.000000 | 192.168.0.114 | 192.168.0.193 | TCP | 1137 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 0.002319 | 192.168.0.193 | 192.168.0.114 | TCP | ftp > 1137 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1452 |
| 3 | 0.002338 | 192.168.0.114 | 192.168.0.193 | TCP | 1137 > ftp [ACK] Seq=1 Ack=1 Win=17424 Len=0 |

Uma vez concluído o processo de handshake, o servidor envia uma mensagem de boas vindas para o cliente. Esta mensagem identifica o servidor como um servidor FTP e diz ao cliente que o servidor está pronto para aceitar suas credenciais de login, como mostrado na figura abaixo.

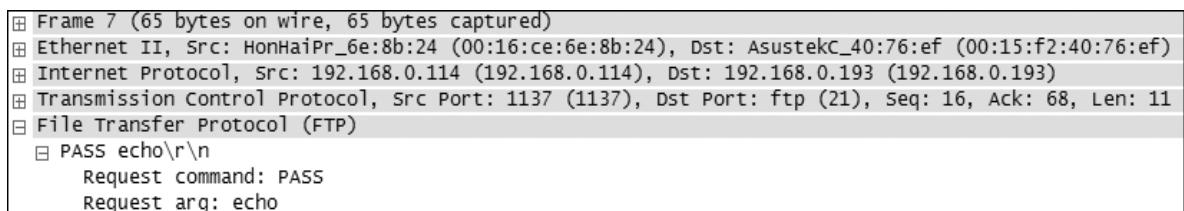


Através dos próximos pacotes, o cliente envia um username (csanders) e uma senha (echo) para o servidor, e o servidor as reconhece (figura abaixo).

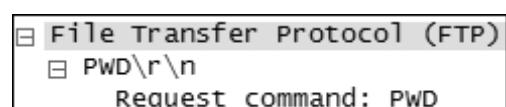
| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|---|
| 5 | 0.005259 | 192.168.0.114 | 192.168.0.193 | FTP | Request: USER csanders |
| 6 | 0.006560 | 192.168.0.193 | 192.168.0.114 | FTP | Response: 331 Password required for csanders. |
| 7 | 0.007647 | 192.168.0.114 | 192.168.0.193 | FTP | Request: PASS echo |
| 8 | 0.009936 | 192.168.0.193 | 192.168.0.114 | FTP | Response: 230 User csanders logged in. |

Esta comunicação está apresentada na coluna da **Info** do painel **Packet List**, mas essa janela só dá um breve resumo do conteúdo do pacote. Se você quiser ir um pouco mais profundo, você pode expandir a seção **FTP** no painel **Packet Details**.

Note que a criptografia não é usada no nosso exemplo, então a senha de FTP pode ser vista claramente no arquivo de captura no pacote 7 (figura abaixo).



Uma conexão cliente usa uma lista de comandos para interagir com um servidor FTP. Eles podem exibir o conteúdo de um diretório, percorrer um diretório, baixar ou apagar um arquivo, e assim por diante. (Para uma lista completa dos comandos disponíveis visíveis em um pacote FTP, consulte RFC 959). Vamos olhar um pouco os comandos FTP usados no nosso arquivo de exemplo, começando com o pacote 15, mostrado na figura ao lado.



Comando CWD

Como você pode ver, o pacote 15 mostra um comando CWD enviado do cliente para o servidor. O CWD está para mudar o diretório de trabalho, e esse comando é enviado toda vez que você falar com um cliente de FTP para mover para um diretório diferente do servidor.

Observe neste exemplo que o comando CWD inclui os pedidos para alterar o diretório de trabalho para /, que é o diretório raiz FTP do servidor. Quando você se loga pela primeira vez em um servidor FTP, o comando CWD é enviado para mudar para o diretório raiz, /. Uma vez que o servidor recebe este comando CWD, ele muda para o diretório raiz e diz ao cliente que / é agora o diretório de trabalho atual.

Comando SIZE

O comando seguinte é o SIZE, mostrado na figura ao lado. Este comando informa o tamanho (Em bytes) de um arquivo particular, e sempre é enviado com um nome de arquivo.

```
File Transfer Protocol (FTP)
  SIZE Music.mp3\r\n
    Request command: SIZE
    Request arg: Music.mp3
```

Note que no pacote 25, o cliente envia o comando SIZE para o servidor para solicitar o tamanho do music.mp3 arquivo. O pacote 26 (figura ao lado) mostra a resposta do servidor, cujo tamanho do arquivo é 4.980.924 bytes.

```
File Transfer Protocol (FTP)
  213 4980924\r\n
    Response code: File status (213)
    Response arg: 4980924
```

Comando RETR

O comando RETR (recuperação), mostrado na figura ao lado, é usado pelo cliente para solicitar o download de um arquivo ao servidor. No pacote 32, o cliente envia o comando RETR para o servidor, solicitando o download do arquivo music.mp3. Depois que o servidor recebe este pedido, começa a enviar os dados para o cliente.

```
File Transfer Protocol (FTP)
  RETR Music.mp3\r\n
    Request command: RETR
    Request arg: Music.mp3
```

Nota:

Os pacotes rotulados como FTP-DATA são aqueles que contêm um arquivo que está sendo baixado ou carregado para um servidor.

Protocolo Telnet

O protocolo Telnet (RFC 854) não é um protocolo seguro, utiliza uma comunicação baseada em texto entre o cliente e servidor. Ele é freqüentemente usado para administrar remotamente servidores, switches, roteadores e outros dispositivos de hardware de rede.

Nota:

Você pode tornar a sua comunicação mais segura usando SSH ao invés do telnet.

Que tipo de comunicação está ocorrendo no presente intercâmbio entre o servidor e o cliente? Começando pelo topo, poderemos obter várias conclusões. O primeiro de vários pacotes de confirmação que estamos definitivamente vendo no tráfego telnet, é por causa das configurações telnet específicas envolvidas entre os dois dispositivos como mostrado na figura ao lado.

```
Telnet
  Command: will Suppress Go Ahead
  Command: Do Terminal Type
  Command: Do Negotiate About Window Size
  Command: Do Terminal Speed
  Command: Do Remote Flow Control
  Command: Do Linemode
  Suboption Begin: Linemode
  Command: Suboption End
```

Cada sessão telnet usa várias opções exclusivas para especificar as taxas de comunicação e os modos de transferência de dados, que devem ser sincronizadas entre o cliente e o servidor antes da comunicação iniciar. Estas opções fazem parte dos primeiros 30 pacotes no arquivo de exemplo.

O primeiro pacote interessante é o número 27, que identifica o servidor como um servidor OpenBSD. O pacote 29 apresenta um prompt de login para o cliente, e no pacote 31 você pode ver que o nome de usuário "fake" é enviado de volta para o servidor. No pacote 36 é requisitada uma senha do cliente, que é respondida no pacote 38 com "user", como mostrado na figura abaixo. Agora você pode ver o quanto o telnet é inseguro. Essa combinação de nome de usuário e senha poderia muito bem ser a senha administrativa

para um dos servidores mais importantes em sua rede, e ainda seria apresentado em texto claro que é lido por qualquer um com um farejador de pacotes e um pouco de conhecimento.

| |
|---|
| Frame 36 (75 bytes on wire, 75 bytes captured) |
| Ethernet II, Src: WesternD_9f:a0:97 (00:00:c0:9f:a0:97), Dst: Lite-OnC_3b:bf:fa (00:a0:cc:3b:bf:fa) |
| Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.2 (192.168.0.2) |
| Transmission Control Protocol, Src Port: telnet (23), Dst Port: 1550 (1550), Seq: 143, Ack: 207, Len: 9 |
| Telnet |
| Data: Password: |
| Frame 38 (72 bytes on wire, 72 bytes captured) |
| Ethernet II, Src: Lite-OnC_3b:bf:fa (00:a0:cc:3b:bf:fa), Dst: WesternD_9f:a0:97 (00:00:c0:9f:a0:97) |
| Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1) |
| Transmission Control Protocol, Src Port: 1550 (1550), Dst Port: telnet (23), Seq: 207, Ack: 152, Len: 6 |
| Telnet |
| Data: user\r\n |

O resto do arquivo de captura mostra o cliente usando a sessão telnet estabelecida sessão para pingar vários sites. Você pode observar esses dados exatamente como são transferidos, olhando para a seção telnet no painel Packet Details.

Serviço de Mensagens MSN

Você pode achar que precisa analisar o tráfego de uma conversa de mensagem instantânea por várias razões. Nós exploramos um cenário possível no Capítulo 5 quando se suspeita de um funcionário da empresa oferecer informações financeiras através de um software de envio de mensagens. Existem várias aplicações de envio de mensagens instantâneas, e enquanto cada uma utiliza seu próprio protocolo, existem certas similaridades em cada uma. Aqui vamos nos concentrar especificamente sobre o tráfego do MSN Messenger Service (MSNMS). Vamos ver se não podemos pegar alguns funcionários no ato.

Nota:

Algumas organizações têm políticas que impedem o uso de software de mensagens, e se esse for o caso, mesmo vendo o protocolo MSNMS em um arquivo de captura alarmes poderão ser disparados.

O arquivo de captura começa com uma comunicação TCP com um simples handshake entre dois clientes, conforme mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|---|
| 1 | 0.000000 | 192.168.0.114 | 207.46.26.167 | TCP | 3331 > 1863 [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 0.098754 | 207.46.26.167 | 192.168.0.114 | TCP | 1863 > 3331 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1452 |
| 3 | 0.098792 | 192.168.0.114 | 207.46.26.167 | TCP | 3331 > 1863 [ACK] Seq=1 Ack=1 Win=17424 Len=0 |

Na sequência deste handshake, o primeiro pacote MSNMS é enviado a partir do 192.168.0.114 para um servidor que resida fora da sua rede local (figura abaixo).

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|--|
| 4 | 0.098991 | 192.168.0.114 | 207.46.26.167 | MSNMS | USR 93 tesla_brian@hotmail.com 1835953129.20013021.2623242 |

Este pacote está sendo enviado de um computador em sua rede para um servidor remoto da Microsoft a fim de estabelecer um handshake que prepara a comunicação. Estes pacotes iniciais são marcados como pacotes USR, como visto na seção MSNMS do pacote no painel **Packet Details**. Você pode ver o endereço de e-mail da pessoa que inicia a conversa (Tesla_brian@hotmail.com) nestes pacotes iniciais (figura abaixo).

```
MSN Messenger Service
USR 93 OK tesla_brian@hotmail.com brian\r\n
```

Os próximos dois pacotes são marcados como pacotes CAL, como mostrado na figura abaixo. Os pacotes CAL são enviados do computador dentro da rede para um servidor MSN, a fim de estabelecer a comunicação com outro usuário MSNMS.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|---------------------------------|
| 6 | 0.199942 | 192.168.0.114 | 207.46.26.167 | MSNMS | CAL 94 tesla_thomas@hotmail.com |
| 7 | 0.300257 | 207.46.26.167 | 192.168.0.114 | MSNMS | CAL 94 RINGING 1835953129 |

Como você pode ver no pacote 7, o usuário correspondente MSNMS tem o endereço de email tesla_thomas@hotmail.com (figura abaixo).

| |
|--|
| Frame 6 (87 bytes on wire, 87 bytes captured) |
| Ethernet II, Src: HonHaiPr_6e:8b:24 (00:16:ce:6e:8b:24), Dst: D-Link_21:99:4c (00:05:5d:21:99:4c) |
| Internet Protocol, Src: 192.168.0.114 (192.168.0.114), Dst: 207.46.26.167 (207.46.26.167) |
| Transmission Control Protocol, Src Port: 3331 (3331), Dst Port: 1863 (1863), Seq: 61, Ack: 42, Len: 33 |
| MSN Messenger Service CAL 94 tesla_thomas@hotmail.com\r\n |

O servidor reconhece agora que recebeu um pacote CAL no pacote 7 e 8 (figura abaixo).

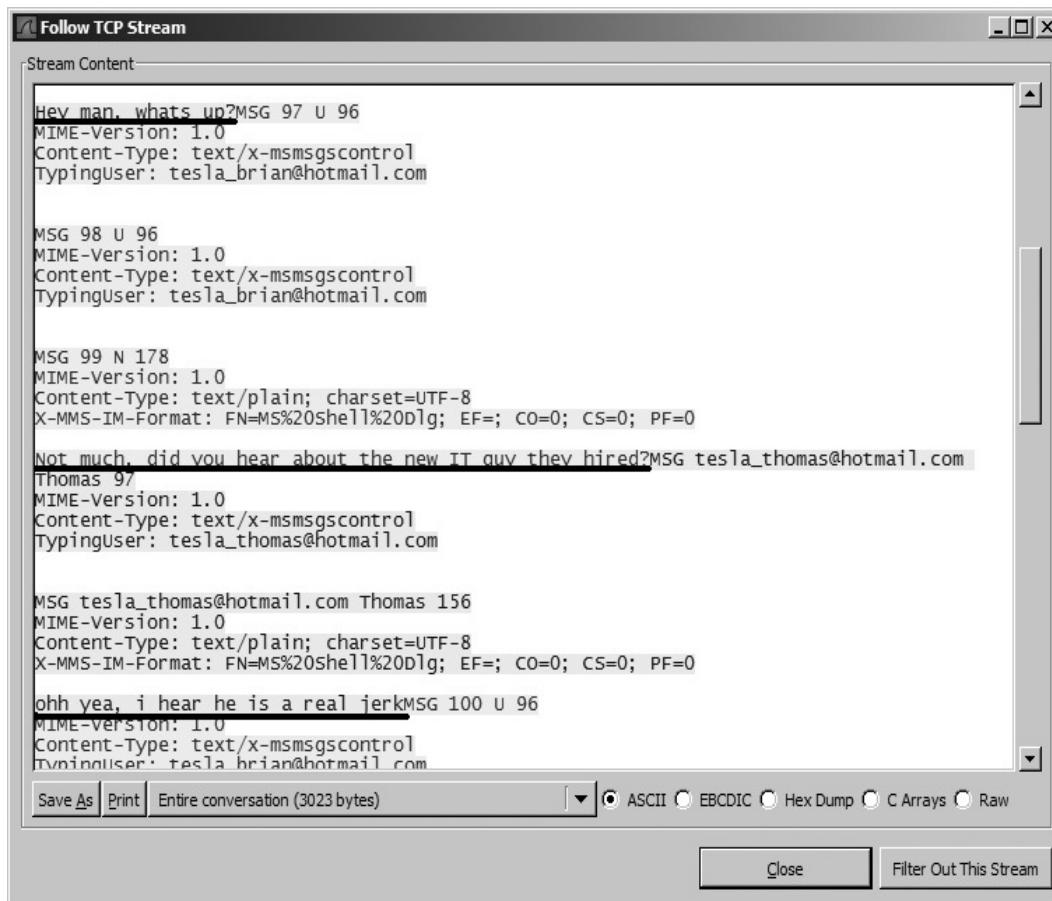
| No. . | Time | Source | Destination | Protocol | Info |
|------------|------|---------------|---------------|----------|---|
| 7 0.300257 | | 207.46.26.167 | 192.168.0.114 | MSNMS | CAL 94 RINGING 1835953129 |
| 8 0.442314 | | 192.168.0.114 | 207.46.26.167 | TCP | 3331 > 1863 [ACK] Seq=94 Ack=69 Win=17356 Len=0 |

O pacote 9 é o último pacote a ser enviado para estabelecer uma comunicação plena. Conforme mostrado na figura abaixo, o packet 9 é um pacote JOI enviado a partir do servidor MSN remoto, indicando que o outro membro (tesla_thomas@hotmail.com, neste caso) tem êxito e se juntou sessão e pode estabelecer a comunicação.

| No. . | Time | Source | Destination | Protocol | Info |
|------------|------|---------------|---------------|----------|--|
| 9 0.510484 | | 207.46.26.167 | 192.168.0.114 | MSNMS | JOI tesla_thomas@hotmail.com Thomas 1616756780 |

O restante do arquivo de captura contém apenas pacotes MSG, que são simplesmente as mensagens enviadas a partir de uma extremidade a outra, neste caso entre Brian e Thomas.

A primeira coisa que provavelmente vem à mente quando você pensa neste conceito é, posso realmente ver o que eles estão dizendo?! Bem, tão assustador como é, a resposta é sim. Tudo. Por um simples clique-direito nos pacotes MSG e selecionando **Follow TCP Stream** (como você aprendeu a fazer no capítulo 5), você pode ver a conversa completa entre Brian e Thomas (figura abaixo). Isso pode fazer você ser um pouco mais cuidadoso quando estiver usando enviadores de mensagens instantâneas enquanto trabalha!



Protocolo Internet de Controle de Mensagens – ICMP

O Internet Control Message Protocol (ICMP) é uma parte do protocolo IP, eu gosto de chamá-lo de protocolo de utilidade, pois é usado para solucionar problemas de outros protocolos. Se você já usou a utilidade de ping, você usou o protocolo ICMP.

Vamos ver o que o tráfego ICMP comum parece. A o arquivo de captura de arquivo incluído apenas contém oito pacotes. Há dois pings separados para dois hosts separados. Vamos olhar os detalhes do pacote um, mostrado na figura abaixo.

Se você expandir a seção ICMP, você vai ver o pouco que existe em um pacote ICMP. O primeiro pacote é rotulado como tipo 8, um pedido (ping). Cada pacote ICMP tem um tipo numérico associado a ele, que determina como o pacote é manipulado pela máquina de destino. (RFC 792 lista todos os diferentes tipos de pacotes ICMP).

| Internet Control Message Protocol | |
|-----------------------------------|-------------------------|
| Type: | 8 (Echo (ping) request) |
| Code: | 0 |
| Checksum: | 0x495c [correct] |
| Identifier: | 0x0300 |
| Sequence number: | 0x0100 |
| Data (32 bytes) | |

O senso comum nos diz que se um computador envia uma solicitação de eco, ele deve receber uma resposta de eco, e isso é o que vemos no arquivo de captura. O Pacote 2 é transmitido de volta a partir do computador remoto e é marcado como ICMP tipo 0, uma resposta eco (ping).

Um padrão de ping a partir de uma linha de comando do Windows pinga um host quatro vezes. Você pode ver o processo de ping no arquivo de captura e na figura abaixo também. O primeiro destino do ping: 192.168.0.1, recebe e responde a quatro pings. Depois disso, outro ping é iniciado a 72.14.207.99 (<http://www.google.com>), que também recebe e responde a quatro pings.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|---------------|---------------|----------|---------------------|
| 1 | 0.000000 | 192.168.0.114 | 192.168.0.1 | ICMP | Echo (ping) request |
| 2 | 0.001085 | 192.168.0.1 | 192.168.0.114 | ICMP | Echo (ping) reply |
| 3 | 0.996773 | 192.168.0.114 | 192.168.0.1 | ICMP | Echo (ping) request |
| 4 | 0.998983 | 192.168.0.1 | 192.168.0.114 | ICMP | Echo (ping) reply |
| 5 | 1.996801 | 192.168.0.114 | 192.168.0.1 | ICMP | Echo (ping) request |
| 6 | 1.999087 | 192.168.0.1 | 192.168.0.114 | ICMP | Echo (ping) reply |
| 7 | 2.996840 | 192.168.0.114 | 192.168.0.1 | ICMP | Echo (ping) request |
| 8 | 2.999177 | 192.168.0.1 | 192.168.0.114 | ICMP | Echo (ping) reply |

Considerações Finais

O objetivo deste capítulo foi apresentar a você o Wireshark para analisar os arquivos de captura e usar esses arquivos capturados para mostrar como alguns protocolos comuns trabalham. Embora tenhamos apenas brevemente apresentado alguns protocolos mais avançados, eu recomendo a leitura das suas RFCs, estudando-as com mais profundidade. Como o livro continua com vários cenários, os estaremos construindo sobre os conceitos básicos que você aprendeu aqui.

7

CENÁRIOS BÁSICOS

Agora chegamos na carne e ossos deste livro, estamos prontos para usar o Wireshark e analisar de verdade os problemas de rede.

Vamos começar dando uma olhada em algumas situações simples onde a nossa capacidade de analisar os pacotes vão nos ajudar a entender melhor o que está acontecendo por trás das cenas. Então, vamos olhar para alguns simples cenários de resolução de problemas do mundo real que você poderia possivelmente encontrar no dia a dia. Vamos mergulhar a fundo.

Perda da Conexão TCP

Um dos problemas mais comuns que encontramos quando estamos resolvendo problemas em uma rede é perda de conectividade. Por enquanto, vamos ignorar os motivos para a perda de conectividade e olhar para o que realmente ocasionou a perda a nível de pacote, e assim poderemos identificar esse tipo de problema.

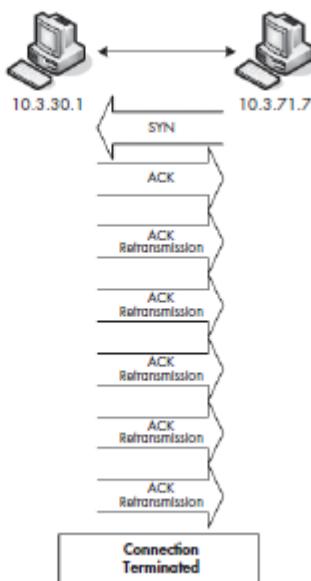
O pequeno arquivo de captura `tcp-lost.pcap-con` (figura abaixo) mostra uma perda de conectividade. O arquivo começa com quatro pacotes padrão TCP ACK enviados entre 10.3.71.7 e 10.3.30.1.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-----------|-------------|----------|--|
| 1 | 0.000000 | 10.3.71.7 | 10.3.30.1 | TCP | 1048 > 1048 [ACK] Seq=0 Ack=0 Win=8760 Len=0 |
| 2 | 0.000000 | 10.3.30.1 | 10.3.71.7 | TCP | 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 3 | 0.000000 | 10.3.71.7 | 10.3.30.1 | TCP | 1043 > 1048 [ACK] Seq=0 Ack=2920 Win=8760 Len=0 |
| 4 | 0.000000 | 10.3.71.7 | 10.3.30.1 | TCP | 1043 > 1048 [ACK] Seq=0 Ack=5840 Win=8760 Len=0 |

O problema começa no pacote 5, onde vemos a primeira retransmissão TCP pacotes (figura abaixo).

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-----------|-------------|----------|---|
| 5 | 0.206000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 6 | 0.806000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 7 | 2.006000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 8 | 4.406000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 9 | 9.211000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |

Por desenho, quando o TCP envia um pacote para um destino e não tem resposta, ele espera um determinado período de tempo e depois retransmite o pacote original. Se não receber a resposta, a fonte (origem) do computador duplica a quantidade de tempo e aguarda por uma resposta antes de enviar outra retransmissão. O conceito de retransmissão TCP é ilustrado na figura abaixo.



Conforme mostrado na figura abaixo, o processo se repete até que as cinco tentativas de retransmissão TCP sejam concluídas, o que sempre leva aproximadamente 9,6 segundo em sua aplicação Windows. Depois de cinco tentativas de retransmissão falhar, a conexão falha completamente e transmissão de dados é perdida.

Se você definir o formato de exibição de tempo do Wireshark para mostrar o tempo que tem decorrido desde o pacote capturado anteriormente (**View > Time Display Format > Seconds Since Beginning of Capture**), você poderá visualizar o incremento do tempo entre os pacotes (figura abaixo).

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-----------|-------------|----------|---|
| 2 | 0.000000 | 10.3.30.1 | 10.3.71.7 | TCP | 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 5 | 0.206000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 6 | 0.600000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 7 | 1.200000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 8 | 2.400000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |
| 9 | 4.805000 | 10.3.30.1 | 10.3.71.7 | TCP | [TCP Retransmission] 1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648 |

Agora, dê uma olhada nos pacotes retransmitidos na figura abaixo. Observe que o número de seqüência (Seq = 5.840) corresponde ao número de ACK do pacote cinco mostrado na parte inferior da figura acima (ACK = 5.840).

Como você aprendeu no capítulo 6, o TCP usa estes números SEQ e ACK para manter um fluxo TCP em ordem. Como o número SEQ mostrado na retransmissão corresponde ao número ACK do pacote, você sabe que é o pacote 5 que foi perdido e agora está sendo retransmitido. A capacidade de localizar o pacote exato em que uma tentativa de retransmissão TCP começa conduz muitas vezes você a pistas que ajudam a determinar exatamente por que essa perda de conectividade ocorreu.

Destinos e Códigos ICMP não encontrados

Ao testar a conectividade da rede, uma das ferramentas mais utilizadas é o utilitário ICMP ping. Se você tiver sorte, o destino que você está pingando vai responder, dizendo que seu ping foi bem sucedido. Infelizmente, muitas vezes você não vai receber uma resposta de ping para trás quando você estiver resolvendo um problema, você receberá uma mensagem Destination unreachable (destino inacessível). Usando um farejador de pacotes (sniffer) em conjunto com um utilitário ICMP poderá obter um pouco mais de informação que apenas o ICMP sozinho faria. Vamos ver se não podemos usar essa mensagem de erro ICMP para isolar o problema.

Destino Inacessível (Unreachable Destination)

Quando você abrir o arquivo destunreachable.pcap, você notará que o primeiro pacote no arquivo de captura é o seu padrão Echo (ping) pacote de solicitação (também conhecido como um pacote ICMP tipo 8) de 10.2.10.2 para 10.4.88.88, como mostrado na figura abaixo.

Para verificar isso, olhe para a seção ICMP do painel **Packet Details**, você deve ver este pacote identificado como tal. Normalmente, porém, que você iria querer receber um Echo (ping) pacote de resposta (também conhecido como um pacote ICMP tipo 0) em resposta ao seu ping.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-----------|-------------|----------|---------------------|
| 1 | 0.000000 | 10.2.10.2 | 10.4.88.88 | ICMP | Echo (ping) request |

Examinando o pacote 2 na figura abaixo, você pode ver que ele também não é um pacote tipo 0, mas sim um pacote tipo 3, que é retornado quando um destino que você estão tentando ping está inacessível.

| |
|---|
| Internet Control Message Protocol |
| Type: 3 (Destination unreachable) |
| Code: 1 (Host unreachable) |
| Checksum: 0xa7a2 [correct] |
| Internet Protocol, Src: 10.2.10.2 (10.2.10.2), Dst: 10.4.88.88 (10.4.88.88) |
| Internet Control Message Protocol |

Nota:

Se o ICMP apenas identifica o tipo de pacote, não temos então muita informação útil. Mas felizmente, ele nos dá um número de código também, como o Código: 1 (host unreachable). (Vários tipos de pacotes ICMP oferecem códigos com um pouco de informação mais específica sobre o pacote.) Observe que o endereço IP de origem no pacote 2 não é o computador que está sendo pingado. Este é um sinal certo de que sua solicitação de eco não veio do seu destino.

O código de ICMP listado (1) nos diz que a solicitação de ping chegou ao roteador ou switch, mas não ao host de destino. Quando um host está inacessível, você também vai ver muitas vezes um broadcast ARP enviado a partir do roteador ou switch. A falta de resposta a este broadcast ARP significa que o dispositivo de envio não pode encontrar o dispositivo de destino, então ele envia um pacote de volta para o computador de origem com um ICMP tipo 0, código 1.

Porta não encontrada (Unreachable Port)

Outra tarefa comum quando estamos resolvendo um problema pingando um dispositivo em uma determinada porta. Isso geralmente é feito para garantir que as portas que são necessárias para a execução de determinados serviços estão abertas e aceitam a comunicação de entrada.

Por exemplo, você pode garantir que o FTP é acessível pingando um computador remoto na porta 21. Se por algum motivo o computador remoto não está aceitando a comunicação de entrada na porta 21, ele irá retornar um pacote ICMP tipo 0, o código 2, o que significa que a porta de destino está inacessível.

Desde que você utilize o ICMP com freqüência em sua rede no dia-a-dia em sua rotina de manutenção, você deve se familiarizar com ele e alguns dos seus tipos e códigos mais comuns.

Pacotes Fragmentados

O Internet Protocol é utilizado para uma volumosa transferência de dados através de uma rede, mas nós muitas vezes ignoramos o fato de que é somente assim que uma grande quantidade de dados pode caber no fio de uma vez. Com o intuito de solucionar as limitações das camadas inferiores, O IP possui uma tecnologia chamada de fragmentação. A fragmentação permite que o protocolo IP possa quebrar grandes quantidades de dados em blocos que podem ser enviados através da camada física e remontados no sistema destino.

Nesta seção, vamos dar uma olhada neste fluxo de dados que tem sido fragmentado pelo IP.

O arquivo de traçado ipfragments.pcap consiste em 24 pacotes que mostram um solicitação ping e sua resposta. De nossa experiência anterior, sabemos que uma típica Seqüência ICMP (ping) solicitação-e-resposta só têm oito pacotes. Então por que nós temos muitos mais aqui? Porque neste caso cada pedido e cada resposta exigem três pacotes em vez de apenas um, por isso há três vezes mais pacotes do que o usual, como você pode ver na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|--|
| 1 | 0.000000 | 192.168.0.114 | 192.168.0.193 | IP | Fragmented IP protocol (proto=ICMP 0x01, off=0) |
| 2 | 0.000085 | 192.168.0.114 | 192.168.0.193 | IP | Fragmented IP protocol (proto=ICMP 0x01, off=1480) |
| 3 | 0.000094 | 192.168.0.114 | 192.168.0.193 | ICMP | Echo (ping) request |
| 4 | 0.004244 | 192.168.0.193 | 192.168.0.114 | IP | Fragmented IP protocol (proto=ICMP 0x01, off=0) |
| 5 | 0.004545 | 192.168.0.193 | 192.168.0.114 | IP | Fragmented IP protocol (proto=ICMP 0x01, off=1480) |
| 6 | 0.004623 | 192.168.0.193 | 192.168.0.114 | ICMP | Echo (ping) reply |

Estes são os pacotes que você veria se capturasse um ping cujo tamanho dos dados fosse maior do que o padrão. Por padrão, um ping envia apenas 32 bytes de dados para o seu destino no Windows. No entanto, como você pode ver, o ping neste traçado está transmitindo 3072 bytes de dados para o cliente. Isso representa um problema, porque o padrão Ethernet só é projetado para lidar com 1.500 bytes em um único pacote. Portanto, O IP deve fragmentar os pacotes em um fluxo de dados, que é o que vemos neste arquivo de rastreamento.

Determinando Quando um Pacote é Fragmentado

Como podemos dizer se um pacote foi fragmentado? Felizmente, tudo o que precisamos fazer é olhar para o painel **Packet Details** no ipfragments.pcap. Aqui está como fazê-lo:

1. No arquivo de captura, selecione o pacote 1, expanda a seção **Internet Protocol** na parte inferior do painel **Packet Details**.
2. Você deverá ver uma seção chamada Flags. Expanda esta seção e você deverá ver três campos de dados, como mostrado na figura abaixo. O que é mais nos interessa é a seção **More Fragments**. Observe que para este pacote, esta seção tem um valor de 1, isso significa que existe mais fragmentos na sequência.

```
Internet Protocol, Src: 192.168.0.114 (192.168.0.114), Dst: 192.168.0.193 (192.168.0.193)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 1500
Identification: 0x61d1 (25041)
Flags: 0x02 (More Fragments)
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..1. = More fragments: Set
Fragment offset: 1480
Time to live: 128
Protocol: ICMP (0x01)
Header checksum: 0x3013 [correct]
Source: 192.168.0.114 (192.168.0.114)
Destination: 192.168.0.193 (192.168.0.193)
Reassembled IP in frame: 3
Data (1480 bytes)
```

3. Olhe para a mesma seção do pacote 2, você verá que ele tem o mesmo valor no campo **More Fragments**.
4. Olhe para o campo **More Fragments** do pacote 3, mostrado na figura abaixo. Ao contrário dos pacotes 1 e 2, este pacote tem um 0 no campo **More Fragments**. Um valor de 0 nos diz que este

pacote é o fim do fluxo de dados e que não há mais fragmentos na seqüência. Os possíveis valores para esse campo e só 1 e 0.

```

Internet Protocol, Src: 192.168.0.114 (192.168.0.114), Dst: 192.168.0.193 (192.168.0.193)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 140
    Identification: 0x61d1 (25041)
  Flags: 0x00
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..0. = More fragments: Not set
  Fragment offset: 2960
  Time to live: 128
  Protocol: ICMP (0x01)
  Header checksum: 0x54aa [correct]
  Source: 192.168.0.114 (192.168.0.114)
  Destination: 192.168.0.193 (192.168.0.193)
  [IP Fragments (3080 bytes): #1(1480), #2(1480), #3(120)]
Internet Control Message Protocol

```

Mantendo as Coisas em Ordem

A próxima pergunta que surge é como estes pacotes fragmentados mantêm a ordem. Uma vez que um dispositivo pode receber múltiplos fluxos de dados de uma só vez, o IP atribui um valor de seqüência para que os sistemas receptores possam saber a ordem dos pacotes fragmentados.

Para ver o valor da seqüência de um pacote fragmentado, olhe na seção **IP** do painel **Packet Details**. Por exemplo, se olhar a seção **IP** do pacote 1 um no arquivo de exemplo, você verá um valor de seqüência 0. Isto diz que este é o primeiro pacote de uma série de pacotes fragmentados.

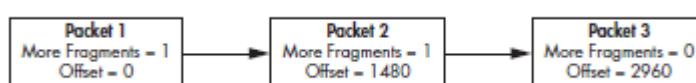
Se for para o segundo pacote, você vai ver uma mudança dramática neste número (figura abaixo): ele sobe para 1.480. A razão para esta mudança é que o valor de seqüência de todos os fragmentos do pacote seguinte é determinado pelo tamanho (dados) do pacote anterior (menos o tamanho do cabeçalho IP, que é de 20 bytes). No caso do pacote 2, esse pacote terá o número da seqüência anterior, que é 0, e é adicionado o tamanho (em bytes) do mesmo, que é 1480.

```

Internet Protocol, Src: 192.168.0.114 (192.168.0.114), Dst: 192.168.0.193 (192.168.0.193)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 1500
    Identification: 0x61d1 (25041)
  Flags: 0x02 (More Fragments)
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..1. = More fragments: Set
  Fragment offset: 1480
  Time to live: 128
  Protocol: ICMP (0x01)
  Header checksum: 0x3013 [correct]
  Source: 192.168.0.114 (192.168.0.114)
  Destination: 192.168.0.193 (192.168.0.193)
  Reassembled IP in frame: 3
Data (1480 bytes)

```

Como o pacote de 2, o pacote 3 tem a seqüência de 1480 acrescentado ao tamanho do pacote anterior que é 1480, resultando em uma nova seqüência de 2960. Este conceito é ilustrado na figura abaixo.



Dê uma olhada em alguns exemplos de tráfego IP fragmentado para ver se você pode seguir um fluxo de dados até o fim e manter esse fluxo em ordem usando a seqüência de cada pacote. (Isto pode vir a ser um desafio maior do que você pensa em um arquivo de captura desordenada.)

Sem Conectividade

Agora vamos usar o Wireshark, pela primeira vez para analisar e solucionar problemas de um problema real de rede. Neste cenário temos dois usuários, Barry e Beth, que se sentam ao lado um do outro em um escritório. Depois de um aumento do orçamento o departamento de TI acaba de adquirir dois novos computadores para Barry e Beth. Você está no comando da instalação destes novos computadores e certificando-se que estão funcionando corretamente. Após a instalação, conexão e configuração dos computadores, você começa a testá-los para se certificar que tudo está funcionando. No entanto, você se depara rapidamente com um problema. O computador de Barry está funcionando perfeitamente, mas por alguma razão o computador de Beth é incapaz de acessar a Internet. Seu objetivo é descobrir o porquê o computador de Beth é incapaz de se conectar à Internet e, em seguida, resolver o problema.

O Que Sabemos

A primeira coisa que você deve sempre fazer quanto a resolução de um problema é fazer uma lista do que você sabe sobre o assunto. Neste caso, sabemos que Barry e Beth ambos estão usando computadores idênticos, novíssimos. Sabemos também ambos os computadores têm ligação à rede, pois você lhes atribuiu endereços IP e teve a certeza de que você poderia pingá-los de outro computador no mesmo segmento de rede. Finalmente, sabemos que as configurações em ambos os computadores devem ser exatamente as mesmas, uma vez que você os configurou um após o outro.

Farejando Através dos Fios

Uma vez que nós estabelecemos o que sabemos sobre o assunto, é hora de fazermos um plano para descobrir o que não sabemos. Começamos por descobrir que tipo de captura de tráfego é preciso ter e onde é preciso colocar o nosso analisador de rede para obtê-las.

O problema a resolver é o do acesso a Internet, a escolha lógica é capturar os pacotes quando o computador de Beth está tentando acessar um site. A rede a qual o computador de Barry e Beth está conectado, não é muito familiar, assim para efeito de comparação, vamos capturar os pacotes do computador de Barry. Nós vamos trabalhar com dois arquivos de captura: um que está ok e outro que não. Comparar os dois nos ajudará a resolver o problema. Este processo é conhecido como linha de base. Nós vamos instalar o Wireshark diretamente em ambos as máquinas.

Análise

Vamos começar por olhar o arquivo de rastreamento do computador de Barry está acessando a Internet com sucesso (`barryscomputer.pcap`). Quando você abre o arquivo de rastreamento, a primeira coisa que você vai ver é uma transação HTTP.

Como você pode ver na figura abaixo, você primeiro tem broadcast ARP procurando um endereço de camada 2 do gateway default 192.168.0.10. Depois do computador de Barry receber uma resposta a esta solicitação, ele inicia um handshake TCP com o servidor remoto. Quando isso for concluído, a transferência de dados do servidor para o cliente começa.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------------|--------------------|----------|--|
| 1 | 0.000000 | Microsoft_2a:45:d2 | Broadcast | ARP | who has 192.168.0.10? Tell 192.168.0.183 |
| 2 | 0.002196 | D-Link_21:99:4c | Microsoft_2a:45:d2 | ARP | 192.168.0.10 is at 00:05:5d:21:99:4c |
| 3 | 0.002259 | 192.168.0.183 | 64.233.161.104 | TCP | 1125 > http [SYN] Seq=0 Len=0 MSS=1460 |
| 4 | 0.054708 | 64.233.161.104 | 192.168.0.183 | TCP | http > 1125 [SYN, ACK] Seq=0 Ack=1 win=8190 Len=0 MSS=1452 |
| 5 | 0.054871 | 192.168.0.183 | 64.233.161.104 | TCP | 1125 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0 |
| 6 | 0.055737 | 192.168.0.183 | 64.233.161.104 | HTTP | GET / HTTP/1.1 |
| 7 | 0.103969 | 64.233.161.104 | 192.168.0.183 | TCP | http > 1125 [ACK] Seq=1 Ack=284 Win=6432 Len=0 |
| 8 | 0.158478 | 64.233.161.104 | 192.168.0.183 | TCP | [TCP segment of a reassembled PDU] |
| 9 | 0.161865 | 64.233.161.104 | 192.168.0.183 | HTTP | HTTP/1.1 200 OK (text/html) |

Agora que sabemos como um pedido WEB bem sucedido deve nesta rede, vamos dar uma olhada no arquivo de captura do computador de Beth (`Bethscomputer.pcap`) para ver se conseguimos encontrar o problema. Não devemos demorar muito tempo para percebermos que algo está definitivamente errado aqui. Conforme mostrado na figura abaixo, o primeiro pacote é um pedido ARP, não muito diferente daquele do arquivo de Barry `barryscomputer.pcap`. No entanto, este pedido ARP não é enviado para o mesmo endereço IP como no último. Aqui, o ARP está procurando por um dispositivo com um endereço IP de 192.168.0.11.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|-------------------|-------------|----------|--|
| 1 | 0.000000 | Microsof_2a:45:d2 | Broadcast | ARP | who has 192.168.0.11? tell 192.168.0.122 |

Imediatamente após esse pacote ARP, vemos um monte de tráfego NetBIOS, como mostrado na figura abaixo. Se outros endereços IP não é sinal de que algo é errado, então tudo isso é definitivamente o tráfego NetBIOS.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|---------------|---------------|----------|---|
| 2 | 1.271630 | 192.168.0.122 | 192.168.0.255 | BROWSE | Request Announcement TESLA-MARKETING |
| 3 | 2.424840 | 192.168.0.122 | 192.168.0.255 | BROWSE | Host Announcement TESLA-MARKETING, Workstation, Server, NT Workstation, Potential |
| 4 | 2.425448 | 192.168.0.122 | 192.168.0.255 | BROWSE | Host Announcement TESLA-MARKETING, Workstation, Server, NT Workstation, Potential |

O NetBIOS é um protocolo mais antigo que normalmente só é usado como um backup quando o TCP/IP não está funcionando. O aparecimento de tráfego NetBIOS aqui significa que, o computador de Beth foi incapaz de se conectar com êxito à Internet com o TCP/IP, e foi revertido para o NetBIOS como meio alternativo de comunicação, mas que também falhou. (A qualquer momento que você vê o NetBIOS sua rede, muitas vezes é um bom sinal de que algo não está certo.)

Vamos nos concentrar na maior anomalia que visto até agora, ou seja, os diferentes endereços IP em cada um dos pacotes ARP. O computador de Barry usa o ARP para encontrar o local do gateway default 192.168.0.10. O computador de Beth tentou fazer a mesma coisa, porém, tentou encontrar a localização do endereço IP 192.168.0.11 e falhou, como mostrado na figura abaixo. Os endereços do gateway default são inconsistentes; algo está errado.

Computador de Barry

```
Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: 00:03:ff:2a:45:d2 (00:03:ff:2a:45:d2)
Sender IP address: 192.168.0.183 (192.168.0.183)
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.0.10 (192.168.0.10)
```

Computador de Beth

```
Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: 00:03:ff:2a:45:d2 (00:03:ff:2a:45:d2)
Sender IP address: 192.168.0.122 (192.168.0.122)
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.0.11 (192.168.0.11)
```

Uma verificação rápida das configurações TCP/IP em ambos os computadores revela a resposta do nosso problema: um erro de digitação. O computador de Barry está definido para ter um gateway default 192.168.0.10, e o computador de Beth está definido para 192.168.0.11, que é o endereço errado.

Resumo

Os erros que você vai encontrar é muitas vezes devido a erros de configuração. Sempre que possível, compare uma máquina que funciona adequadamente com a que não para você identificar o problema. No cenário anterior, pudemos localizar o pacote exato em que as coisas não se encaixavam corretamente. Uma vez definida a causa do seu problema, você irá resolvê-lo mais facilmente.

Fantasma no Internet Explorer

Este cenário começa com uma chamada perturbadora para o help desk de um usuário em sua rede chamado Chad. De acordo com o Chad, o seu computador recentemente hospedeiro de uma possessão demoníaca. Apesar de seus melhores esforços, a home page inicial em seu navegador da Internet continua a mudar sozinha apontando para outro site. Mesmo que ele mude manualmente de volta como deveria ser, suas mudanças são revertidas depois que ele reinicia o seu computador. Seu objetivo aqui é chegar ao fundo dessa possessão e de realizar um exorcismo dos fantasmas que invadiram o computador do Chade.

O Que Sabemos

Chad foi da nossa empresa há muito tempo e sabemos que ele não tem uma grande competência técnica. Na verdade, ele geralmente faz mau uso do que bom uso do seu computador. (Eu não suponho que você conheça usuários como ele?) Do ponto de vista técnico, sabemos que o computador de Chad tem cerca de dois anos, roda o sistema Windows XP, e usa Internet Explorer 6 como seu navegador.

Farejando Através dos Fios

Porque esse problema ocorre apenas como o computador do Chad, sabemos que a única máquina que temos para capturar pacotes é a do Chad. Além disso, parece que a página inicial do computador do Chad é redefinida toda vez seu computador é reiniciado, vamos fazer a nossa captura no momento do boot.

Neste caso, não podemos instalar o Wireshark diretamente na máquina do Chad e capturar os pacotes que precisamos, o modo hubbing é um bom método para usarmos. Se você não lembra como esta técnica é usada, consulte nosso a discussão sobre ela em "Hubbing" na página 19. A captura começará assim que o computador for ligado e será interrompida logo que ele for completamente reiniciado, sem a necessidade da interação do usuário.

Análise

Embora não haja nenhuma interação do usuário com o computador durante a captura, você pode ficar um pouco chocado quando você abre o arquivo de rastreamento (hauntedbrowser.pcap) e ver os pacotes TCP e HTTP bombardeando através dos fios, como mostrado na figura abaixo. Durante o processo de inicialização normal, você deve raramente, ou nunca, ver pacotes como estes sendo enviados.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|----------------|----------|--|
| 1 | 0.000000 | 192.168.0.184 | 24.46.230.187 | TCP | 1038 > 1706 [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 0.000019 | 192.168.0.184 | 69.206.254.66 | TCP | 1039 > 3531 [SYN] Seq=0 Len=0 MSS=1460 |
| 3 | 0.001354 | 192.168.0.184 | 24.46.230.187 | TCP | 1038 > 1706 [SYN] Seq=0 Len=0 MSS=1460 |
| 4 | 0.002375 | 192.168.0.184 | 69.206.254.66 | TCP | 1039 > 3531 [SYN] Seq=0 Len=0 MSS=1460 |
| 5 | 0.338822 | 192.168.0.184 | 64.124.109.200 | HTTP | GET /command/Commandv6.07.asp?Key=&t=26962 HTTP/1.1 |
| 6 | 0.340546 | 192.168.0.184 | 64.124.109.200 | HTTP | [TCP out-of-order] GET /command/Commandv6.07.asp?Key=&t=26962 HTTP/1.1 |
| 7 | 0.638241 | 192.168.0.184 | 64.124.109.200 | TCP | 1040 > http [ACK] Seq=286 Ack=243 win=65041 Len=0 |
| 8 | 0.638386 | 192.168.0.184 | 64.124.109.200 | TCP | [TCP Dup ACK 7#1] 1040 > http [ACK] Seq=286 Ack=243 win=65041 Len=0 |
| 9 | 0.800253 | 192.168.0.184 | 64.124.109.200 | TCP | 1040 > http [ACK] Seq=286 Ack=613 win=64671 Len=0 |
| 10 | 0.800403 | 192.168.0.184 | 64.124.109.200 | TCP | [TCP Dup ACK 9#1] 1040 > http [ACK] Seq=286 Ack=613 win=64671 Len=0 |

Olhando mais de perto esses pacotes, podemos imediatamente tirar algumas conclusões. Em primeiro lugar, sabemos que todos esses pedidos HTTP estão sendo gerados pelo computador do Chade, porque o seu endereço IP é listado como a fonte de todos os pacotes TCP e HTTP. Além disso, você pode ver no pacote 5 (figura abaixo) que este computador está enviando pacotes HTTP para um sistema na Internet com o comando GET, o que significa que ele está tentando fazer o download de dados.

```
HTTP/1.1\r\n
GET /command/Commandv6.07.asp?Key=&t=26962\r\n
Request Method: GET
Request URI: /command/Commandv6.07.asp?Key=&t=26962
Request Version: HTTP/1.1
User-Agent: Mozilla/3.0 (compatible; MSIE 4.0; Win32)\r\n
Host: command.weatherbug.com\r\n
Connection: Keep-Alive\r\n
Cookie: wxbug_cookie=has_cookies=1; RMID=4aecf9dc45a025d0; RMFD=011H3KJTO104ym|o1058k; RMFS=011H3KHLU1052U; LMB1per12h=1\r\n
```

Dada essa informação, podemos supor que algo está sendo executado no computador do Chad na inicialização que não deveria ocorrer. Um olhar o painel de baixo do Packet List fornece-nos uma idéia do que pode ser a raiz do nosso problema. Os pacotes 11 e 12 fazem um pedido DNS ao servidor no domínio na weatherbug.com, como mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|--|
| 11 | 3.725242 | 192.168.0.184 | 205.152.37.23 | DNS | Standard query A deskwx.weatherbug.com |
| 12 | 3.734060 | 192.168.0.184 | 205.152.37.23 | DNS | Standard query A deskwx.weatherbug.com |

Considerando que a página inicial do Chad continua a mudar quando seu computador é reiniciado, nós provavelmente encontramos o nosso culpado. Após uma investigação mais aprofundada do computador do Chad, nossa hipótese se mostra correta, e vemos que o computador tem um programa desktop WeatherBug sendo executado em segundo plano, configurado para transferir as informações de tempo e exibi-las na home page após cada reinicialização. Após a desinstalação deste software, o problema cessa.

Resumo

Você vai descobrir que muitos problemas em computadores e redes não são da responsabilidade de um computador particular ou da própria rede, mas sim o software que está rodando nele. Neste cenário, um programa de monitoramento do tempo, havia sido instalado no computador do Chade , causando-lhe o pensamento que o mesmo estava "possuído" por seu navegador da web mudando a sua home page inicial após cada reinicialização. Ao capturar e analisar os pacotes com o Wireshark, fomos capazes de descobrir este programa rodando silenciosamente em segundo plano.

Problemas com o FTP

Neste seguinte cenário, vamos imaginar que você acabou de configurar um novo servidor FTP para sua companhia. Os clientes irão se conectar a esse servidor FTP tanto internamente e externamente para fazer download e upload de grandes quantidades de dados. Você definiu o software do servidor FTP e, criou um nome de usuário e senha genérica para utilização por todos os funcionários. No entanto, por alguma razão quando você está tentando testar o servidor de um computador remoto, você não consegue acessá-lo através software cliente de FTP.

O Que Sabemos

Sabemos que este servidor é novo e acaba de ser criada utilizando o Windows Server 2003, com todas as últimas atualizações e service packs instalados. Verificamos que o software de FTP está configurado corretamente e está ativo. Verificamos também que o cliente que tenta se conectar ao servidor de FTP está usando o endereço IP apropriado e as credenciais de login.

Farejando Através dos Fios

Porque esse problema envolve um servidor e uma máquina cliente, nós teremos um arquivo de captura de ambos os computadores. A captura do cliente será feita quando o software de cliente FTP tenta se conectar ao servidor. A captura a partir do servidor será feita no momento em que o cliente está tentando se conectar ao software de FTP. Ao capturar os arquivos desta forma, seremos capazes de determinar se o problema tem origem junto do cliente ou ao servidor, em seguida, poderemos prosseguir com uma investigação mais aprofundada. Nós vamos instalar o Wireshark diretamente nestas duas máquinas para o propósito destas capturas.

Análise

Vamos começar com o cliente para se certificar de que a comunicação está sendo iniciada como deve ser. Olhando para o ftpclientdenied.pcap arquivo de captura (figura abaixo), e ver que ele está fazendo exatamente o que deve fazer. Começa o processo handshake TCP através da emissão de um pacote SYN para o servidor remoto, 192.168.0.182. No entanto, o servidor não responde, de modo que o cliente emite mais dois pacotes SYN para tentar estabelecer uma comunicação.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|---------------------------------------|
| 1 | 0.000000 | 192.168.0.193 | 192.168.0.182 | TCP | 1596 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 2.944417 | 192.168.0.193 | 192.168.0.182 | TCP | 1596 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 3 | 8.979791 | 192.168.0.193 | 192.168.0.182 | TCP | 1596 > ftp [SYN] Seq=0 Len=0 MSS=1460 |

Este processo continua por cerca de nove segundos antes que o cliente determine que é incapaz de se conectar ao servidor. O cliente está fazendo exatamente o que é suposto fazer sobre o início do handshake TCP, por isso é seguro assumir que o problema mais provável não resida no cliente.

Agora vamos olhar para o rastreamento do ponto de vista do servidor no arquivo de captura ftpserverdenied.pcap. Os dois arquivos de captura parecem-se similar, na verdade, a única diferença entre os dois arquivos é que a fonte e os endereços de destino nos pacotes SYN estão sendo comutados (figura abaixo). Isso nos diz que os pacotes enviados do cliente são realmente enviados ao servidor, mas que por alguma razão desconhecida o servidor não os tem aceitados.

Cliente

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|---------------------------------------|
| 1 | 0.000000 | 192.168.0.193 | 192.168.0.182 | TCP | 1596 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 2.944417 | 192.168.0.193 | 192.168.0.182 | TCP | 1596 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 3 | 6.035374 | 192.168.0.193 | 192.168.0.182 | TCP | 1596 > ftp [SYN] Seq=0 Len=0 MSS=1460 |

Servidor

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|---------------------------------------|
| 1 | 0.000000 | 192.168.0.193 | 192.168.0.182 | TCP | 1637 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 2.992575 | 192.168.0.193 | 192.168.0.182 | TCP | 1637 > ftp [SYN] Seq=0 Len=0 MSS=1460 |
| 3 | 6.035158 | 192.168.0.193 | 192.168.0.182 | TCP | 1637 > ftp [SYN] Seq=0 Len=0 MSS=1460 |

Há tipicamente três razões principais para que o computador rejeite os pacotes que lhe foram enviados.

- O suposto serviço que é para aceitar esses pacotes não está funcionando. Porque sabemos que o software do servidor FTP está funcionando e pronto para aceitar conexões, esse não pode ser o nosso problema.
- O servidor está enfrentando uma quantidade muito elevada de tráfego. Nestas situações o buffer do servidor pode estar muito cheio e ficando incapaz de se comunicar com alguns clientes. Mais uma vez, isso não pode ser a causa do nosso problema, porque o servidor acaba de ser criado e está sem carga alguma.
- Os pacotes estão sendo intencionalmente bloqueados. O que intencionalmente pode bloquear os pacotes recebidos por um computador? Algo está fazendo exatamente o que é suposto fazer! Após uma análise mais aprofundada do servidor, nós achamos que o firewall do Windows está ativado e ele está bloqueando todo o tráfego de entrada nas portas FTP.

Resumo

A análise de pacotes nem sempre leva você direto a causa do seu problema. De fato, neste cenário nada específico na captura identificou o firewall como o problema. No entanto, esta análise permitiu reduzir o problema especificamente ao servidor.

Às vezes, você deve solucionar problemas que afetam dezenas ou mesmo centenas de sistemas. Se você pode usar a análise de pacotes nestas situações afim de diminuir o problema em um computador específico, então você terá que ter guardado uma enorme quantidade de tempo.

Falha no Servidor Remoto

Alguns usuários da rede são absolutamente inacreditáveis. Alguma vez você já teve um usuário que sempre culpa o departamento de TI para cada pequeno problema que ele ou ela tem? Erin é exatamente esse tipo de usuário. A qualquer momento em que a rede esteja funcionando abaixo do estado ideal esperado, ela fica contente que você saiba.

Neste cenário específico, Erin está tentando enviar um pedido online de alguns produtos. O problema é que quando ela submete o formulário para encomendar o produto, ele retorna um erro HTTP 403 (proibido). Nós sabemos que este erro é quase sempre causado por um problema no website remoto, mas Erin se queixou o suficiente para que o seu patrão lhe pedisse para provar-lhe que este é verdadeiramente o caso. Temos que mostrar que esse problema é devido ao servidor remoto, e não algo no nível de pacote.

O Que Sabemos

Sabemos que Erin não tem sido capaz de enviar dados através do formulário web em questão, mas ela pode submeter qualquer outro formulário web que ela necessite alcançar. Olhando para o código-fonte do site em questão, vemos que ela está usando um formulário HTML padrão, sem nada diferente anexado a ele.

Farejando Através dos Fios

Instalar o Wireshark no computador de Erin é a maneira mais fácil de obter a captura do arquivo de que precisamos. Uma vez instalado, podemos começar o processo de captura e Erin pode então tentar preencher e enviar o seu formulário, é neste ponto que começaremos o processo de análise.

Análise

O arquivo de rastreamento (http-culpa post.pcap) começa com um handshake TCP padrão entre o computador de Erin, 24.4.97.251, e o servidor remoto, 216.23.168.114, como mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|----------------|----------------|----------|---|
| 1 | 0.000000 | 24.4.97.251 | 216.23.168.114 | TCP | 2580 > http [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 0.027956 | 216.23.168.114 | 24.4.97.251 | TCP | http > 2579 [SYN, ACK] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 |
| 3 | 0.028045 | 24.4.97.251 | 216.23.168.114 | TCP | 2579 > http [ACK] Seq=0 Ack=1 Win=65535 Len=0 |

Pouco tempo depois, começa a comunicação HTTP entre o cliente e o servidor. Observe na coluna **Info** (figura abaixo) que não demora muito para o cliente receber a mensagem HTTP 403 do servidor, que é a fonte da denúncia.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|----------------|-------------|----------|------------------------------------|
| 9 | 0.065496 | 216.23.168.114 | 24.4.97.251 | HTTP | HTTP/1.1 403 Forbidden (text/html) |

O erro 403 acontece no pacote 9. Porque este é realmente é único fluxo de dados na captura que estamos preocupados, clique com o botão direito e escolha **Follow TCP Stream** para ler o texto com a remontagem da transação HTTP, como mostrado na figura abaixo.

```
HTTP/1.1 403 Forbidden
Content-Length: 1758
Content-Type: text/html
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Mon, 27 Nov 2006 06:31:02 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>The page cannot be displayed</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=Windows-1252">
<STYLE type="text/css">
    BODY { font: 8pt/12pt verdana }
    H1 { font: 13pt/15pt verdana }
    H2 { font: 8pt/12pt verdana }
    A:link { color: red }
    A:visited { color: maroon }
</STYLE>
</HEAD><BODY><TABLE width=500 border=0 cellspacing=10><TR><TD>

<h1>The page cannot be displayed</h1>
You have attempted to execute a CGI, ISAPI, or other executable program from a
directory that does not allow programs to be executed.
<hr>
<p>Please try the following:</p>
<ul>
<li>Contact the web site administrator if you believe this directory should allow
execute access.</li>
</ul>
<h2>HTTP Error 403.1 - Forbidden: Execute access is denied.<br>Internet Information
Services (IIS)</h2>
<hr>
<p>Technical information (for support personnel)</p>

```

Save As | Print | Entire conversation (3287 bytes) | ▾ | ASCII EBCDIC Hex Dump C Arrays Raw

Close | Filter Out This Stream

Olhando para este fluxo TCP, vemos primeiro os dados do formulário sendo enviados a partir de nosso cliente para o servidor. Neste ponto, devemos ver uma resposta do servidor com algo dizendo que os dados do formulário foram aceitos, mas em vez disso, vemos a resposta de 403. Isso é suficiente para provar que o problema encontra-se no servidor remoto e não na sua rede.

Resumo

A análise de pacotes geralmente é uma grande ferramenta para usar quando você tem de provar o que você realmente está fazendo. Não só às vezes você tem que provar suas convicções ao seu gerente, mas às vezes provar a você mesmo.

Neste caso, a interpretação simples do texto do fluxo TCP pode ser mostrada ao seu supervisor para pôr fim ao discurso de Erin contra o departamento de TI.

Um Programa Mal Intencionado

Este cenário é muito parecido com a situação do computador assombrado do Chad. Neste caso, porém, temos um pouco mais a passar. Mandy é outro usuário em nossa rede que está se queixando de coisas estranhas acontecendo em seu navegador. O navegador continua a mudar a sua home page para uma de um site de segurança falso em momentos aleatórios ao longo do dia. Não só isso, ele vê um punhado de pop-ups e seu computador é geralmente lento.

Se você tem alguma experiência de reparo em computador, provavelmente você está muito certo que este é um problema de spyware. No entanto, ao invés de apenas usar uma ferramenta de remoção de spyware, nós vamos fazer um rastreamento do computador, de modo que possamos ver exatamente o que este spyware está fazendo para o computador de Mandy ter tantos problemas.

O Que Sabemos

Nós não precisamos saber muito para resolver este problema particular. Sabemos que o computador de Mandy está funcionando lentamente e que o seu browser está sendo invadido constantemente. Seu computador está executando um software antivírus, de modo que o vírus não seja uma preocupação para nós.

Farejando Através dos Fios

Ao solucionar um problema relacionado a spyware, é sempre uma boa idéia começar o seu arquivo de rastreamento quando o computador é reiniciado. A maioria das aplicações spyware tendem a verificar por atualizações e infecta o computador na reinicialização.

Vamos começar nosso arquivo de captura, logo que o computador for ligado e continuar a capturar os pacotes até um minuto ou mais após o processo de inicialização ter sido concluído. Neste caso, o hubbing ou envenenamento de cache ARP são os melhores métodos a utilizar para interceptar os pacotes desta máquina. Há muito tráfego fluindo em nossa rede, vamos criar o arquivo de captura usando um filtro que só pega o tráfego de/para o computador de Mandy.

Análise

Este é um arquivo de captura muito grande, então vamos começar pelo começo. Os dois primeiros pacotes (mostrado na figura abaixo) são bastante comuns quando um computador inicia-se e começa a inicializar sua pilha TCP/IP.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|----------------|-----------------|----------|--|
| 1 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID 0x9a71fd19 |
| 2 | 0.210908 | Intel_b7:f2:f5 | Broadcast | ARP | who has 24.6.125.19? Gratuitous ARP |

O primeiro pacote mostra que o computador faz uma solicitação de IP ao servidor DHCP. Normalmente, há uma resposta para este pacote a partir do servidor DHCP, mas uma vez que este é um pacote broadcast, o nosso filtro de captura não permitirá que ele seja mostrado.

O segundo pacote é um pacote ARP que chamamos de ARP gratuito. O ARP gratuito é um pacote broadcast ARP que é usado para garantir que nenhuma outra máquina na rede tem o mesmo endereço IP que máquina de origem. Você só deve ver os pedidos ARP gratuito saindo, se você vê uma resposta ARP gratuito, o que significa que outro computador na rede tem o seu endereço IP. Nesta captura vemos apenas os pedidos, então estamos em boa forma.

O terceiro pacote da captura é o que nós nos preocuparemos. Neste momento do processo de inicialização do computador, o TCP/IP ainda não foi totalmente inicializado: Você pode ver que ele ainda envia os seus pacotes ARP gratuito, como mostrado na figura abaixo. Mas o pacote 3 mostra que um dispositivo fora da nossa rede está tentando se comunicar com o computador de Mandy na porta 5554.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|----------------|-----------------|----------|--|
| 1 | 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID 0x9a71fd19 |
| 2 | 0.210908 | Intel_b7:f2:f5 | Broadcast | ARP | who has 24.6.125.19? Gratuitous ARP |
| 3 | 0.727104 | 67.70.67.186 | 24.6.125.19 | TCP | 2431 > 5554 [SYN] Seq=0 Len=0 MSS=1440 |
| 4 | 0.020358 | Intel_b7:f2:f5 | Broadcast | ARP | who has 24.6.125.19? Gratuitous ARP |
| 5 | 0.796948 | 67.70.67.186 | 24.6.125.19 | TCP | 2860 > 9898 [SYN] Seq=0 Len=0 MSS=1440 |
| 6 | 0.204513 | Intel_b7:f2:f5 | Broadcast | ARP | who has 24.6.125.19? Gratuitous ARP |

Neste ponto, do processo de inicialização, nenhuma máquina deve estar tentando se comunicar com o computador de Mandy, uma vez que a mesma ainda não está pronta a se comunicar. Portanto, o computador de Mandy simplesmente descarta o pacote e continua com seu processo de inicialização. Outro pacote como este aparece no pacote 5 do arquivo de captura, porém desta vez, o pacote alterou a porta que ele está usando e tenta se conectar a porta 9898, como mostrado na figura abaixo. Muito complicado.

| |
|---|
| ■ Transmission Control Protocol, Src Port: 2860 (2860), Dst Port: 9898 (9898), Seq: 0, Len: 0 |
| Source port: 2860 (2860) |
| Destination port: 9898 (9898) |
| Sequence number: 0 (relative sequence number) |
| Header length: 28 bytes |
| ■ Flags: 0x02 (SYN) |
| window size: 16384 |
| Checksum: 0x1ba8 [correct] |
| ■ Options: (8 bytes) |

Mais uma vez, o computador de Mandy não está pronto para a comunicação e simplesmente descarta o pacote.

Depois do computador de Mandy estar pronto para finalmente aceitar a comunicação, ele recebe mais um desses pacotes no pacote 10. O computador de Mandy não tem qualquer serviço sendo executado na porta requisitada para poder aceitar o handshake TCP, assim que seu computador responde para o computador remoto com um pacote TCP RST, encerra a comunicação, como mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|------------|---------------|---------------|----------|--|
| 10 | 556.468939 | 203.101.42.68 | 24.6.125.19 | TCP | 1560 > 4899 [SYN] Seq=0 Len=0 MSS=1460 |
| 12 | 556.478580 | 24.6.125.19 | 203.101.42.68 | TCP | 4899 > 1560 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 |
| 14 | 557.229229 | 203.101.42.68 | 24.6.125.19 | TCP | 1560 > 4899 [SYN] Seq=0 Len=0 MSS=1460 |
| 15 | 557.229235 | 24.6.125.19 | 203.101.42.68 | TCP | 4899 > 1560 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 |
| 16 | 557.932751 | 203.101.42.68 | 24.6.125.19 | TCP | 1560 > 4899 [SYN] Seq=0 Len=0 MSS=1460 |
| 17 | 557.932857 | 24.6.125.19 | 203.101.42.68 | TCP | 4899 > 1560 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 |

Esse processo se repete ao longo da próxima série de vários pacotes. O computador de Mandy está fazendo exatamente o que é suposto ser feito para recusar esta comunicação.

Filtrando o Que é Bom

Se você continuar a rolar até o pacote 68, você verá a primeira comunicação legítima, como mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|-------------|----------------|----------------|----------|---|
| 68 | 1132.623271 | 24.6.125.19 | 216.148.227.68 | DNS | Standard query A updatekeepalive.mcafee.com |
| 69 | 1132.658710 | 216.148.227.68 | 24.6.125.19 | DNS | Standard query response A 216.49.88.118 |

Aqui o computador de Mandy começa a se comunicar com o seu software antivírus e começa o download de suas atualizações. Estes pacotes são válidos, e já que estamos olhando apenas para os pacotes suspeitos, vamos filtrar esses por filtragem de todo o tráfego de/para o endereço IP da McAfee indicado no pacote 68 (figura abaixo).

Nota:

Esperamos que você se lembre de como criar filtros da nossa discussão anterior. O filtro que você deseja criar para esconder qualquer tipo de tráfego de/par o servidor da McAfee é **!ip.addr == 216.49.88.118**.

| | | | | |
|---------------------------------|---|---------------|-------|-------|
| Filter: !ip.addr==216.49.88.118 | ▼ | Expression... | Clear | Apply |
|---------------------------------|---|---------------|-------|-------|

Tentativas de Conexão Remota

Uma vez que você tem esse filtro configurado, o próximo pacote de nosso interesse é o pacote 147, mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|-------------|----------------|-------------|----------|-------------------------|
| 147 | 1202.816709 | 221.143.42.254 | 24.6.125.19 | Messen | NetrSendMessage request |

Este é um pacote mensageiro enviado a partir de um dispositivo na Internet. Você pode ler o conteúdo da mensagem, exibindo os pacotes através do painel **Packet Bytes**, como mostrado na figura abaixo.

| | | |
|------|---|--------------------|
| 00b0 | 00 00 d2 00 00 00 00 00 00 00 00 00 00 41 4c | AL |
| 00c0 | 45 52 54 3a 20 44 41 4e 47 45 52 4f 55 53 20 53 | ERT: DAN GEROUS S |
| 00d0 | 50 59 57 41 52 45 20 56 49 52 55 53 20 46 4f 55 | PYWARE V IRUS FOU |
| 00e0 | 4e 44 0a 44 45 4c 45 54 45 20 54 48 49 53 20 53 | ND. DELET E THIS S |
| 00f0 | 50 59 57 41 52 45 20 56 49 52 55 53 20 49 4d 4d | PYWARE V IRUS IMM |
| 0100 | 45 44 49 41 54 45 4c 59 0a 56 49 53 49 54 20 54 | EDIATELY .VISIT T |
| 0110 | 48 45 20 57 45 42 53 49 54 45 20 57 57 57 2e 50 | HE WEBSI TE WWW.P |
| 0120 | 34 55 32 2e 43 4f 4d 20 54 4f 20 43 4f 4e 54 49 | 4U2.COM TO CONTI |
| 0130 | 4e 55 45 0a 0a 4e 4f 54 45 3a 20 44 49 53 41 42 | NUE.. NOT E: DISAB |
| 0140 | 4c 49 4e 47 20 59 4f 55 52 20 22 4d 45 53 53 45 | LING YOU R "MESSE |
| 0150 | 4e 47 45 52 22 20 41 4e 44 20 22 41 4c 45 52 54 | NGER" AN D "ALERT |
| 0160 | 45 52 22 20 57 49 4e 44 4f 57 53 20 53 45 52 56 | ER" WIND OWS SERV |
| 0170 | 49 43 45 20 57 49 4c 4c 20 42 4c 4f 43 4b 20 54 | ICE WILL BLOCK T |
| 0180 | 48 45 53 45 20 4d 45 53 53 41 47 45 53 0a 0a 00 | HESE MES SAGES... |

Felizmente, o serviço de mensagens está desabilitado em nossa rede, de modo que Mandy nunca verá esta mensagem. Você pode verificar que esta mensagem nunca será entregue vendo o pacote ICMP Destination unreachable enviado pelo nosso computador ao computador remoto logo após a tentativa de conexão inicial, como mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|-------------|----------------|----------------|----------|--|
| 147 | 1202.816709 | 221.143.42.254 | 24.6.125.19 | Messen | NetrSendMessage request |
| 148 | 1202.816878 | 24.6.125.19 | 221.143.42.254 | ICMP | Destination unreachable (Port unreachable) |

No pacote 210 (figura abaixo), começamos a ver uma coisa muito preocupante.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|-------------|--------------|--------------|----------|---|
| 210 | 1617.530663 | 24.136.28.59 | 24.6.125.19 | TCP | 3092 > 1025 [SYN] Seq=0 Len=0 MSS=1460 |
| 211 | 1617.530814 | 24.6.125.19 | 24.136.28.59 | TCP | 1025 > 3092 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460 |
| 212 | 1617.534487 | 24.136.28.59 | 24.6.125.19 | TCP | 3095 > 1025 [SYN] Seq=0 Len=0 MSS=1460 |
| 213 | 1617.534608 | 24.6.125.19 | 24.136.28.59 | TCP | 1025 > 3095 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460 |
| 214 | 1617.598282 | 24.136.28.59 | 24.6.125.19 | TCP | 3099 > 3127 [SYN] Seq=0 Len=0 MSS=1460 |

Assim como antes, temos um computador remoto tentando estabelecer uma comunicação com o computador de Mandy, iniciando um handshake TCP. No entanto, ao contrário de antes, seu computador realmente responde desta vez, via porta 1025. Isso significa que existe um serviço em execução nesta porta que está aguardando por uma conexão a partir do exterior. Isso nunca é uma coisa boa!

Aproximando-se do Problema

Neste ponto, você pode ir vendo mais o nosso arquivo de captura e continuar a vendo as mesmas coisas. Várias tentativas de conexão são feitas para o computador de Mandy, algumas das quais são bem sucedidas e algumas das quais não são. Não obstante, até agora, essas tentativas de conexão realmente não resultou em muito mais interesse por nossa parte, isto é, até o número de pacotes 357, mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|-------------|----------------|----------------|------------------|--|
| 357 | 9901.421104 | 24.191.223.102 | 24.6.125.19 | DCERPC Bind: | call_id: 127 IsystemActivator v0.0 |
| 358 | 9901.421594 | 24.6.125.19 | 24.191.223.102 | DCERPC Bind_ack: | call_id: 127 Provider rejection, reason: Abstract syntax not supported |
| 360 | 9901.535034 | 24.191.223.102 | 24.6.125.19 | ISyste | RemoteCreateInstance request |
| 362 | 9901.535260 | 24.6.125.19 | 24.191.223.102 | DCERPC Fault: | call_id: 229 ctx_id: 0 status: nca_unk_if |

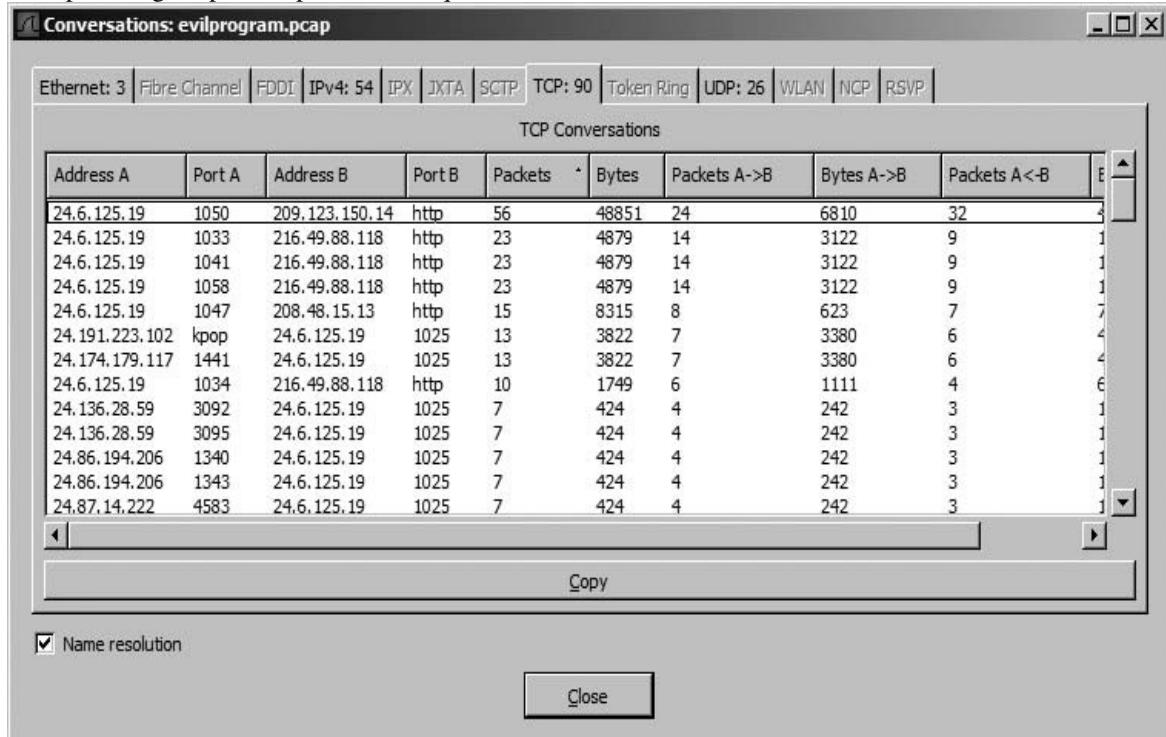
No pacote 357 é um DCEPRC, ou um pacote Remote Procedure Call (RPC). O RPC é um protocolo usado para executar programas remotamente no sistema. Vamos ver, aqui temos um computador fora da nossa rede tentando Iniciar um programa remotamente em um computador dentro de nossa rede. Não temos um PhD em ciência da computação para descobrir que isto não deveria acontecer.

Agora, vamos querer ver bem de perto o computador de Mandy para ver exatamente a comunicação de volta para o sistema remoto. Ao monitorar essa comunicação, você acabará por chegar ao pacote 381, no qual o nosso cliente faz um pedido DNS para updates.virtumonde.com, como mostrado na figura abaixo.

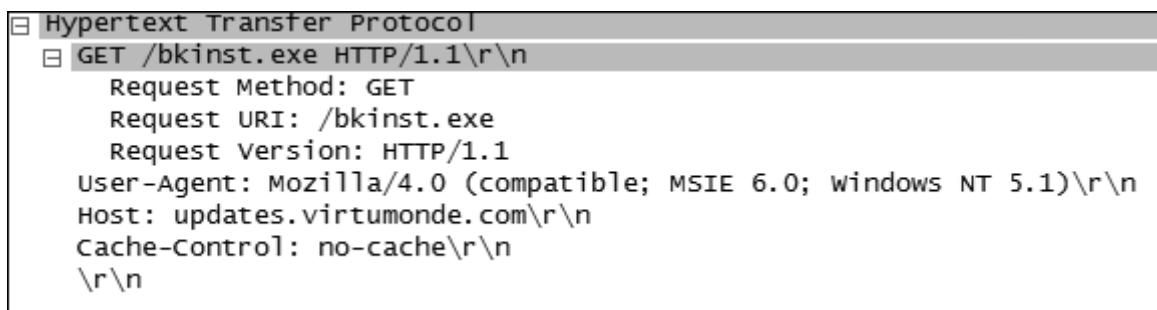
| No. | Time | Source | Destination | Protocol | Info |
|-----|-------------|----------------|----------------|----------|---|
| 381 | 11477.76286 | 24.6.125.19 | 216.148.227.68 | DNS | Standard query A updates.virtumonde.com |
| 382 | 11477.83280 | 216.148.227.68 | 24.6.125.19 | DNS | Standard query response A 208.48.15.13 A 208.48.15.11 |

Se algo como isso acontecer e você não estiver familiarizado com o site sendo consultado, faça uma pesquisa na Internet. Se você procurar a palavra-chave virtumonde, você vai encontrar um monte de resultados relacionados ao spyware e o servidor de hospedagem.

Vamos dar uma olhada na comunicação entre o computador de Mandy e o servidor remoto virtumonde. Para fazer isso, abra a janela **Conversations** e filtre todo o tráfego entre o nosso anfitrião, 24.6.125.19, e o servidor virtumonde, 208.48.15.13 (veja a figura abaixo). Uma vez que você fizer isso, você terá apenas alguns pacotes para olhar, o que torna as coisas muito mais fáceis.



Continuando a lista de pacotes, que vemos no pacote de 386 que o nosso cliente vai para o servidor virtumonde e pede o download de um arquivo chamado bkinst.exe (figura abaixo).



Se você fizer uma pesquisa na Internet para este arquivo, você verá que ele está associado com o spyware, invasor do navegador, e praticamente qualquer outra coisa ruim que você possa pensar. Você encontrou com sucesso o problema que afeta o computador de Mandy.

Resumo

Neste cenário, aprendemos que a razão pelo qual o computador de Mandy estava fazendo coisas estranhas foi relacionada a uma aplicação spyware que estava sendo descarregada em seu computador através de um serviço RPC em segundo plano. Mas qual foi o ponto positivo para passarmos por tudo isso apenas para descobrir algo que já sabíamos?

Passamos por este processo de análise para que pudéssemos entender melhor o que estava acontecendo na rede. Se o computador de Mandy era capaz de ser infectado com este spyware, as chances são grandes acontecer com o computador de outra pessoa. Aprender as portas e os serviços utilizados neste

processo de comunicação nos permitirá bloqueá-los no nível do firewall para evitar problemas no futuro. Mesmo que um problema possa parecer de simples correção, ir a até no final da análise para descobrir exatamente o porquê isso está acontecendo pode ser muito útil no futuro.

Considerações Finais

Os cenários previstos neste capítulo são muito simples, mas são também muito importantes para ajudar você a se familiarizar com o Wireshark, a análise de pacotes, e solução de problemas de rede em geral. O resto do livro está escrito em grande parte no mesmo formato, mas com foco nas diferentes áreas de análise de pacotes do mundo real.

8

LUTANDO CONTRA A LENTIDÃO DA REDE

Como um administrador de rede, muito do seu tempo será gasto corrigindo os computadores e serviços que estão sendo executados mais devagar do que o esperado. A queixa mais comum ouvida pelo pessoal de TI é que a rede está lenta. No entanto, só porque alguém diz que a rede está funcionando lentamente, não significa que um problema de rede seja a culpa.

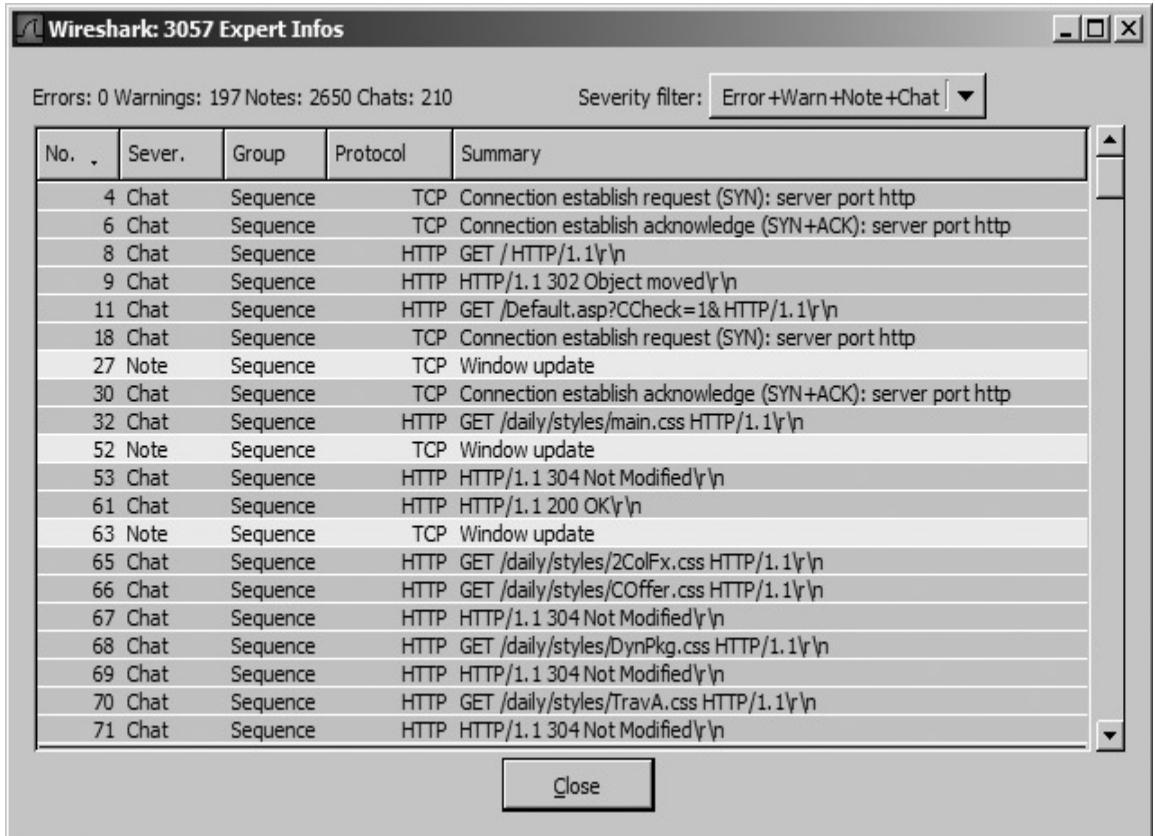
Portanto, antes de começar a resolver um problema de rede lenta, primeiro você tem que determinar como a rede realmente está, de fato, com lentidão. Neste capítulo, vamos olhar para várias situações diferentes em que um usuário está reclamando que a rede está lenta.

Anatomia do Download Lento

Vamos olhar para a anatomia de um download lento a nível de pacote. Percorrer todos os pacotes (como mostrado na figura abaixo), você vai ver um monte de padrão HTTP e tráfego TCP, e isso mostra o download, tendo lugar. Como aprendemos em nossa discussão sobre HTTP no capítulo 6, o HTTP é usado para solicitar os dados de um servidor web e, em seguida o TCP é usado para fazer o download desses dados do servidor remoto.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|----------------|----------------|----------|--|
| 12 | 5.516729 | 216.251.114.10 | 10.0.52.164 | TCP | [TCP segment of a reassembled PDU] |
| 13 | 5.517004 | 10.0.52.164 | 216.251.114.10 | TCP | 2468 > http [ACK] Seq=1605 Ack=1893 Win=258060 Len=0 |
| 14 | 5.517708 | 216.251.114.10 | 10.0.52.164 | TCP | [TCP segment of a reassembled PDU] |
| 15 | 5.517965 | 10.0.52.164 | 216.251.114.10 | TCP | 2468 > http [ACK] Seq=1605 Ack=3273 Win=258060 Len=0 |
| 16 | 5.518722 | 216.251.114.10 | 10.0.52.164 | TCP | [TCP segment of a reassembled PDU] |
| 17 | 5.518771 | 10.0.52.164 | 216.251.114.10 | TCP | 2468 > http [ACK] Seq=1605 Ack=4653 Win=258060 Len=0 |
| 18 | 5.709920 | 10.0.52.164 | 216.251.114.10 | TCP | 2470 > http [SYN] Seq=0 Len=0 MSS=1460 WS=2 |
| 19 | 5.723110 | 216.251.114.10 | 10.0.52.164 | TCP | [TCP segment of a reassembled PDU] |
| 20 | 5.723346 | 10.0.52.164 | 216.251.114.10 | TCP | 2468 > http [ACK] Seq=1605 Ack=6033 Win=258060 Len=0 |
| 21 | 5.724099 | 216.251.114.10 | 10.0.52.164 | TCP | [TCP segment of a reassembled PDU] |

A fim de filtrar o tráfego anormal que está tornando o nosso download lento, usaremos a janela **Expert Infos**. Para abrir essa janela, clique em **Analyze** na barra de menu e selecione **Expert Infos**. Você deve ver algo como a figura abaixo.



Por default, a janela **Expert Infos** mostra todos os avisos, erros, notas, e tráfego de conversação (bate-papo) do nosso arquivo de captura. O tráfego de conversação (bate-papo) em geral não é suspeito (pelo menos para essa finalidade), vamos modificar a configuração default selecionando **Error+Warn+Note** na caixa suspensa próxima as palavras **Severity filter**. Nossa nova janela **Expert Infos** será semelhante a da figura abaixo.

Wireshark: 2847 Expert Infos

Errors: 0 Warnings: 197 Notes: 2650 Chats: 210 Severity filter: Error+Warn+Note

| No. | Sever. | Group | Protocol | Summary |
|-----|--------|----------|----------|---------------|
| 27 | Note | Sequence | TCP | Window update |
| 52 | Note | Sequence | TCP | Window update |
| 63 | Note | Sequence | TCP | Window update |
| 158 | Note | Sequence | TCP | Window update |
| 162 | Note | Sequence | TCP | Window update |
| 168 | Note | Sequence | TCP | Window update |
| 186 | Note | Sequence | TCP | Window update |
| 229 | Note | Sequence | TCP | Window update |
| 230 | Note | Sequence | TCP | Window update |
| 254 | Note | Sequence | TCP | Window update |
| 258 | Note | Sequence | TCP | Window update |
| 266 | Note | Sequence | TCP | Window update |
| 270 | Note | Sequence | TCP | Window update |
| 274 | Note | Sequence | TCP | Window update |
| 334 | Note | Sequence | TCP | Window update |
| 338 | Note | Sequence | TCP | Window update |
| 346 | Note | Sequence | TCP | Window update |
| 350 | Note | Sequence | TCP | Window update |
| 354 | Note | Sequence | TCP | Window update |
| 358 | Note | Sequence | TCP | Window update |

Close

Observe na figura acima que a abundância dos pacotes em nosso arquivo de captura são os pacotes **TCP Window update**. A taxa de transmissão de dados através de uma rede é determinada pelo tamanho da janela de recepção TCP. Quando os clientes estão transferindo dados, eles vão enviar constantemente a atualização da janela TCP para os outros com a sua capacidade de receber dados aumentando ou diminuindo essa janela. Estes pacotes são usados para notificar um cliente que ele precisa aumentar ou diminuir o tamanho dos dados a serem transmitidos. Você pode pensar nisso como alguém pressionando o botão em uma fonte de água para você. Se o botão for pressionado demais, você não será capaz de capturar toda a água na boca, então você deve instruir a pessoa a diminuir a pressão sobre o botão. Por outro lado, se a pessoa não está pressionando o botão o suficiente, você não beberá água tanto quanto você poderia beber.

Em seguida, vemos o nosso primeiro pacote problemático. À medida que o download começa, nós começamos a ver o segmento TCP anterior sendo perdido, como mostrado na figura abaixo.

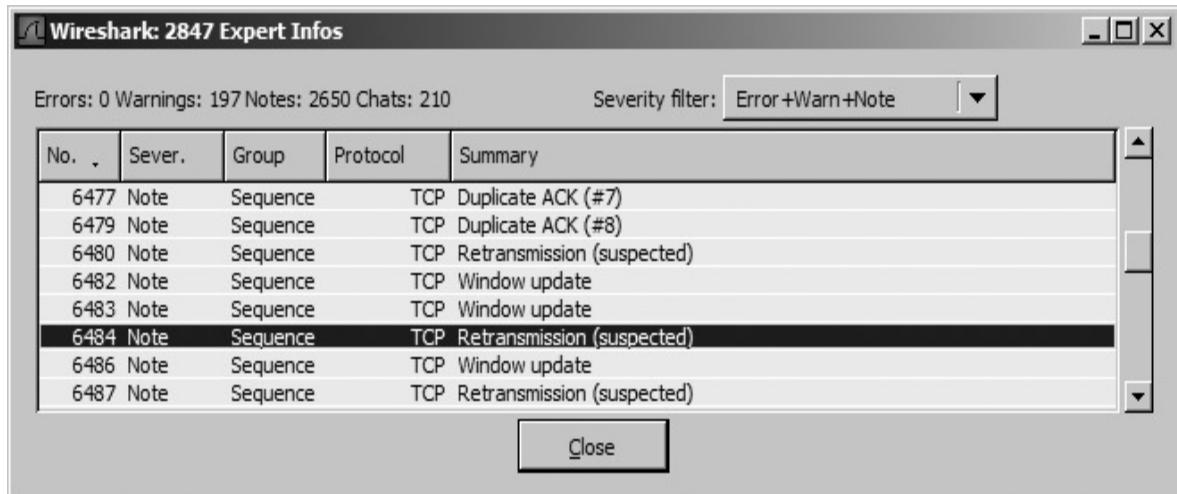
Wireshark: 2847 Expert Infos

Errors: 0 Warnings: 197 Notes: 2650 Chats: 210 Severity filter: Error+Warn+Note

| No. | Sever. | Group | Protocol | Summary |
|-----|--------|----------|----------|---|
| 577 | Warn | Sequence | TCP | Previous segment lost (common at capture start) |
| 578 | Note | Sequence | TCP | Duplicate ACK (#1) |
| 579 | Warn | Sequence | TCP | Out-Of-Order segment |
| 581 | Note | Sequence | TCP | Window update |
| 583 | Warn | Sequence | TCP | Previous segment lost (common at capture start) |
| 584 | Note | Sequence | TCP | Duplicate ACK (#1) |
| 585 | Warn | Sequence | TCP | Out-Of-Order segment |
| 587 | Note | Sequence | TCP | Window update |
| 593 | Warn | Sequence | TCP | Previous segment lost (common at capture start) |
| 594 | Note | Sequence | TCP | Duplicate ACK (#1) |
| 595 | Warn | Sequence | TCP | Out-Of-Order segment |
| 597 | Warn | Sequence | TCP | Out-Of-Order segment |
| 599 | Note | Sequence | TCP | Window update |
| 644 | Note | Sequence | TCP | Window update |
| 650 | Note | Sequence | TCP | Window update |
| 651 | Warn | Sequence | TCP | Previous segment lost (common at capture start) |
| 652 | Note | Sequence | TCP | Duplicate ACK (#1) |
| 653 | Warn | Sequence | TCP | Out-Of-Order segment |
| 655 | Note | Sequence | TCP | Window update |
| 672 | Note | Sequence | TCP | Window update |

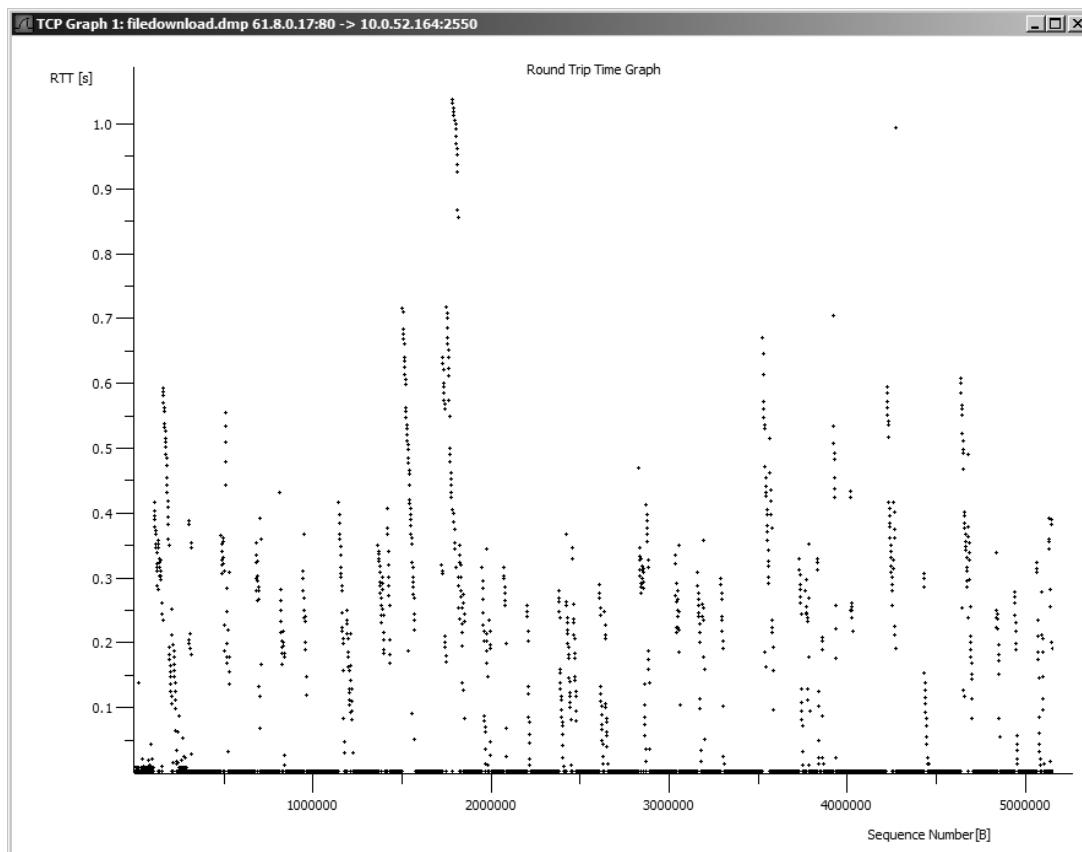
Close

Estes pacotes dizem-nos que durante o curso da transferência de dados, um pacote subitamente foi descartado. Em resposta, o cliente envia os pacotes Duplicate ACK para o servidor, solicitando que o pacote perdido seja enviado novamente. O cliente continua a enviar Duplicate ACKs até que recebe o pacote solicitado. Vemos então a retransmissão do pacote descartado como um TCP Retransmission na janela **Expert Infos**, como mostrado na figura abaixo.



No início do nosso download, vemos apenas um ou dois ACKs Duplicate em uma linha, mas como o download progride, começamos a ver mais e mais. Isto diz-nos que estamos a experimentar mais latência. Se você continuar a percorrer o resto da captura, você vai ver que ele está cheio de segmentos perdidos e Duplicate ACKs - o indicador de download lento no processo.

Convenientemente, o Wireshark permite que o gráfico do fluxo de TCP para o download, como mostrado na figura abaixo. Você pode acessar o gráfico, clicando em um pacote relacionado com o fluxo que deseja analisar (eu selecionei o pacote de número 1023) e escolhendo **Statistics > TCP Stream Graph > Round Trip Time Graph**. O TCP Stream Graph recurso do Wireshark é uma ótima maneira de visualizar o throughput de dados quando se trata de um fluxo TCP.



Embora este gráfico possa não ser esteticamente agradável, é uma ótima maneira de comparar o tempo de ida e volta (RTT) ao longo de uma captura de pacotes. Repare, por exemplo, que perto do começo do gráfico desta captura, vemos o RTT com mais de um segundo. Isto é completamente inaceitável para o download um arquivo. Mesmo quando fazemos um download de um arquivo da Internet, vemos tempo não superior a 0,1 segundos, com tempo ideal de não mais de 0,04 segundo (40 milissegundos). Este gráfico mostra-nos logo que temos um grande problema.

Uma Rota Lenta

O primeiro passo para resolver qualquer problema de rede lenta é determinar a origem do problema. No cenário seguinte, o help desk acabou de receber um telefonema de Owen, que está reclamando que sua conexão Internet está extremamente lenta.

O Que Sabemos

Não há muito que nós precisamos saber antes de podermos começar a endereçar esta queixa bastante simples. Verificamos que a questão da Internet lenta persiste independentemente do site visitado. E, após uma investigação aprofundada, nós sabemos que cada máquina que se encontra na mesma rede do Owen estão enfrentando o mesmo problema.

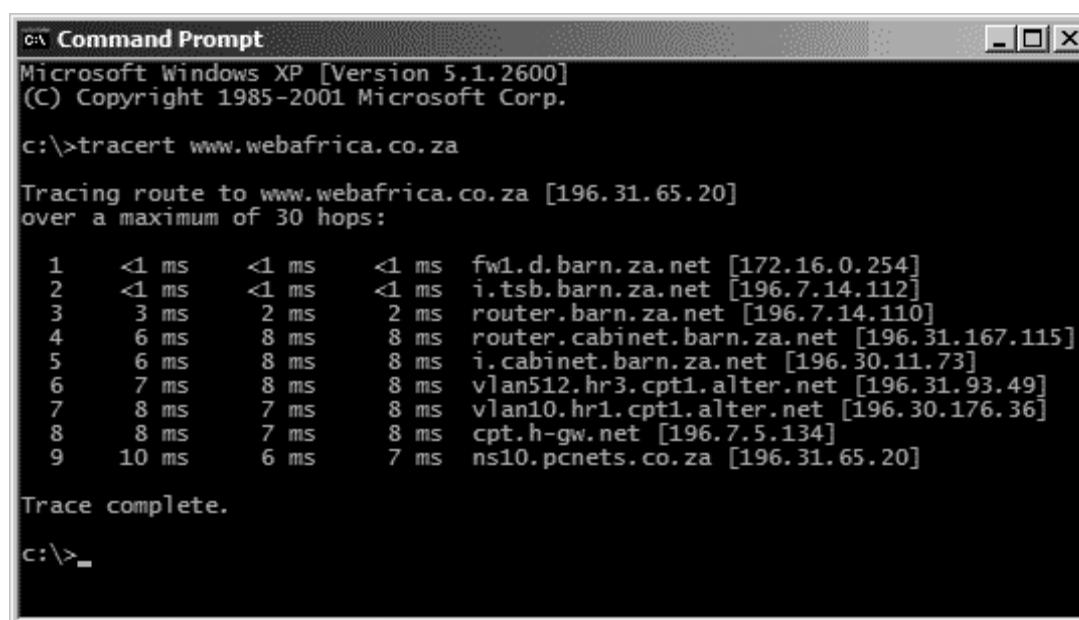
Farejando Através dos Fios

Uma vez que Owen foi o primeiro a reclamar sobre este problema, vamos realizar a análise de seu computador (embora provavelmente qualquer computador na rede seja suficiente). Vamos instalar o Wireshark diretamente em sua máquina para obter a captura de pacotes que precisamos.

Desde que o problema esteja afetando vários computadores, sabemos que não é um problema com o computador de Owen especificamente, uma captura apenas de seu computador que tenta acessar a Internet não vai nos dar as informações que necessitamos. Em vez disso, vamos usar o utilitário ICMP traceroute para obter uma melhor idéia de onde reside o problema.

O traceroute é uma ferramenta de diagnóstico baseado no ICMP (UDP baseados em Unix) que envia pacotes para cada roteador ao longo de um caminho, progredindo até atingir um destino especificado. O traceroute irá relatar algumas informações básicas sobre quaisquer atrasos que enfrentamos (como mostrado na saída na figura abaixo), mas para obter uma compreensão real sobre onde está o gargalo, vamos capturar os resultados do traceroute com o Wireshark.

Eu incluí uma imagem de uma tela de saída de traceroute como amostra na figura abaixo. Cada linha representa o tempo que leva para atravessar uma rede em uma rota até o destino.



```
c:\> Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\>tracert www.webafrica.co.za

Tracing route to www.webafrica.co.za [196.31.65.20]
over a maximum of 30 hops:

 1    <1 ms      <1 ms      <1 ms  fw1.d.barn.za.net [172.16.0.254]
 2    <1 ms      <1 ms      <1 ms  i.tsb.barn.za.net [196.7.14.112]
 3    3 ms       2 ms       2 ms   router.barn.za.net [196.7.14.110]
 4    6 ms       8 ms       8 ms   router.cabinet.barn.za.net [196.31.167.115]
 5    6 ms       8 ms       8 ms   i.cabinet.barn.za.net [196.30.11.73]
 6    7 ms       8 ms       8 ms   wlan512.hr3.cpt1.alter.net [196.31.93.49]
 7    8 ms       7 ms       8 ms   wlan10.hr1.cpt1.alter.net [196.30.176.36]
 8    8 ms       7 ms       8 ms   cpt.h-gw.net [196.7.5.134]
 9   10 ms      6 ms       7 ms   ns10.pcnets.co.za [196.31.65.20]

Trace complete.

c:\>
```

Análise

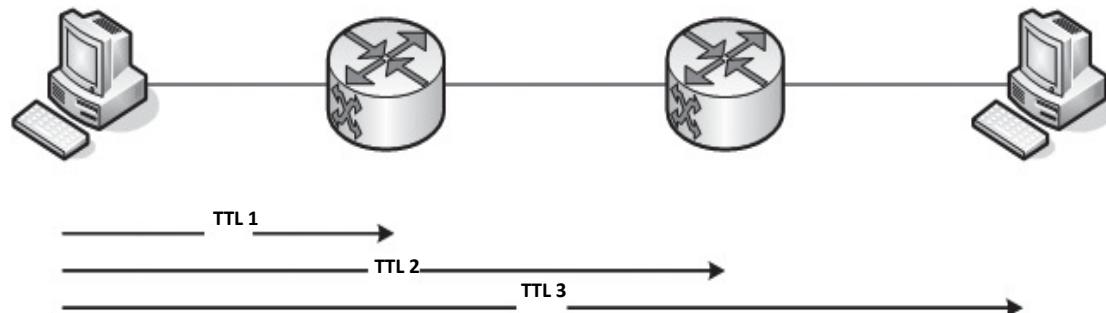
Olhando para o arquivo de captura (icmp-traceret slow.pcap, figura abaixo), a primeira coisa que vemos são os pacotes Echo (ping) a serem enviados a partir do computador de Owen para um host remoto.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|----------------|----------|---------------------|
| 1 | 0.000000 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |
| 2 | 3.364382 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |
| 3 | 6.368126 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |
| 4 | 9.371704 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |

Estes pacotes diferentes dos pacotes ping regular é uma importante maneira, como você verá se olhar na seção IP do painel **Packet Details**. A diferença é que o valor de TTL destes pacotes está definido como um, como mostrado na figura abaixo.

| |
|---|
| Internet Protocol, src: 24.6.126.218 (24.6.126.218), dst: 198.173.244.32 (198.173.244.32) |
| Version: 4 |
| Header length: 20 bytes |
| Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00) |
| Total Length: 92 |
| Identification: 0xb5f6 (46582) |
| Flags: 0x00 |
| Fragment offset: 0 |
| Time to live: 1 |
| Protocol: ICMP (0x01) |
| Header checksum: 0xb1fc [correct] |
| Source: 24.6.126.218 (24.6.126.218) |
| Destination: 198.173.244.32 (198.173.244.32) |

O valor Time-To-Live (TTL) é um valor numérico que determina quantas vezes um pacote saltar de um roteador até outro através de uma rede até chegar ao seu destino final. Um valor de 1 significa que um traceroute irá enviar um pacote para o dispositivo de destino, mas que o pacote termina quando ele atinge o primeiro roteador ao longo do percurso, neste mesmo tempo, um pacote ICMP TTL expirado será enviado de volta. Depois de recebido este ICMP TTL expired packet, o traceroute irá enviar outro pacote com um valor TTL de 2, o que causará um ICMP TTL expired packet a ser enviado de volta uma vez que o pacote atinge o segundo roteador ao longo do percurso. Esse processo continua até que um pacote tenha o valor de TTL que é apenas o suficiente para chegar ao destino, como ilustrado na figura abaixo.



Aplicando o nosso conhecimento recente sobre TTL em nossa situação atual, podemos ver imediatamente um problema com o primeiro pacote enviado. Este pacote tem um TTL de valor de 1, por isso imediatamente atinge o roteador interno da nossa rede que deveria nos responder, mas isso não acontece.

Uma vez que o computador de Owen não recebeu uma resposta imediata de volta ao primeiro pacote TTL de valor 1, ele espera cerca de três segundos (como mostrado no campo Time do Wireshark na figura abaixo) e então envia outra solicitação.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|----------------|----------|---------------------|
| 2 | 3.364382 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |
| 3 | 6.368126 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |

Quando o computador de Owen não recebe nenhuma resposta para esta segunda tentativa, ele aguarda aproximadamente três segundos e envia um último pacote para o roteador, que também não obtém êxito, como mostrado na figura abaixo.

| No. * | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|----------------|----------|---------------------|
| 3 | 6.368126 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |
| 4 | 9.371704 | 24.6.126.218 | 198.173.244.32 | ICMP | Echo (ping) request |

Neste ponto, o traceroute obtém uma resposta do primeiro roteador, assim que seu próximo pacote (pacote 4) que tem um valor TTL de 2. Este pacote chega ao segundo roteador com êxito, e o computador de Owen recebe o esperado ICMP tipo 11, código 0, que tem a mensagem Time-To-Live exceeded, mostrado na figura abaixo.

| No. * | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|--------------|----------|--|
| 5 | 9.393904 | 12.244.25.161 | 24.6.126.218 | ICMP | Time-to-live exceeded (Time to live exceeded in transit) |

Esse processo continua até o resto da captura, o valor de TTL é continuamente incrementado até que o destino seja alcançado.

O que podemos determinar a partir desta análise traceroute? Primeiro de tudo, sabemos que nosso problema reside no roteador interno da nossa rede, porque nunca conseguimos receber uma resposta ICMP dele. Roteadores são dispositivos muito complicados, por isso não nos aprofundaremos na semântica do que exatamente está errado com o nosso roteador. O ponto chave é que nós obtemos êxito determinando onde reside o problema: no roteador interno da nossa rede.

Resumo

Mais uma vez, o Wireshark nos salvou inúmeras horas de resolução de problemas nos permitindo rapidamente localizar a fonte do nosso problema. Enquanto o Wireshark não vai nos dizer o que há de errado com o nosso roteador ou como corrigi-lo, sabemos agora o suficiente para virar a nossa atenção para a configuração do roteador para saber mais sobre o problema.

Nós também aprendemos algumas coisas novas sobre o ICMP, bem como a forma de trabalho com o utilitário traceroute. (o traceroute tem várias outras opções configuráveis e de uso, você pode descobrir mais sobre eles, fazendo uma busca rápida pela Internet.)

Visão Dupla

Neste cenário, você terá instalado e configurado um novo computador para o Jeff, o empregado mais novo da empresa. Normalmente, quando você instala um novo computador, você espera que ele seja mais rápido que o resto dos dispositivos em sua rede. No entanto, depois de pouco tempo, os relatórios de Jeff passa por períodos de pico de utilização, o seu computador está com lentidão grave a ponto de que determinados serviços de rede tornam-se indisponíveis.

O Que Sabemos

Primeiro de tudo, sabemos que o computador de Jeff é novo, assim deve funcionar com um desempenho ideal. Fora isso, não há outros relatos de lentidão rede, durante o pico ou não de utilização. Sabemos também que Jeff é um usuário que usa muito da largura de banda. A maioria das suas tarefas está relacionada com a utilização da rede, e muitas vezes ele executa múltiplas aplicações net-centric de uma vez. Estas aplicações, juntamente com a Internet e clientes de email, criam uma carga acima da média do tráfego, mas que a nossa rede deve ser capaz de lidar facilmente.

Farejando Através dos Fios

Porque esse problema está relacionado apenas ao computador de Jeff, vamos instalar o Wireshark diretamente nele. A melhor época para analisar este problema é quando ele está acontecendo, que é durante o tempo de pico de uso. Queremos que Jeff possa executar sua rotina diária, por isso vamos iniciar a captura do arquivo, deixe-o funcionar por alguns minutos, enquanto Jeff faz as suas tarefas, e termine-as para que possamos olhar os dados coletados.

Análise

O título deste cenário realmente se torna claro quando você abre o arquivo de rastreamento, double-vision.pcap. Imediatamente, você vai notar tudo duplicado, todos os pacotes no arquivo de captura são repetidos, como você pode ver no início da captura na figura abaixo. Isto definitivamente não é normal.

Nota:

Por razões de simplicidade, vamos apenas olhar para seis pacotes, uma vez que é realmente tudo o que é necessário para nossos propósitos. Basta lembrar que todos os pacotes são duplicados para todas as comunicações a partir do computador de Jeff.

| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|--------------|--------------|----------|---|
| 1 | 0.000000 | 12.234.13.89 | 12.234.14.63 | TCP | 47063 > http [ACK] Seq=0 Ack=0 Win=4096 Len=0 |
| 2 | 0.025583 | 12.234.13.89 | 12.234.14.63 | TCP | [TCP Dup ACK 1#1] 47063 > http [ACK] Seq=0 Ack=0 Win=4096 Len=0 |
| 3 | 0.025776 | 12.234.14.63 | 12.234.13.89 | TCP | http > 47063 [RST] Seq=0 Len=0 |
| 4 | 0.066826 | 12.234.14.63 | 12.234.13.89 | TCP | http > 47063 [RST] Seq=0 Len=0 |
| 5 | 15.001667 | 12.234.13.89 | 12.234.14.63 | TCP | 1093 > 424 [SYN] Seq=0 Len=0 MSS=1460 |
| 6 | 15.086461 | 12.234.13.89 | 12.234.14.63 | TCP | 1093 > 424 [SYN] Seq=0 Len=0 MSS=1460 |

Existem duas causas comuns para pacotes duplicados em um arquivo de captura: inconsistências no encaminhamento e configuração incorreta do espelhamento de porta. Antes de chegarmos até essa nossa conclusão e tentar determinar a causa disso tudo, vamos ter certeza que os pacotes que estamos olhando são verdadeiras duplicatas um do outro.

Uma maneira de determinar se os dois pacotes são idênticos é olhar para o número de identificação IP de cada um em seu cabeçalho IP. Você vai encontrar esse ID na seção IP de um pacote no painel Packet Details. Você verá que o primeiro e segundo pacotes têm o mesmo número de identificação, 0xc509, como mostrado na figura abaixo.

| |
|---|
| Internet Protocol, Src: 12.234.13.89 (12.234.13.89), Dst: 12.234.14.63 (12.234.14.63) |
| Version: 4 |
| Header length: 20 bytes |
| Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00) |
| Total Length: 40 |
| Identification: 0xc509 (50441) |
| Flags: 0x00 |
| Fragment offset: 0 |
| Time to live: 47 |
| Protocol: TCP (0x06) |
| Header checksum: 0x915b [correct] |
| Source: 12.234.13.89 (12.234.13.89) |
| Destination: 12.234.14.63 (12.234.14.63) |

O mesmo é verdade para os terceiro e quarto pacotes, ambos têm um ID da transação 0xaca7, como mostrado na figura abaixo. Continuando a lista, nós achamos que o mesmo é verdadeiro para cada par de pacotes no arquivo de captura.

| |
|---|
| Internet Protocol, Src: 12.234.14.63 (12.234.14.63), Dst: 12.234.13.89 (12.234.13.89) |
| Version: 4 |
| Header length: 20 bytes |
| Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00) |
| Total Length: 40 |
| Identification: 0xaca7 (44199) |
| Flags: 0x00 |
| Fragment offset: 0 |
| Time to live: 128 |
| Protocol: TCP (0x06) |
| Header checksum: 0x58bd [correct] |
| Source: 12.234.14.63 (12.234.14.63) |
| Destination: 12.234.13.89 (12.234.13.89) |

Agora que sabemos que todos os pacotes são duplicatas exatas tanto quanto a carga em uso, podemos começar a tentar determinar qual das duas soluções possíveis é mais provável que seja correta, roteamento inconsistente ou erro de espelhamento de porta. Para esse fim, vamos olhar para os valores TTL dos pacotes.

Se esses valores diferem, sinaliza um problema de roteamento interno; se são iguais, então temos provavelmente um problema de espelhamento de porta.

Conforme mostrado na figura abaixo, vemos que o valor TTL do pacote 1 é 47, e o valor do pacote 2 é 46. Isso nos diz que definitivamente temos um problema de roteamento interno. O fato que o segundo pacote foi diminuído de 1 mostra que o mesmo passou por um roteador em algum lugar e depois foi devolvido de volta à nossa máquina.

Porque este problema é está apenas ocorrendo com o computador de Jeff, concluímos que deve ser isolado nele, mais do que em um roteador na rede. Após uma investigação mais aprofundada, achamos que o seu novo computador foi configurado com uma máscara de subrede errada.

| |
|----------------------|
| Packet 1: |
| Fragment offset: 0 |
| Time to live: 47 |
| Protocol: TCP (0x06) |
| Packet 2: |
| Fragment offset: 0 |
| Time to live: 46 |
| Protocol: TCP (0x06) |

Resumo

Se uma máquina está configurada com uma máscara de subrede errada, o resultado pode ser uma multiplicidade de problemas, incluindo a prevenção da comunicação dele com os outros computadores. Neste caso, todos os pacotes enviados a partir do computador de Jeff são devolvidos, essencialmente duplicando a quantidade de tráfego que o computador tem que lidar diminuindo a comunicação diminuindo tremendamente durante os horários de pico.

Será que o Servidor Emperrou?

Surpresa! Outro usuário está se queixando de uma conexão de Internet lenta. Desta vez, Eric reclama que ele não pode acessar uma parte do site da Novell para o download de alguns softwares necessários. Cada vez que ele visita o site, o seu browser carrega e carrega, mas nada acontece. Deve ser um problema com a rede, certo?

O Que Sabemos

Após uma verificação completa da rede, determinamos que o acesso à Internet é normal para todas as máquinas exceto para a máquina de Eric. Portanto, o problema deve ser específico da estação de trabalho de Eric. Seu computador está executando o Windows, e está completamente atualizado com todos os service packs mais recentes e patches. Após mais investigação, descobrimos que o único problema é com uma seção especial do site da Novell.

Farejando Através dos Fios

Porque o problema aqui é apenas com o computador do Eric, podemos instalar o Wireshark em seu sistema e capturar os pacotes que precisamos. O problema ocorre quando ele visita uma seção especial do site da Novell, por isso vamos obter este arquivo de rastreamento, enquanto este problema específico está ocorrendo.

Análise

Quando você abre o http-client refuse.pcap (mostrado na figura abaixo) deve ser capaz de identificar imediatamente a comunicação HTTP, já que há um pedido HTTP após o handshake TCP inicial. Na verdade, esse pedido HTTP parece normal até os pacotes 28 e 29, como você verá abaixo. Vamos pular até eles e ver se podemos localizar o problema.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|---|
| 1 | 0.000000 | 67.161.32.69 | 130.57.5.25 | TCP | 1782 > http [SYN] Seq=0 Len=0 MSS=1460 WS=2 |
| 2 | 0.027029 | 130.57.5.25 | 67.161.32.69 | TCP | http > 1782 [SYN, ACK] Seq=0 Ack=1 Win=6144 Len=0 MSS |
| 3 | 0.027068 | 67.161.32.69 | 130.57.5.25 | TCP | 1782 > http [ACK] Seq=1 Ack=1 Win=258060 Len=0 |
| 4 | 0.028241 | 67.161.32.69 | 130.57.5.25 | HTTP | GET /img/flash/load_stream.html?temp=1&id=webex_conve |
| 5 | 0.061432 | 130.57.5.25 | 67.161.32.69 | TCP | http > 1782 [ACK] Seq=1 Ack=926 Win=5219 Len=0 |
| 6 | 0.072229 | 130.57.5.25 | 67.161.32.69 | TCP | [TCP segment of a reassembled PDU] |
| 7 | 0.073391 | 130.57.5.25 | 67.161.32.69 | TCP | [TCP segment of a reassembled PDU] |
| 8 | 0.073430 | 67.161.32.69 | 130.57.5.25 | TCP | 1782 > http [ACK] Seq=926 Ack=2761 Win=258060 Len=0 |
| 9 | 0.074556 | 130.57.5.25 | 67.161.32.69 | TCP | [TCP segment of a reassembled PDU] |
| 10 | 0.074752 | 130.57.5.25 | 67.161.32.69 | TCP | [TCP segment of a reassembled PDU] |
| 11 | 0.074770 | 67.161.32.69 | 130.57.5.25 | TCP | 1782 > http [ACK] Seq=926 Ack=4301 Win=258060 Len=0 |

Fique de olho na coluna Tempo nesta captura. Todos os pacotes são recebidos sem atraso até o pacote 28. Estamos no meio de uma transação HTTP quando de repente há uma defasagem de 9 segundos entre os pacotes de 27 e 28.

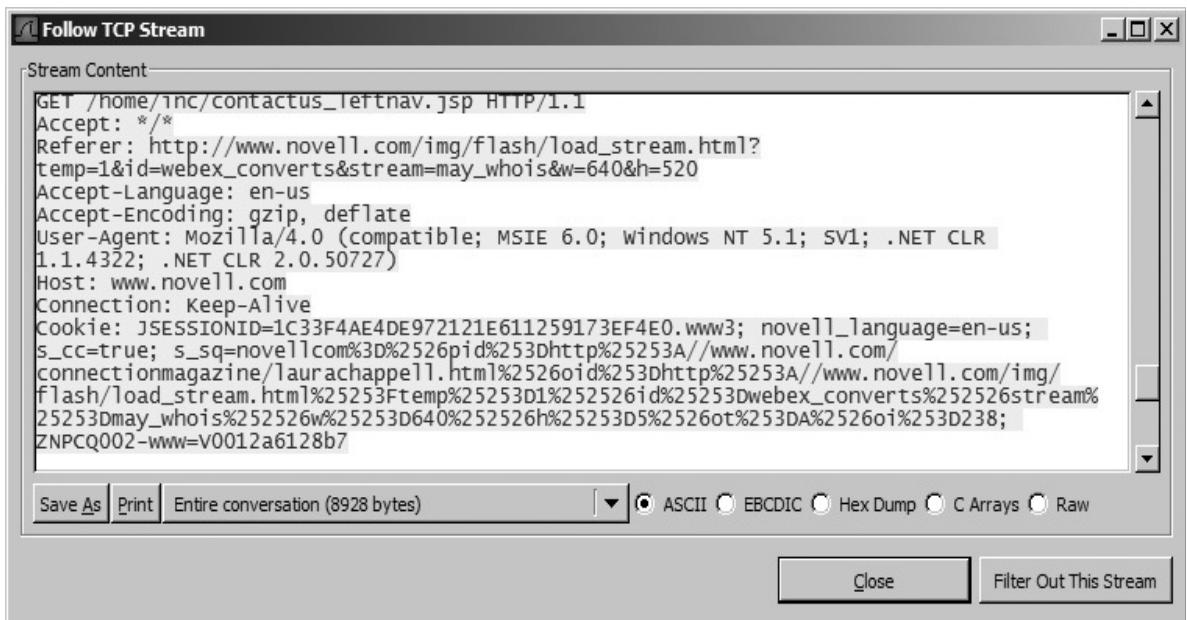
No mundo das comunicações de rede, um atraso de 9 segundos entre pacotes é completamente inaceitável, a menos que você esteja esperando por alguma forma de entrada de dados do usuário. Depois de passar 9 segundos, o servidor não pode mais transmitir os dados de volta que ele precisa ao cliente, por isso envia um pacote RST para terminar a conexão. Nossa cliente não desistiu ainda, e ele espera um adicional de 55 segundos (como mostrado na figura abaixo) antes de reconhecer o reset.

| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|---------------|--------------|----------|--|
| 28 | 0.000000 | 216.52.17.206 | 67.161.32.69 | TCP | http > 1783 [RST] Seq=0 Len=0 |
| 29 | 55.834680 | 67.161.32.69 | 130.57.5.25 | TCP | 1782 > http [RST, ACK] Seq=0 Ack=0 Win=0 Len=0 |

O servidor deixou de se comunicar com o cliente, e temos de encontrar o porquê. Poderíamos passar o passo a passo em toda a captura e analisar cada pacote, mas seria um processo extremamente longo e tedioso. Em vez disso, nós vamos tomar o caminho mais fácil.

Uma vez que estamos lidando com uma transação HTTP, o fluxo TCP deve ser de fácil leitura, enquanto nós podemos seguir o arquivo de rastreamento. Depois de abrir o fluxo TCP, observe que as cores diferentes são usadas para mostrar a comunicação: O vermelho é utilizado para transferência de dados de nossos clientes, e o azul é usado para mostrar os dados transferidos do servidor remoto.

Olhando para esse tráfego, você não consegue ver nada além do que o normal do HTML sendo transferido? Se você navegar até a segunda parte do tráfego que vem do nosso cliente, você verá um pedido para obter um applet Flash a partir do servidor Novell, como mostrado na figura abaixo. Isto é onde reside o problema. A página web Owen está tentando visualizar uma solicitação de um objeto Flash, este tipo de pedido pode ser facilmente bloqueado por um bloqueador de pop-up. Isso é apenas o que está acontecendo aqui.



Resumo

Após um pouco de investigação sobre os dados Flash sendo chamado a partir do site da Novell, você aprende que o site tenta abrir o seu conteúdo principal em uma nova janela flash, que o bloqueador de pop-up do Internet Explorer do Eric está bloqueando. Enquanto o navegador não for capaz de nos dar alguma informação útil sobre o problema (exceto uma mensagem de tempo limite de conexão), usamos o Wireshark, alguns conceitos básicos de análise de pacotes, e um pouco de paciência para localizar o ponto exato onde o processo de comunicação estava sendo prejudicado.

Uma falha de Análise

Neste cenário seguinte, um dos usuários da nossa rede tem chamado no help desk reclamando que a rede está funcionando muito lentamente. Ele não pode acessar a Internet ou qualquer rede-centric aplicações a uma velocidade razoável, e ele é realmente ficando para trás em seu trabalho. O que está tornando as coisas tão lentas?

O Que Sabemos

Depois de examinar outros usuários da rede, aprendemos que o problema da Internet é generalizado. Todos os usuários relatam que a Internet é tão lenta que é quase inutilizável. O roteador de borda de sua rede também indica alta utilização do processador, mostrando que ele está lidando com um tráfego muito substancial, tanto de saída como de entrada.

Nota:

O roteador de borda descreve a localização de um roteador em uma rede. Um roteador de borda está na extremidade de uma rede e a conecta ao mundo exterior.

Farejando Através dos Fios

Porque o roteador de borda manipula todo o tráfego entre a rede local e a Internet, e mostra uma carga de processamento alta, o roteador de borda é o melhor ponto de análise aqui.

Análise

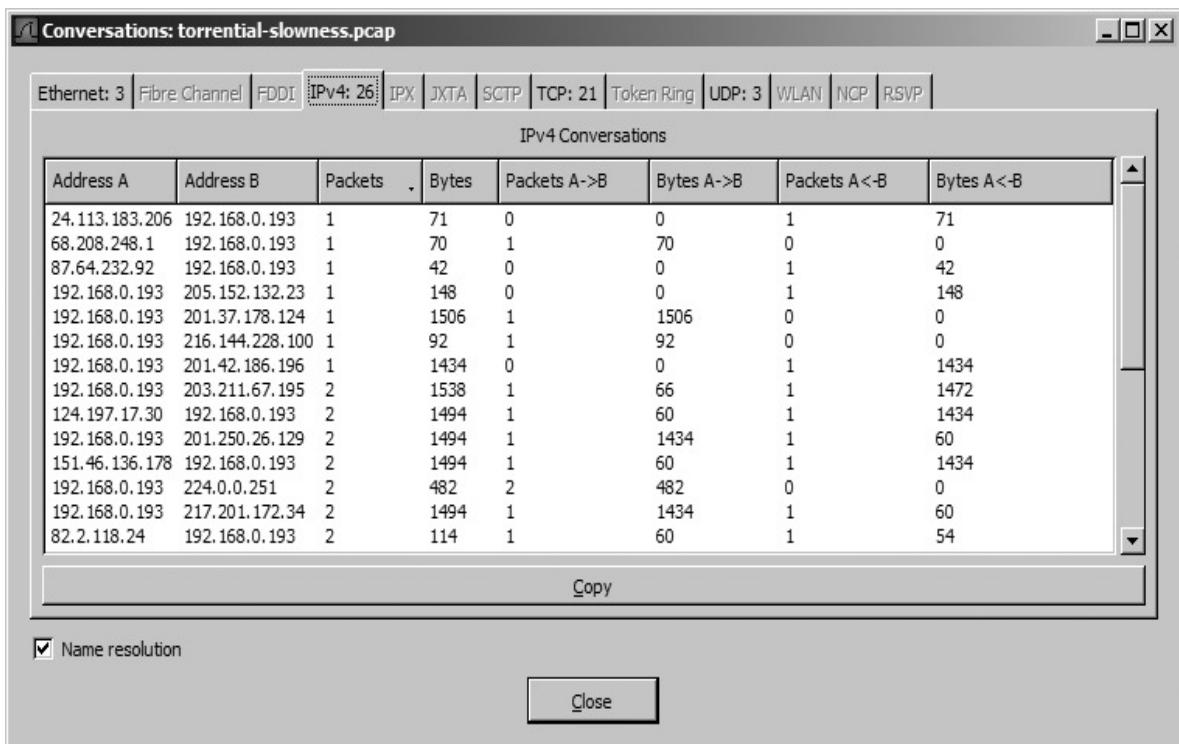
Vamos usar espelhamento de porta para enfrentar este cenário, porque obviamente não podemos instalar o Wireshark em um roteador.

Os pacotes incluídos na captura `slowness.pcap-torrencais` oferece apenas uma breve amostragem das muitas conexões acontecendo em nossa rede, como mostrado na figura abaixo.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|----------------|----------------|----------|--|
| 1 | 0.000000 | 203.211.67.195 | 192.168.0.193 | TCP | 11766 > 1534 [PSH, ACK] Seq=0 Ack=0 Win=65535 Len=1418 |
| 2 | 0.000033 | 192.168.0.193 | 203.211.67.195 | TCP | 1534 > 11766 [ACK] Seq=7090 Ack=4294965878 Win=65535 Len=0 |
| 3 | 0.000053 | 124.197.17.30 | 192.168.0.193 | TCP | 26507 > 4512 [ACK] Seq=0 Ack=0 Win=64227 Len=0 |
| 4 | 0.000070 | 192.168.0.193 | 124.197.17.30 | TCP | 4512 > 26507 [PSH, ACK] Seq=8117 Ack=0 Win=65108 Len=1380 |
| 5 | 0.000121 | 213.17.90.77 | 192.168.0.193 | TCP | 14173 > 3331 [ACK] Seq=0 Ack=0 Win=65535 Len=0 |
| 6 | 0.000136 | 192.168.0.193 | 213.17.90.77 | TCP | 3331 > 14173 [PSH, ACK] Seq=216 Ack=0 Win=65492 Len=1452 |
| 7 | 0.000147 | 192.168.0.193 | 213.17.90.77 | TCP | 3331 > 14173 [PSH, ACK] Seq=1668 Ack=0 Win=65492 Len=1452 |
| 8 | 0.009806 | 189.142.91.6 | 192.168.0.193 | TCP | 3326 > 6881 [ACK] Seq=0 Ack=0 Win=64240 Len=0 |
| 9 | 0.009844 | 192.168.0.193 | 189.142.91.6 | TCP | 6881 > 3326 [PSH, ACK] Seq=8280 Ack=0 Win=65006 Len=1380 |
| 10 | 0.025255 | 142.68.42.31 | 192.168.0.193 | TCP | 6881 > 4853 [ACK] Seq=0 Ack=0 Win=17280 Len=0 |

Um sistema dentro da nossa rede (192.168.0.193) aparece repetidamente nesta captura, fazendo e recebendo ligações de um monte de sistemas fora da nossa rede. Mais preocupante ainda, a maior parte do tráfego que está sendo enviado com a bandeira do TCP PSH, o que obriga um computador receptor a ignorar o seu buffer e repassar todo o tráfego diretamente, à frente de qualquer outro tráfego. Isso é quase sempre um mau sinal.

Ainda pior, a maioria dessas conexões já passou a fase do handshake TCP, o que significa que existe transferência de dados de/para o nosso cliente. Você pode ter uma noção de quantas dessas conexões estão ocorrendo, observando os diálogos das conversações mostrados na figura abaixo.



Em apenas um segundo de captura, existem 27 diferentes conversações TCP acontecendo!

A maneira mais simples para aliviar este problema seria ir para ao computador ofensor e dar uma vasculhada, mas o que pode ser divertido nisso? Nós vamos fazer a forma de analisar os pacotes.

Olhando para os pacotes, o primeiro curso de ação pode ser rastrear os endereços IP remotos e ver onde eles estão localizados, geralmente através da realização de uma pesquisa WHOIS em cada endereço IP. No entanto, neste caso você encontrará rapidamente que a maioria desses endereços IP não aponta para qualquer empresa ou mesmo para a mesma área, mas sim em locais diferentes em torno do mundo.

Para melhor avaliar os pacotes, você poderia ver se o fluxo TCP detém todas as informações valiosas. Neste caso, seguindo as provas inúteis do fluxo TCP, os dados apresentados não nos levam a nada, como você pode ver na figura abaixo.

O Servidor POP de Email

Em termos de importância, aos olhos dos empregados o e-mail está no mesmo nível da Internet . Sendo esse o caso, quando não está trabalhando, você estará ouvindo sobre isso.

Neste cenário, todos os usuários da rede estão reclamando que seu e-mail está tomando um tempo extremamente longo para chegar ao seu destino. Tanto para o caso do e-mail enviado a outros domínios, mesmo o caso do e-mail enviado para os colegas de trabalho dentro da mesma organização. Vamos chegar a fundo nisto.

O Que Sabemos

Todo o e-mail da nossa empresa é gerenciado através de um servidor de email. Depois de fazer algumas pesquisas, nós confirmamos que este problema existe para todos os clientes de e-mail em nossa rede. Considerando que um e-mail típico do escritório, normalmente é entregue instantaneamente, a entrega está agora a tomar de 10 a 15 minutos. A demora é a mesma para o recebimento de e-mail externo.

Farejando Através dos Fios

Porque o nosso problema está relacionado a um serviço que está hospedado em um servidor de email, vamos colocar o nosso analisador de lá. O problema tem sido consistente durante todo o dia de trabalho, portanto, qualquer hora é boa para capturar os pacotes.

Análise

Quando você olhar para os resultados da captura (e-mail troubles.pcap) você verá exatamente o que você deve ver em um servidor de e-mail: pacotes de e-mail. Há um monte de Post Office Protocol (POP) pacotes

que entram em nosso servidor de e-mail (veja a figura abaixo), mas apenas quantos e em que velocidade? Talvez o nosso servidor de e-mail esteja sendo sobrecarregado.

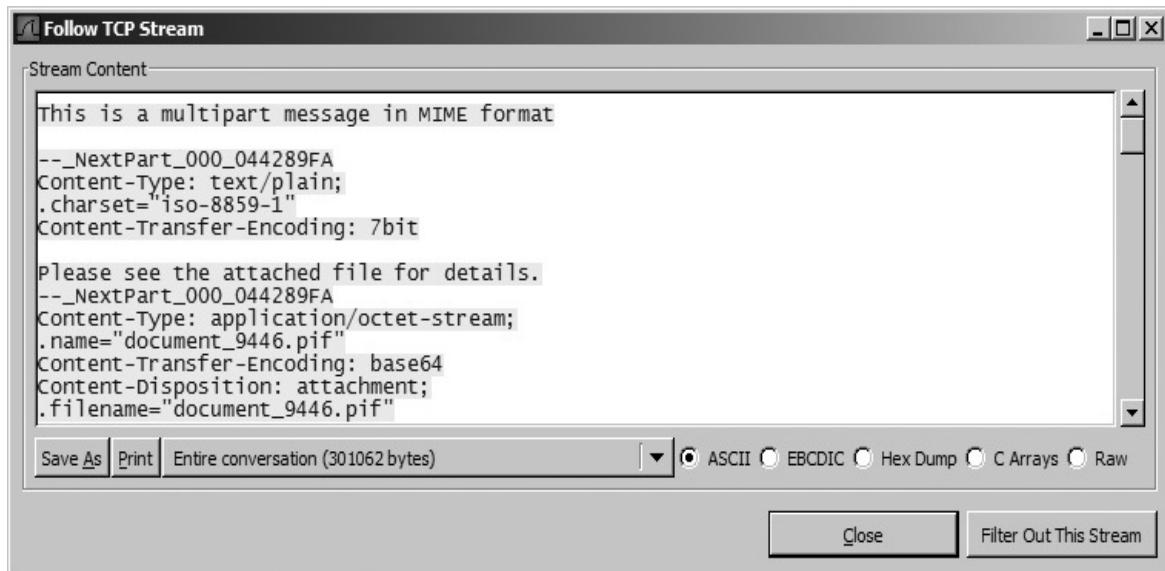
| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|---------------|---------------|----------|--|
| 1 | 0.000000 | 12.234.13.202 | 161.58.73.170 | POP | Request: RETR 20 |
| 2 | 0.079320 | 161.58.73.170 | 12.234.13.202 | TCP | pop3 > 1567 [ACK] Seq=0 Ack=9 Win=49152 Len=0 |
| 3 | 0.090650 | 161.58.73.170 | 12.234.13.202 | POP | Response: +OK 100220 octets |
| 4 | 0.091089 | 161.58.73.170 | 12.234.13.202 | POP | Continuation |
| 5 | 0.091117 | 12.234.13.202 | 161.58.73.170 | TCP | 1567 > pop3 [ACK] Seq=9 Ack=2920 Win=64512 Len=0 |
| 6 | 0.092467 | 161.58.73.170 | 12.234.13.202 | POP | Continuation |

Para determinar a taxa em que nós estamos recebendo os pacotes POP, altere o formato de exibição **Seconds Since Begenning or Capture** e olhe para o último pacote no arquivo. Este resultado diz-nos que estamos vendo dois minutos de tráfego intenso, como mostrado na coluna Time na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|------------|---------------|---------------|----------|---|
| 360 | 121.664143 | 12.234.13.202 | 161.58.73.170 | TCP | 1567 > pop3 [ACK] Seq=27 Ack=301035 Win=63337 Len=0 |

Agora podemos começar a olhar para cada fluxo de comunicação para ver se alguma coisa anormal está acontecendo.

A grande coisa sobre um pacote POP é que se você quiser ver o conteúdo da mensagem de email, tudo que você tem a fazer é ler o TCP stream associado a ele. Por exemplo, se você fizer isso para o pacote 1, você verá que este e-mail inclui o texto, bem como um anexo, document_9446.pif, como mostrado na figura abaixo.



Olhando ainda mais por este fluxo, vemos outra mensagem de outro endereço de e-mail com aparência suspeita, mas também tem um arquivo PIF anexado a ela.

Uma busca rápida por arquivos PIF irá dizer-lhe que se trata de Arquivos de Informação de Programa, não algo que você vê normalmente através de e-mail. Não só isso, esses arquivos tendem a ser executáveis muito grandes. Durante o decurso deste arquivo de captura, estes arquivos chegam a partir de múltiplas fontes.

O que temos aqui é um influxo de spam (e, possivelmente, vírus ou wormcontaining) e-mail que está sobrecarregando o nosso servidor e tornando o tráfego de email muito lento através da rede.

Resumo

Nosso servidor de e-mail está sendo esmagado por um alto volume de spam com grandes anexos. Esta condição é muito comum de ver quando monitoramos o desempenho de um servidor de email. Como uma organização cresce, a quantidade de spam recebido cresce com ele. No caso da nossa rede, os usuários podem ser pacientes e cair fora da lentidão, ou a organização pode investir em algum tipo de solução de filtragem de spam.

Acessando o Gnutella

Este cenário é ao longo das mesmas linhas o nosso cenário BitTorrent. Tina, uma usuária em nossa rede, chama e reclama que seu computador está incrivelmente executando devagar, se ela está fazendo algo localmente, sobre a rede local, ou através da Internet.

O Que Sabemos

Este cenário apresenta o mesmo caso como nosso exemplo anterior BitTorrent. Como tal, sabemos que este problema é generalizado e atinge outros usuários também. No entanto, os outros usuários são apenas relatam sobre velocidades lentas quando tratam com a Internet e aplicações net-centric. O roteador de borda da nossa rede de comunicação está com processamento alto e uma grande quantidade de tráfego de entrada e de saída.

Farejando Através dos Fios

Neste caso, todos os sintomas de computadores infectados são consistentes com o nosso exemplo do BitTorrent, com a exceção do computador de Tina. Não são apenas seus pedidos net-centric lento, mas seu computador que está em geral se arrastando.

Porque seu computador está apresentando sintomas únicos, vamos supor que o problema está relacionado ao seu computador, de modo que é por onde vamos começar a nossa análise. No entanto, como o computador de Tina está tão lento, instalar o Wireshark diretamente nele não poderia ser melhor a idéia, já que a lentidão pode causar a perda de pacotes durante o processo de captura. Nós vamos usar o espelhamento de porta em seu lugar.

Análise

Este arquivo de captura (gnutella.pcap) é longo, mas se parece muito com a captura do BitTorrent, em sua maior parte. Como você pode ver na figura abaixo, O computador de Tina, 10.1.4.176, parece estar tentando se comunicar com vários hospedeiros diferentes fora da nossa rede. As maioria dessas tentativas voltam sem resposta após o SYN inicial ou são negadas pelo cliente com um pacote RST.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|--|
| 1 | 0.000000 | 10.1.4.176 | 66.68.99.53 | TCP | 3663 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 0.028257 | 10.1.4.176 | 198.82.59.65 | TCP | 3684 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 3 | 0.060831 | 198.82.59.65 | 10.1.4.176 | TCP | 6346 > 3684 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 |
| 4 | 0.499894 | 10.1.4.176 | 198.82.59.65 | TCP | 3684 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 5 | 0.531212 | 198.82.59.65 | 10.1.4.176 | TCP | 6346 > 3684 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 |
| 6 | 0.999962 | 10.1.4.176 | 198.82.59.65 | TCP | 3684 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 7 | 1.030592 | 198.82.59.65 | 10.1.4.176 | TCP | 6346 > 3684 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 |

Vários fatores podem estar causando essas falhas de conexões, mas antes de investigar mais profundamente, vamos ver exatamente a quantidade de tráfego que são confrontadas para que possamos determinar a extensão do nosso problema. Uma boa maneira de fazer isto é olhar para os diálogos das transações para ver quantas conversas TCP e IP estão acontecendo, como mostrado na figura abaixo.

A janela **Conversations** mostra que esse arquivo de rastreamento contém 81 transações IP e 243 TCP, como você pode ver nas abas na parte superior da figura abaixo. Esse grande número de transações é normalmente aceitável se você está vendo o tráfego capturado a partir de um servidor, mas esta é uma estação de trabalho, não é normal ver estas transações sobre um curto período de tempo.

Conversations: gnutella.pcap

Ethernet: 2 | Fibre Channel | FDDI | **IPv4: 81** | IPX | JXTA | SCTP | TCP: 243 | Token Ring | UDP | WLAN | NCP | RSVP

IPv4 Conversations

| Address A | Address B | Packets | Bytes | Packets A->B | Bytes A->B | Packets A-<B | Bytes A-<B |
|------------|-----------------|---------|-------|--------------|------------|--------------|------------|
| 10.1.4.176 | 24.201.82.175 | 8 | 496 | 8 | 496 | 0 | 0 |
| 10.1.4.176 | 66.68.178.113 | 8 | 496 | 8 | 496 | 0 | 0 |
| 10.1.4.176 | 213.134.172.104 | 8 | 496 | 8 | 496 | 0 | 0 |
| 10.1.4.176 | 213.112.19.252 | 8 | 496 | 8 | 496 | 0 | 0 |
| 10.1.4.176 | 66.68.99.53 | 9 | 558 | 9 | 558 | 0 | 0 |
| 10.1.4.176 | 65.27.219.8 | 9 | 558 | 9 | 558 | 0 | 0 |
| 10.1.4.176 | 65.2.8.194 | 9 | 558 | 9 | 558 | 0 | 0 |
| 10.1.4.176 | 65.92.95.244 | 10 | 620 | 10 | 620 | 0 | 0 |
| 10.1.4.176 | 217.0.84.236 | 10 | 620 | 10 | 620 | 0 | 0 |
| 10.1.4.176 | 217.80.211.133 | 10 | 620 | 10 | 620 | 0 | 0 |
| 10.1.4.176 | 168.191.249.194 | 10 | 620 | 10 | 620 | 0 | 0 |
| 10.1.4.176 | 206.183.20.219 | 11 | 682 | 11 | 682 | 0 | 0 |
| 10.1.4.176 | 213.66.196.164 | 11 | 682 | 11 | 682 | 0 | 0 |
| 10.1.4.176 | 217.84.38.130 | 11 | 682 | 11 | 682 | 0 | 0 |

Name resolution

Se você olhar para algumas destas transações TCP, você vai ver que cada uma envolve um host remoto. Você pode dizer que a maioria dessas transações não foi bem sucedida, uma vez que o número de pacotes para cada uma é muito baixo.

A fim de obter as informações que realmente precisamos para avaliar a comunicação acontecendo aqui, precisamos ver uma transação bem-sucedida. A melhor forma de fazê-lo é a partir da janela **Conversations** que já temos em aberto. Com a guia **IPv4:81** selecionada nessa janela, clique em Packets para classificar todos as transações pelo número de pacotes que contêm, como mostrado na figura abaixo.

Conversations: gnutella.pcap

Ethernet: 2 | Fibre Channel | FDDI | **IPv4: 81** | IPX | JXTA | SCTP | TCP: 243 | Token Ring | UDP | WLAN | NCP | RSVP

IPv4 Conversations

| Address A | Address B | Packets | Bytes | Packets A->B | Bytes A->B | Packets A-<B | Bytes A-<B |
|------------|-----------------|---------|-------|--------------|------------|--------------|------------|
| 10.1.4.176 | 205.251.201.194 | 48 | 2928 | 24 | 1488 | 24 | 1440 |
| 10.1.4.176 | 213.66.32.81 | 27 | 2196 | 15 | 1278 | 12 | 918 |
| 10.1.4.176 | 198.82.59.65 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 24.45.10.102 | 24 | 1524 | 12 | 744 | 12 | 780 |
| 10.1.4.176 | 24.16.138.119 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 217.80.241.56 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 65.92.46.3 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 24.178.36.229 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 64.156.183.243 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 24.101.212.95 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 24.184.236.185 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 213.100.81.169 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 65.196.165.41 | 24 | 1464 | 12 | 744 | 12 | 720 |
| 10.1.4.176 | 65.68.0.63 | 24 | 1464 | 12 | 744 | 12 | 720 |

Name resolution

Você deve ver a comunicação entre o computador de Tina e o host remoto, 65.34.1.56, no topo da lista, como mostrado na figura abaixo.

Conversations: gnutella.pcap

Ethernet: 2 | Fibre Channel | FDDI | IPv4: 81 | IPX | JXTA | SCTP | TCP: 243 | Token Ring | UDP | WLAN | NGP | RSVP

TCP Conversations

| Address A | Port A | Address B | Port B | Packets | Bytes | Packets A->B | Bytes A->B | Packets A<-B | Bytes A<-B |
|----------------|--------|------------|--------|---------|-------|--------------|------------|--------------|------------|
| 65.34.1.56 | 6346 | 10.1.4.176 | 3766 | 11 | 1153 | 5 | 652 | 6 | 501 |
| 213.66.32.81 | 6346 | 10.1.4.176 | 3736 | 9 | 732 | 4 | 306 | 5 | 426 |
| 213.66.32.81 | 6346 | 10.1.4.176 | 3815 | 9 | 732 | 4 | 306 | 5 | 426 |
| 65.34.1.56 | 6346 | 10.1.4.176 | 3852 | 9 | 1039 | 4 | 592 | 5 | 447 |
| 213.66.32.81 | 6346 | 10.1.4.176 | 3880 | 9 | 732 | 4 | 306 | 5 | 426 |
| 198.82.59.65 | 6346 | 10.1.4.176 | 3684 | 8 | 488 | 4 | 240 | 4 | 248 |
| 24.45.10.102 | 6346 | 10.1.4.176 | 3691 | 8 | 508 | 4 | 260 | 4 | 248 |
| 24.16.138.119 | 6346 | 10.1.4.176 | 3710 | 8 | 488 | 4 | 240 | 4 | 248 |
| 217.80.241.56 | 6346 | 10.1.4.176 | 3711 | 8 | 488 | 4 | 240 | 4 | 248 |
| 62.211.209.170 | 6346 | 10.1.4.176 | 3713 | 8 | 488 | 4 | 240 | 4 | 248 |
| 65.92.46.3 | 6346 | 10.1.4.176 | 3715 | 8 | 488 | 4 | 240 | 4 | 248 |
| 24.178.36.229 | 6346 | 10.1.4.176 | 3716 | 8 | 488 | 4 | 240 | 4 | 248 |
| 217.81.165.208 | 6346 | 10.1.4.176 | 3722 | 8 | 488 | 4 | 240 | 4 | 248 |

Name resolution

Agora, veja só esses pacotes clicando nesta conversa, selecionando **Apply as Selected**, selecionando **Apply a Filter**, e depois escolhendo **A<->B**. O resultado é que você verá somente os pacotes mostrados na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|-----------|------------|-------------|----------|---|
| 426 | 52.436286 | 10.1.4.176 | 65.34.1.56 | TCP | 3766 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 429 | 52.514080 | 65.34.1.56 | 10.1.4.176 | TCP | 6346 > 3766 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460 |
| 430 | 52.514799 | 10.1.4.176 | 65.34.1.56 | TCP | 3766 > 6346 [ACK] Seq=1 Ack=1 Win=8760 Len=0 |
| 431 | 52.515422 | 10.1.4.176 | 65.34.1.56 | Gnutel | |
| 432 | 52.722234 | 65.34.1.56 | 10.1.4.176 | TCP | 6346 > 3766 [ACK] Seq=1 Ack=31 Win=17490 Len=0 |
| 433 | 52.724068 | 10.1.4.176 | 65.34.1.56 | Gnutel | |
| 434 | 52.844222 | 65.34.1.56 | 10.1.4.176 | Gnutel | |

Os pacotes mostrados na figura acima oferecem algumas informações adicionais que nos leva diretamente ao problema. Especificamente, os pacotes 431, 433 e 434 são identificados como pacotes Gnutel. Estes pacotes são característicos do tráfego Gnutella enviados ou recebidos através da rede Gnutella de compartilhamento de arquivos. Clicando neles obtemos um pouco mais de detalhes, como mostrado na figura abaixo.

| |
|--|
| Transmission Control Protocol, Src Port: 3766 (3766), Dst Port: 6346 (6346), Seq: 1, Ack: 1, Len: 30 |
| Gnutella Protocol |
| Gnutella upload / download stream |

O painel **Packet Details** do pacote 431 (na figura acima) realmente não nos dá todas as informações úteis, a não ser de que este pacote é um download/upload que atravessa a rede Gnutella. Se olharmos para o painel **Packet Bytes** (figura abaixo), porém, vemos algo um pouco alarmante.

| | |
|--|--------------------|
| 0000 00 50 54 ff 3c 77 00 10 7b 24 37 30 08 00 45 00 | .PT.<w.. {\$70..E. |
| 0010 00 46 3f dd 40 00 7f 06 6a ca 0a 01 04 b0 41 22 | .F?@... j.....A" |
| 0020 01 38 0e b6 18 ca 03 7c 21 cd 9b 41 89 a0 50 18 | :8.....!..A..P. |
| 0030 22 38 06 6c 00 00 47 45 54 20 2f 67 65 74 2f 32 | "8..GE T /get/2 |
| 0040 34 2f 53 6f 72 6f 72 69 74 79 20 53 65 78 20 4b | 4/Sorori ty Sex K |
| 0050 69 74 74 65 | itte |

Este fluxo de dados em particular mostra um comando GET de um download de um arquivo com um nome que contém as palavras sorority sex kitten. Encontramos o nosso tráfego suspeito.

Como um breve aparte, aqui está outra maneira de dizer que este é o tráfego Gnutella. Se você olhar para todas as tentativas de transações tendo lugar, você vai notar que as informações de cabeçalho do painel **Packet List** mostram toda esta comunicação acontecendo na porta 6346, como mostrado na figura abaixo.

| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|------------|---------------|----------|--|
| 583 | 75.544302 | 10.1.4.176 | 65.27.229.23 | TCP | 3803 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 584 | 75.544499 | 10.1.4.176 | 24.28.233.147 | TCP | 3802 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 585 | 75.544750 | 10.1.4.176 | 65.2.243.188 | TCP | 3801 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 586 | 75.545190 | 10.1.4.176 | 24.178.197.5 | TCP | 3804 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |
| 587 | 75.545446 | 10.1.4.176 | 24.249.29.119 | TCP | 3799 > 6346 [SYN] Seq=0 Len=0 MSS=1460 |

Uma busca rápida por esse número de porta em <http://www.iana.org> irá listar os serviços relacionados a esta porta.

Resumo

A rede Gnutella é comumente usada para a descarga e distribuição de vários tipos de arquivo. Esta idéia pode parecer grande no início, mas, infelizmente resultou em uma grande rede peer-to-peer de pornografia, bem como de software pirata, filmes e música.

Neste cenário, parece que a Tina, ou alguém usando o computador da Tina, tem instalado algum tipo de cliente Gnutella para baixar material pornográfico.

Considerações Finais

Se você olhar como cada um destes cenários foi resolvido, você notará que a maioria dos problemas não estavam realmente relacionados à rede. Isso é muito comum quando se trata de denúncias sobre uma rede lenta. Normalmente, não é a rede que está lenta, mas problemas com computadores individuais ou aplicações que os usuários executam que a tornam desse jeito.

9

ANÁLISE BASEADA EM SEGURANÇA

Neste capítulo, vamos mergulhar em vários cenários relacionados com a segurança de rede e trabalhar através deles com o Wireshark. Com o aparecimento de ameaças de hackers, ladrões de identidade, e roubo de dados das empresas, você não pode deixar de ser capaz de analisar a segurança de sua rede no nível do pacote.

Sistema Operacional Fingerprinting

O Sistema Operacional (OS) fingerprinting é uma técnica usada por hackers para identificar o sistema operacional de um computador remoto, a fim de obter informações que possa invadi-lo. O OS fingerprinting trabalha usando uma máquina remota para enviar vários comandos a um computador de destino. Quando a máquina remota recebe as respostas a esses comandos, ele pode interpretar essas respostas para tentar adivinhar o sistema operacional que o computador de destino está usando. Conhecer o sistema operacional de um computador permitirá encontrar rapidamente as falhas de funcionamento deste sistema.

Quando você abrir o osfingerprinting.pcap, você verá vários tipos diferentes de tráfego ICMP, como mostrado na figura abaixo. Alguns desses tráfegos, como pedido Echo (ping), e resposta Echo (ping), são comuns e não devem ser motivo de alarme. No entanto, o tráfego como Timestamp request/reply, Address mask request, e Information request são incomuns.

| No. - | Time | Source | Destination | Protocol | Info |
|-------|-----------|-----------|-------------|----------|----------------------|
| 11 | 1.863030 | 10.0.0.29 | 10.0.0.2 | ICMP | Timestamp request |
| 12 | 1.863238 | 10.0.0.2 | 10.0.0.29 | ICMP | Timestamp reply |
| 13 | 1.869470 | 10.0.0.29 | 10.0.0.2 | ICMP | Timestamp request |
| 14 | 1.869609 | 10.0.0.2 | 10.0.0.29 | ICMP | Timestamp reply |
| 15 | 2.739445 | 10.0.0.29 | 10.0.0.2 | ICMP | Address mask request |
| 16 | 2.742531 | 10.0.0.29 | 10.0.0.2 | ICMP | Address mask request |
| 17 | 7.062589 | 10.0.0.29 | 10.0.0.2 | ICMP | Information request |
| 18 | 7.064628 | 10.0.0.29 | 10.0.0.2 | ICMP | Information request |
| 19 | 11.354823 | 10.0.0.29 | 10.0.0.2 | ICMP | Echo (ping) request |
| 20 | 11.355045 | 10.0.0.2 | 10.0.0.29 | ICMP | Echo (ping) reply |
| 21 | 11.359669 | 10.0.0.29 | 10.0.0.2 | ICMP | Echo (ping) request |
| 22 | 11.359816 | 10.0.0.2 | 10.0.0.29 | ICMP | Echo (ping) reply |

O tráfego ICMP não usual que vemos na figura acima, sugere que nosso sistema é o alvo de um ataque usando o ICMP através de varreduras OS fingerprinting. A máquina que faz o ataque envia esses pedidos e usa a resposta do sistema alvo (se houver) para determinar o sistema operacional em execução na máquina alvo.

Nota:

Como nunca veremos o tráfego ICMP tipos 13, 15 ou 17 em circunstâncias normais, nós podemos criar um filtro mostrando somente esse tipo de tráfego rapidamente. Esse filtro é `icmp.type==13 || icmp.type==15 || icmp.type==17`.

Um Simples Escaneamento (varredura) de Porta

Hackers podem usar varreduras de portas para obter informações importantes sobre uma rede. Usando um software especializado em varredura de porta, um hacker pode tentar se conectar a um dispositivo através de uma porta específica, como a 21 (FTP) e 80 (HTTP). Com as informações recebidas a partir desses exames, o hacker pode encontrar portas abertas que podem permitir o acesso à sua rede. Pense em uma porta aberta como um túnel secreto em um castelo bem guardado. Uma vez que o hacker conhece esse túnel, ele pode muito bem ser capaz de entrar sem usar nenhum outro truque. A figura abaixo, com base no arquivo de captura portscan.pcap, mostra como um software de varredura de portas trabalha.

| No. - | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|----------------------------------|
| 7 | 0.607512 | 10.100.25.14 | 10.100.18.12 | TCP | 16748 > telnet [SYN] Seq=0 Len=0 |
| 8 | 0.707986 | 10.100.25.14 | 10.100.18.12 | TCP | 12502 > ftp [SYN] Seq=0 Len=0 |
| 9 | 0.808340 | 10.100.25.14 | 10.100.18.12 | TCP | 30382 > 6000 [SYN] Seq=0 Len=0 |
| 10 | 0.904949 | 10.100.25.14 | 10.100.18.12 | TCP | 27986 > 1025 [SYN] Seq=0 Len=0 |
| 11 | 1.004235 | 10.100.25.14 | 10.100.18.12 | TCP | 25488 > smtp [SYN] Seq=0 Len=0 |
| 12 | 1.110883 | 10.100.25.14 | 10.100.18.12 | TCP | 6729 > sunrpc [SYN] Seq=0 Len=0 |
| 13 | 1.212836 | 10.100.25.14 | 10.100.18.12 | TCP | 29169 > 1028 [SYN] Seq=0 Len=0 |
| 14 | 1.307771 | 10.100.25.14 | 10.100.18.12 | TCP | 24305 > 9100 [SYN] Seq=0 Len=0 |
| 15 | 1.407052 | 10.100.25.14 | 10.100.18.12 | TCP | 17851 > 1029 [SYN] Seq=0 Len=0 |
| 16 | 1.512738 | 10.100.25.14 | 10.100.18.12 | TCP | 10985 > finger [SYN] Seq=0 Len=0 |

Como você pode ver na figura acima, existem muitos poucos pacotes sendo enviados entre 10.100.25.14 (a máquina local) e 10.100.18.12 (um computador remoto). Quando você der uma olhada nesses pacotes, você vai ver exatamente porque eles são tão suspeitos.

Nosso arquivo de rastreamento mostra que todos os pacotes enviados a partir do computador remoto estão sendo enviados para um número de porta diferente na máquina local (por exemplo, 21 e 1028). Mas o mais importante, é que estas portas podem ser exploradas comumente, por um telnet, microsoft-ds, FTP e SMTP. Quando você vê um computador remoto enviando vários pacotes para estas portas exploradas, você pode tipicamente assumir que uma varredura de porta está ocorrendo.

Uma Impressora Abarrotada de Impressão

Mesmo as menores organizações podem ter várias impressoras em sua rede. O custo do papel, da tinta, da manutenção, da propriedade, pode levar ao interesse de uma impressora de baixo volume conectada a rede. Neste cenário uma da impressora na nossa rede começou a imprimir lixo por completo, e ninguém sabe a fonte do mesmo. Nosso objetivo é encontrar a fonte de tais documentos misteriosos e pôr fim a impressão deles.

O Que Sabemos

Nossa impressora é uma impressora de alto volume conectada a rede através de um servidor. Não tem nenhuma permissão especial atribuída a ela, nem qualquer capacidade extra de registro. O problema é constante. Mesmo quando limpamos sua fila de impressão, ela enche imediatamente e começa a imprimir novamente.

Farejando Através dos Fios

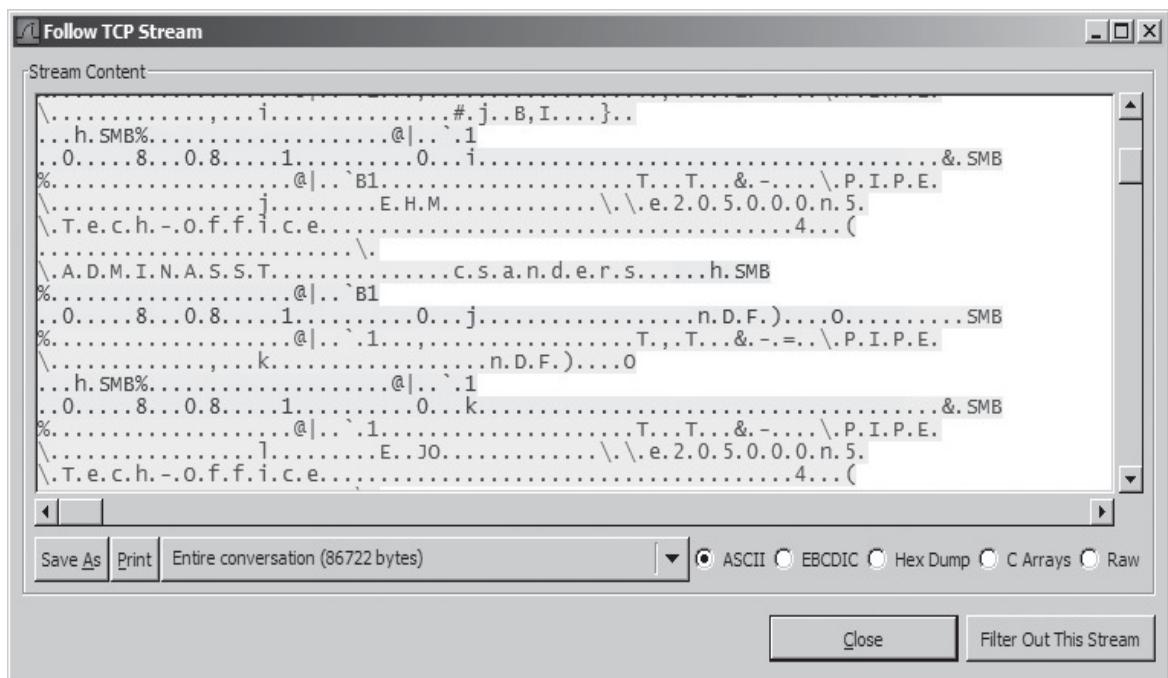
Porque a impressora problemática está instalada em um servidor, haverá muito tráfego fluindo na rede, e nós vamos ter um monte de dados para classificar através dela. Independente disso, o instalar o Wireshark diretamente no servidor é o melhor caminho a ser seguido. Como o problema parece ser uma constante, podemos capturar pacotes a qualquer momento.

Análise

O arquivo de captura printerproblem.pcap é um exemplo muito bom do tráfego para uma impressora. Como você pode ver na figura abaixo, o nosso servidor, 10.100.16.15, está recebendo um fluxo maciço de pacotes de SPOOLS de um cliente dentro de nossa rede, 10.100.17.47.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|---|
| 49 | 0.058610 | 10.100.17.47 | 10.100.16.15 | SPOOLS | DeletePrinterIC request |
| 50 | 0.058679 | 10.100.16.15 | 10.100.17.47 | SPOOLS | DeletePrinterIC response |
| 51 | 0.059582 | 10.100.17.47 | 10.100.16.15 | SPOOLS | OpenPrinterEx request, \\e205000n5\Tech-office |
| 52 | 0.059799 | 10.100.16.15 | 10.100.17.47 | SPOOLS | OpenPrinterEx response |
| 53 | 0.060312 | 10.100.17.47 | 10.100.16.15 | SPOOLS | ClosePrinter request |
| 54 | 0.060396 | 10.100.16.15 | 10.100.17.47 | SPOOLS | ClosePrinter response |
| 55 | 0.061042 | 10.100.17.47 | 10.100.16.15 | SPOOLS | StartDocPrinter request, OpenPrinterEx(\\e205000n5\Tech-office) |
| 56 | 0.062040 | 10.100.16.15 | 10.100.17.47 | SPOOLS | StartDocPrinter response |

É bastante fácil de identificar a fonte de impressão neste caso, mas nós ainda não resolvemos o problema. Para saber mais sobre o que está acontecendo, vamos visualizar o fluxo de dados TCP a ser enviado para a impressora. Você verá que os dados estão sendo impressos a partir do Microsoft Word e que o username da pessoa que está imprimindo é csanders (figura abaixo).



Resumo

Embora não tenhamos parado o fluxo de pacotes SPOOLS neste cenário, usamos o Wireshark para encontrar rapidamente a fonte do problema misterioso de impressão. Tendo identificado a origem, podemos descobrir o porquê esta informação está sendo enviada para a impressora. (Muito provavelmente, o cliente 10.100.17.47 em nossa rede está comprometido de alguma forma.)

Um FTP Interrompido

O FTP é um dos meios mais utilizados para transferência de grandes quantidades de dados. A empresa que estaremos olhando tem um servidor de FTP interno que ela usa para manter todo o seu software de pré-lançamento. Ultimamente, o técnico de TI encarregado da manutenção e acompanhamento deste servidor tem notado uma grande quantidade de tráfego no servidor depois de algumas horas. Infelizmente, o software servidor de FTP não tem a funcionalidade de registro, por isso a única maneira de obter uma boa compreensão do que está acontecendo é obter uma captura de pacotes. Queremos identificar a razão do aumento do uso da largura de banda pelo servidor e eliminar a fonte.

O Que Sabemos

O servidor FTP está executando um software muito antigo com nenhuma funcionalidade de registro. Todos os grandes desenvolvedores da empresa ter usernames e contas que lhes permitem pleno acesso a todos os arquivos no servidor. Este servidor é também configurado para que ele possa ser acessado de fora da rede, de modo que os desenvolvedores possam trabalhar de casa.

Farejando Através dos Fios

Uma vez que este servidor está em nossa rede, instalar o Wireshark sobre ele possa parecer o melhor método a utilizar. No entanto, já que o servidor está passando por um nível muito elevado de carga de tráfego, os pacotes poderão ser descartados se acabar entupindo muito o servidor, por isso vamos usar o espelhamento de porta em seu lugar.

Análise

Quando você abre a captura crack.pcap ftp, você vai ver um monte de coisas acontecendo em um período muito curto de tempo. De nossa discussão sobre FTP no capítulo 6, você deverá estar familiarizado com a forma como o processo de autenticação FTP acontece.

Após o handshake TCP inicial, um processo de login normalmente ocorre de modo que o usuário pode começar a interagir com o servidor. Nesta captura, saltamos direito para o processo de autenticação de username e password, e como você pode ver no pacote 4 (figura abaixo), esta tentativa de autenticação falha.

```
File Transfer Protocol (FTP)
  530 Login incorrect.\r\n
    Response code: Not logged in (530)
    Response arg: Login incorrect.
```

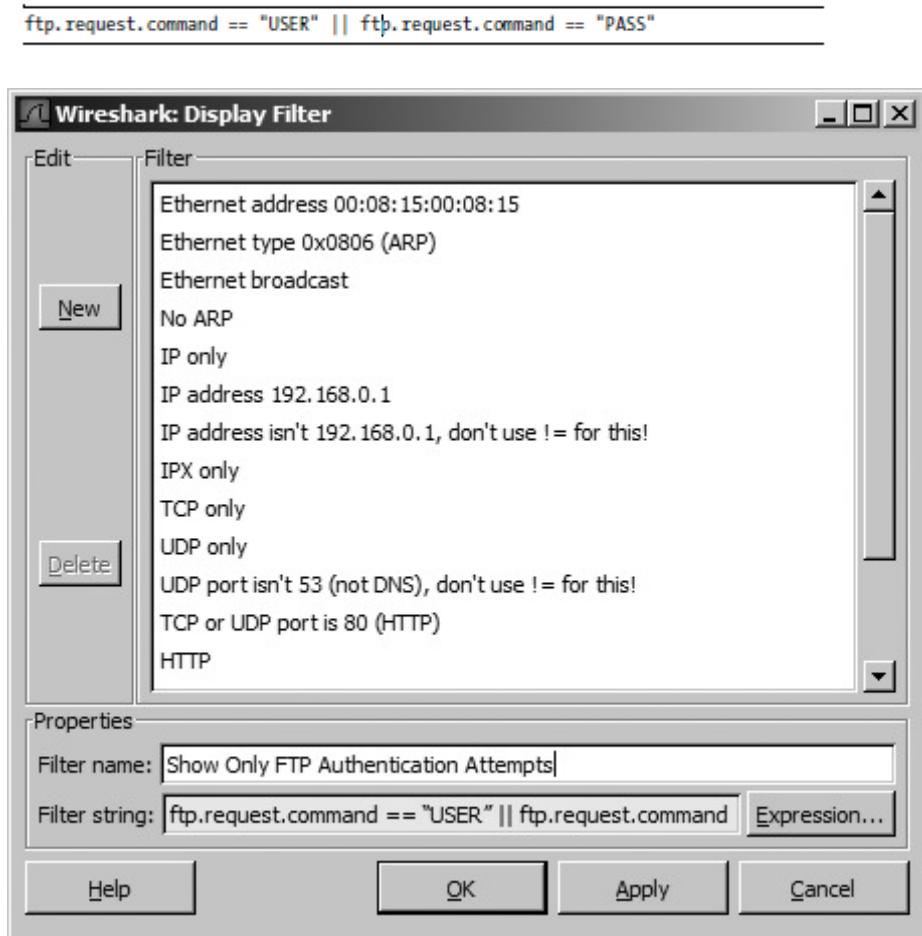
Podemos supor que o usuário que está tentando fazer login tenha digitado incorretamente sua senha, mas essa suposição é rapidamente deixada de lado nos próximos pacotes. Como mostrado na figura abaixo, vemos muitas falhas de autenticação.

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|----------------|----------------|----------|--|
| 27 | 0.104111 | 10.121.70.151 | 10.234.125.254 | TCP | ftp > 2220 [FIN, ACK] Seq=22 Ack=1 Win=49152 [TCP CHECKSUM INCORRECT] Len=0 |
| 28 | 0.104155 | 10.234.125.254 | 10.121.70.151 | TCP | 2220 > ftp [ACK] Seq=1 Ack=23 Win=17447 [TCP CHECKSUM INCORRECT] Len=0 |
| 29 | 0.108560 | 10.121.70.151 | 10.234.125.254 | FTP | Response: 530 Login incorrect. |
| 30 | 0.108773 | 10.121.70.151 | 10.234.125.254 | TCP | ftp > 2221 [ACK] Seq=34 Ack=14 Win=49152 [TCP CHECKSUM INCORRECT] Len=0 |
| 31 | 0.112332 | 10.234.125.254 | 10.121.70.151 | TCP | 2222 > ftp [FIN, ACK] Seq=13 Ack=56 Win=17447 [TCP CHECKSUM INCORRECT] Len=0 |
| 32 | 0.120024 | 10.121.70.151 | 10.234.125.254 | FTP | Response: 530 Login incorrect. |
| 33 | 0.121851 | 10.234.125.254 | 10.121.70.151 | TCP | 2221 > ftp [FIN, ACK] Seq=14 Ack=56 Win=17447 [TCP CHECKSUM INCORRECT] Len=0 |
| 34 | 0.122830 | 10.121.70.151 | 10.234.125.254 | TCP | ftp > 2223 [ACK] Seq=34 Ack=11 Win=49152 [TCP CHECKSUM INCORRECT] Len=0 |
| 35 | 0.141432 | 10.121.70.151 | 10.234.125.254 | TCP | ftp > 2222 [ACK] Seq=56 Ack=14 Win=49152 [TCP CHECKSUM INCORRECT] Len=0 |
| 36 | 0.141886 | 10.121.70.151 | 10.234.125.254 | TCP | ftp > 2222 [FIN, ACK] Seq=56 Ack=14 Win=49152 [TCP CHECKSUM INCORRECT] Len=0 |
| 37 | 0.141939 | 10.234.125.254 | 10.121.70.151 | TCP | 2222 > ftp [ACK] Seq=14 Ack=57 Win=17447 [TCP CHECKSUM INCORRECT] Len=0 |
| 38 | 0.145312 | 10.121.70.151 | 10.234.125.254 | TCP | ftp > 2221 [ACK] Seq=56 Ack=15 Win=49152 [TCP CHECKSUM INCORRECT] Len=0 |
| 39 | 0.145896 | 10.121.70.151 | 10.234.125.254 | FTP | Response: 530 Login incorrect. |

Imediatamente após a tentativa de autenticação não vemos outra tentativa de login para o servidor (10.121.70.151) de um cliente dentro da nossa própria rede (10.234.125.254). O curioso sobre esta solicitação é que o usuário está tentando se logar usando a conta de administrador, como visto no pacote 10 na figura abaixo.

```
File Transfer Protocol (FTP)
  331 Password required for admin.\r\n
    Response code: User name okay, need password (331)
    Response arg: Password required for admin.
```

Esta é uma grande oportunidade de usar um filtro de visualização para mostrar apenas os pacotes que representam uma tentativa de login FTP, assim como:



Agora, se olharmos na coluna **Info** de cada tentativa de login, podemos ver que as passwords estão sendo usadas em ordem alfabética, ou seja, o atacante está percorrendo cada letra do alfabeto em sucessão. Este é um sinal indicador de que alguém está tentando adivinhar a senha de uma conta usando um ataque estilizado dicionário. O ataque estilizado dicionário é aquele em que as senhas são adivinhadas com base em um usuário ou em uma máquina criando dicionário de palavras. Se você olhar para o tempo entre cada tentativa, você também pode ver que essas tentativas de adivinhar a password estão acontecendo rápido demais para ser feita por um ser humano, elas provavelmente estão sendo feitas por uma ferramenta de um hacker. Encontramos com sucesso a fonte da utilização de nosso aumento de banda.

Resumo

Temos confirmado que um computador dentro da nossa rede está sendo atacado por um programa hacker projetado para executar um ataque de dicionário sobre o servidor FTP. Mas nosso trabalho não está feito ainda. Neste ponto, precisamos determinar se o empregado cuja máquina está a lançar o ataque é responsável por fazê-lo ou se o seu computador foi comprometido a partir de outro computador remotamente.

O Vírus (Worm) Blaster

A crescente ameaça de vírus e worms que se espalham pela Internet assusta os administradores de sistema e usuários finais. Neste cenário, Eddy faz uma chamada no help desk com a preocupação de que seu computador foi infectado com um vírus. Toda vez que ele inicia seu computador ele recebe uma mensagem que o mesmo será desligado em 60 segundos. Uma vez que este timer expira, o computador é desligado como indicado. Este processo continua se repetindo continuamente e ele não acessa seu computador por mais de 60 segundos de cada vez.

O Que Sabemos

Sabemos que Eddy tende a ser cuidadoso com a segurança, de modo que um spyware não é uma preocupação imediata. Nossa empresa utiliza um software antivírus, no entanto é descentralizado e na maior parte gerenciado pelo usuário.

Farejando Através dos Fios

Toda vez que você suspeitar que um vírus ou worm pode ser a causa de um computador com problema, normalmente não é uma idéia sensata instalar um sniffer diretamente nessa máquina. Programas mal-intencionados muitas vezes podem funcionar contra farejadores de pacotes não lhes permitindo executar corretamente. Nossa melhor estratégia aqui é usar o espelhamento de porta. A captura começará logo que o computador é ligado e terminará quando o computador se desligar após o temporizador de 60 segundos expirar.

Análise

O arquivo de captura blaster.pcap, mostrado na figura abaixo, registra alguns pacotes TCP sendo transmitidos de nosso computador para outro computador suspeito na rede local via portas 1793 e 4444. Estes pacotes são capturados em um momento quando nada está ativo na máquina que não seja no período de 60 segundos, dessa atividade suspeita.

| No. ▾ | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|--|
| 1 | 0.000000 | 10.234.0.239 | 10.234.2.116 | TCP | 1793 > 4444 [ACK] Seq=0 Ack=0 Win=17330 [TCP C] |
| 2 | 0.000191 | 10.234.2.116 | 10.234.0.239 | TCP | 4444 > 1793 [PSH, ACK] Seq=0 Ack=0 Win=64475 [TCP S] |
| 3 | 0.218319 | 10.234.0.239 | 10.234.2.116 | TCP | 1793 > 4444 [ACK] Seq=0 Ack=20 Win=17310 [TCP S] |
| 4 | 1.673435 | 10.234.0.239 | 10.234.2.116 | TCP | 1793 > 4444 [PSH, ACK] Seq=0 Ack=20 Win=17310 |
| 5 | 1.673773 | 10.234.2.116 | 10.234.0.239 | TCP | 4444 > 1793 [PSH, ACK] Seq=20 Ack=18 Win=64457 |
| 6 | 1.859752 | 10.234.0.239 | 10.234.2.116 | TCP | 1793 > 4444 [ACK] Seq=18 Ack=38 Win=17292 [TCP S] |
| 7 | 3.713980 | 10.234.0.239 | 10.234.2.116 | TCP | 1793 > 4444 [PSH, ACK] Seq=18 Ack=38 Win=17292 |
| 8 | 3.900264 | 10.234.2.116 | 10.234.0.239 | TCP | 4444 > 1793 [ACK] Seq=38 Ack=30 Win=64445 [TCP S] |

Uma das melhores maneiras de identificar o tráfego de um vírus ou worm é olhar para os dados sendo enviados através da rede. Vamos olhar para cada pacote em nossa captura no painel Packet Bytes na parte inferior da janela principal do Wireshark. Os dados do primeiro pacote parecem inocentes o suficiente, isso não uma informação muito usual, como você pode ver na figura abaixo.

| | | |
|------|---|--------------------|
| 0000 | 00 d0 59 aa af 80 00 01 96 3c 3f a8 08 00 45 00 | .Y.....<?...E. |
| 0010 | 00 28 08 ed 40 00 7f 06 d9 ac 0a ea 00 ef 0a ea | (. @..... |
| 0020 | 02 74 07 01 11 5c 76 be 16 50 cd 5a 82 b2 50 10 | .t...\\v. .P.Z..P. |
| 0030 | 43 b2 59 73 00 00 00 00 00 00 00 00 00 00 00 | C.Ys..... |

Frame (frame), 60 bytes

Movendo-se para o segundo pacote, no entanto (figura abaixo), vemos uma referência para o C:\WINNT\System32. Este é um dos mais importantes diretórios do sistema Windows 2000, pois contém muitos arquivos de sistema usados para carregar e executar o Windows. Vendo um pacote de rede referenciando esse local, é um sinal de problemas.

| | | |
|------|---|--------------------|
| 0000 | 00 80 ad d1 84 d7 00 d0 59 aa af 80 08 00 45 00 | Y....E. |
| 0010 | 00 3c 00 3a 40 00 80 06 e1 4b 0a ea 02 74 0a ea | .<.:@... K...t.. |
| 0020 | 00 ef 11 5c 07 01 cd 5a 82 b2 76 be 16 50 50 18 | ...\\...Z ..v..PP. |
| 0030 | fb db 73 31 00 00 0d 0a 43 3a 5c 57 49 4e 4e 54 | .s1... C:\WINNT |
| 0040 | 5c 73 79 73 74 65 6d 33 32 3e | \system3 2> |

Data (data), 20 bytes

Mais uma vez, o terceiro pacote não fornece nenhuma informação útil, mas o quarto mostra algo que pode ser motivo de preocupação, como mostrado na figura abaixo.

| | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| 0000 | 00 | d0 | 59 | aa | af | 80 | 00 | 01 | 96 | 3c | 3f | a8 | 08 | 00 | 45 | 00 | ..Y..... .<...E. |
| 0010 | 00 | 3a | 08 | ef | 40 | 00 | 7f | 06 | d9 | 98 | 0a | ea | 00 | ef | 0a | ea | .:.@..... |
| 0020 | 02 | 74 | 07 | 01 | 11 | 5c | 76 | be | 16 | 50 | cd | 5a | 82 | c6 | 50 | 18 | .t...\\V. .P.Z.P. |
| 0030 | 43 | 9e | a0 | 4d | 00 | 00 | 73 | 74 | 61 | 72 | 74 | 20 | 6d | 73 | 62 | 6c | C..M..st art msbl |
| 0040 | 61 | 73 | 74 | 2e | 65 | 78 | 65 | 0a | | | | | | | | | ast.exe. |

Data (data), 18 bytes

O painel Packet Bytes do quarto pacote mostra uma referência direta ao arquivo msblast.exe. Se você esteve envolvido com TI até a última parte de 2003, este nome deve saltar para fora em você imediatamente. Entretanto, se você não esteve, não se preocupe o Google é nosso amigo. A busca por este nome trará muita informação sobre o worm Blaster origem do problema no computador de Eddy.

Resumo

Neste cenário, fomos confrontados com um computador com software anti-vírus que não estava funcionando corretamente, o problema acabou por ser o worm Blaster.

Quando você suspeitar que você pode estar lidando com um vírus ou worm, você pode geralmente descobrir tudo o que você precisa saber sobre a ameaça de executar uma Pesquisa na Internet para os sintomas. Depois de identificar o vírus ou worm você poderá tratá-lo, você pode pesquisá-lo e aprender a combatê-lo.

Informações Confidenciais

Neste cenário você é o agente de segurança de rede em uma grande empresa multinacional corporação. Você acaba de ser alertado pelo seu superior que um empregado ouviu outros dois funcionários discutindo a possibilidade de vender informações da empresa. Sua tarefa neste cenário é para monitorar os computadores dos dois funcionários suspeitos para ver se você pode descobrir seus planos.

O Que Sabemos

Este cenário é baseado na especulação de outro funcionário. Enquanto nós ainda não podemos verificar se o que foi ouvido é verdadeiro ou se foi apenas retirado do contexto, sabemos que os dois funcionários em questão são muito esclarecidos em relação ao uso do computador , então nossas observações devem ser realizadas com o máximo de cuidado.

Farejando Através dos Fios

Porque não queremos que os nossos colaboradores com experiência em tecnologia saibam iremos monitorá-los, queremos ter absoluta certeza que os computadores que serão monitorados não mostrem sinais de estarem sendo observados. Por este motivo, vamos usar o espelhamento de porta, apesar de estarmos dentro da nossa própria rede. Definiremos um espelhamento separado e uma captura para cada computador a ser monitorado.

Análise

Ao longo do dia de trabalho destes dois trabalhadores, um monte de pacotes é gerado. Na maioria dos casos, esses pacotes são legítimos, portanto, o primeiro passo é procurar por tráfego que possa ser suspeito. Utilizar filtros tornam fácil a busca do tráfego, tais como DCEPRC, NetBIOS, ou ICMP, que não deveríamos ver, em circunstâncias normais. Eu apliquei esse filtro para a captura covertinfo.pcap, o resultado são dois pacotes a partir de um computador do empregado, conforme mostrado na figura abaixo.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|--------------|--------------|----------|---------------------|
| 1 | 0.000000 | 10.100.17.48 | 10.100.18.5 | ICMP | Echo (ping) request |
| 2 | 0.000015 | 10.100.18.5 | 10.100.17.48 | ICMP | Echo (ping) reply |

Estes pacotes podem parecer com os pacotes ICMP padrão, mas a origem e endereço de destino pertencem aos computadores dos nossos dois funcionários suspeitos. Por que eles estariam pingando-se durante o meio do dia?

Em seguida, como no cenário anterior vamos dar uma olhada no painel **Packet Bytes** para ver se conseguimos encontrar alguma coisa interessante neste pacote de ping. Ao fazê-lo, vemos algo um pouco alarmante, como mostrado na figura abaixo.

| | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|----------|
| 0000 | 00 | 15 | c5 | 37 | e1 | c1 | 00 | 0b | db | 71 | d7 | 39 | 08 | 00 | 45 | 00 | ...7.... | .q.9..E. |
| 0010 | 00 | a8 | 52 | 87 | 00 | 00 | 80 | 01 | af | d1 | 0a | 64 | 11 | 30 | 0a | 64 | ..R..... | ..d.0.d |
| 0020 | 12 | 05 | 08 | 00 | 1c | 2f | e4 | 0e | a3 | 0d | bc | 44 | 8d | 15 | 00 | 00 |//.. | ...D.... |
| 0030 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 6c | 75 | 65 | 43 | 68 | 61 | 74 | 31 | 30 |B1 | ueChat10 |
| 0040 | 2e | 31 | 30 | 30 | 2e | 31 | 37 | 2e | 34 | 38 | 20 | 20 | 20 | 54 | 72 | 61 | .100.17. | 48_Tra |
| 0050 | 6e | 73 | 66 | 65 | 72 | 20 | 61 | 6c | 6c | 20 | 6f | 66 | 20 | 74 | 68 | 65 | nsfer al | l of the |
| 0060 | 20 | 66 | 75 | 6e | 64 | 73 | 20 | 74 | 6f | 20 | 61 | 63 | 63 | 6f | 75 | 6e | funds t o accoun | |
| 0070 | 74 | 20 | 6e | 75 | 6d | 62 | 65 | 72 | 20 | 31 | 31 | 39 | 32 | 38 | 32 | 38 | t number | 1192828 |
| 0080 | 32 | 33 | 31 | 2d | 30 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 231-0 | |
| 0090 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | | |
| 00a0 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | | |
| 00b0 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | | | | | | | | | | |

Esse pacote ping está longe de ser normal. Por uma questão de fato, ele está carregando uma carga secreta com detalhes que os nossos funcionários cuidam para que não saibamos!

Resumo

A tecnologia utilizada neste cenário é conhecida como **Loki**, é um meio de envio de informações pela rede através de métodos ocultos. O termo **Loki** vem do primeiro projeto que incorpora dados em pacotes ICMP. Na nossa situação, o ICMP foi usado como veículo para transmitir mensagens entre nossos dois funcionários com intenção maliciosa.

A utilização de canais secretos de comunicação não é uma nova tecnologia, mas está evoluindo constantemente. Não é incomum encontrar dados escondidos em outros tipos de pacotes, bem como os cabeçalhos TCP e pacotes ARP. Lembre-se sempre do painel **Packet Bytes**, mesmo que você não o use freqüentemente, às vezes será a única forma de ver os segredos do conteúdo de um pacote.

O Ponto de Vista de um Hacker

Ao longo deste livro, nós olhamos as coisas do ponto de vista de um administrador de rede. Mas o que acontece quando um hacker com algum conhecimento de análise de pacotes decide dar uma espiada no que está trafegando em nossa rede? Neste cenário, assumimos a identidade de um hacker que tenta acessar informações sensíveis sobre a nossa rede local em nossa empresa.

O Que Sabemos

Mesmo que você seja um funcionário da empresa você está tentando violar o acesso a rede, você tem um acesso limitado aos recursos de rede. É uma rede padrão Ethernet, utilizando-se de alguns switchs e roteadores. Todos os computadores da rede rodando várias versões do Windows com privilégios de acesso definidos em uma base por usuário.

Farejando Através dos Fios

Alguns hackers desejam capturar as senhas de administradores de rede para pode acessá-la com privilégios. Outros simplesmente querem quebrar a segurança da mesma. Neste caso, queremos acessar um roteador na rede e depois fazer alguns danos sérios. Os administradores de rede estão sempre se deparando com essas coisas, por isso o processo deve ser simples o suficiente para monitorar a comunicação entre um administrador de rede e um roteador para interceptar uma senha.

Felizmente, tanto o administrador da rede e o roteador de destino estão na mesma sub-rede do computador que faremos o nosso ataque. Nós usaremos o Cain & Abel para criar o envenenamento de cache ARP entre o administrador de rede computador, 10.100.18.5, e o roteador de rede, 10.100.16.1, assim como que fizemos no Capítulo 2.

Análise

Depois de um tempo, conseguimos obter um ficheiro de captura que contém o tráfego telnet do administrador da rede o login no roteador. Para fins do presente cenário, a figura abaixo mostra apenas o tráfego relativo a esta sessão telnet.

| No. . | Time | Source | Destination | Protocol | Info |
|-------|----------|-------------|-------------|----------|--|
| 1 | 0.000000 | 10.100.18.5 | 10.100.16.1 | TCP | 3756 > telnet [SYN] Seq=0 Len=0 MSS=1460 |
| 2 | 0.001244 | 10.100.16.1 | 10.100.18.5 | TCP | telnet > 3756 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 |
| 3 | 0.001263 | 10.100.18.5 | 10.100.16.1 | TCP | 3756 > telnet [ACK] Seq=1 Ack=1 Win=65535 Len=0 |
| 4 | 0.003143 | 10.100.16.1 | 10.100.18.5 | TELNET | Telnet Data ... |
| 5 | 0.003201 | 10.100.18.5 | 10.100.16.1 | TELNET | Telnet Data ... |
| 6 | 0.004161 | 10.100.16.1 | 10.100.18.5 | TELNET | Telnet Data ... |
| 7 | 0.150539 | 10.100.18.5 | 10.100.16.1 | TCP | 3756 > telnet [ACK] Seq=4 Ack=6 Win=65530 Len=0 |
| 8 | 0.151553 | 10.100.16.1 | 10.100.18.5 | TELNET | Telnet Data ... |
| 9 | 0.351722 | 10.100.18.5 | 10.100.16.1 | TCP | 3756 > telnet [ACK] Seq=4 Ack=16 Win=65520 Len=0 |
| 10 | 1.285806 | 10.100.18.5 | 10.100.16.1 | TELNET | Telnet Data ... |
| 11 | 1.287056 | 10.100.16.1 | 10.100.18.5 | TCP | telnet > 3756 [ACK] Seq=16 Ack=5 Win=8192 Len=0 |
| 12 | 1.287275 | 10.100.16.1 | 10.100.18.5 | TELNET | Telnet Data ... |

Quando discutimos sobre o telnet no capítulo 6, notamos que ele normalmente usa texto em sua transmissão de dados. Telnet é geralmente usado remotamente para administrar switches, servidores e roteadores, como é aqui. A maioria desses dispositivos tem características que permitem a você fazer o login de forma segura, normalmente via SSH, mas isto é algo que os administradores de sistemas freqüentemente negligenciam. Uma vez que a comunicação está a passar visível, devemos ser capazes de encontrar as credenciais de login para este roteador com um pouco de paciência.

Telnet é um protocolo seqüencial, o que significa que tudo acontece em uma série definida. Portanto, a melhor maneira de localizar o processo de login é através de visualizarmos os dados dos pacotes Telnet um a um. Vemos o início do processo de autenticação claramente no pacote 8, conforme mostrado na figura abaixo.

| |
|--|
| Frame 8 (64 bytes on wire, 64 bytes captured) |
| Ethernet II, Src: Enterasy_31:4a:f0 (00:11:88:31:4a:f0), Dst: dell_37:e1:c1 (00:15:c5:37:e1:c1) |
| Internet Protocol, Src: 10.100.16.1 (10.100.16.1), Dst: 10.100.18.5 (10.100.18.5) |
| Transmission Control Protocol, Src Port: telnet (23), Dst Port: 3756 (3756), Seq: 6, Ack: 4, Len: 10 |
| Telnet |
| Data: Username: |

Se você olhar no painel **Packet Details** no campo Telnet, você verá que os dados que estão sendo passadas a partir do servidor é o pedido de um username. Os próximos pacotes de resposta ao servidor deve conter o nome, mas é um pouco mais complicado do que isso.

Como você pode ver na figura abaixo, o pacote 10 contém apenas a letra a. Isto não se parece com um nome de usuário típico, e não é.

| |
|---|
| 0000 00 11 88 31 4a f0 00 15 c5 37 e1 c1 08 00 45 00 . . . 1J... .7....E. |
| 0010 00 29 50 9c 40 00 80 06 73 65 0a 64 12 05 0a 64 .)P.@... se.d...d |
| 0020 10 01 0e ac 00 17 24 0f 27 9a e0 a9 10 7c 50 18\$. '.... P. |
| 0030 ff f0 cc 7a 00 00 61z..a |

O próximo pacote enviado pelo cliente para o servidor nos dá outra parte do quebra-cabeça, a letra d, conforme mostrado na figura abaixo. Nós estamos vendo a resposta do administrador sendo enviada ao servidor em um pacote de cada vez. Este processo prossegue por algum tempo até que possamos finalmente ver a palavra admin. Não muito original, né? É provavelmente o padrão.

| |
|---|
| 0000 00 11 88 31 4a f0 00 15 c5 37 e1 c1 08 00 45 00 . . . 1J... .7....E. |
| 0010 00 29 50 9f 40 00 80 06 73 62 0a 64 12 05 0a 64 .)P.@... sb.d...d |
| 0020 10 01 0e ac 00 17 24 0f 27 9b e0 a9 10 7d 50 18\$. '....}P. |
| 0030 ff ef c9 79 00 00 64y..d |

No pacote de 24 vemos um pedido de senha, como mostrado na figura abaixo.

| |
|--|
| Frame 24 (64 bytes on wire, 64 bytes captured) |
| Ethernet II, Src: Enterasy_31:4a:f0 (00:11:88:31:4a:f0), Dst: dell_37:e1:c1 (00:15:c5:37:e1:c1) |
| Internet Protocol, Src: 10.100.16.1 (10.100.16.1), Dst: 10.100.18.5 (10.100.18.5) |
| Transmission Control Protocol, Src Port: telnet (23), Dst Port: 3756 (3756), Seq: 23, Ack: 11, Len: 10 |
| Telnet |
| Data: Password: |

Mais uma vez, vemos pacotes que vão através do fio que nos dão a senha uma letra de cada vez (figura abaixo).

| | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| 0000 | 00 | 11 | 88 | 31 | 4a | f0 | 00 | 15 | c5 | 37 | e1 | c1 | 08 | 00 | 45 | 00 | ...1J....7....E. |
| 0010 | 00 | 29 | 50 | b2 | 40 | 00 | 80 | 06 | 73 | 4f | 0a | 64 | 12 | 05 | 0a | 64 | .)P.@... 50.d...d |
| 0020 | 10 | 01 | 0e | ac | 00 | 17 | 24 | 0f | 27 | a1 | e0 | a9 | 10 | 8d | 50 | 18 |\$.'.....P. |
| 0030 | ff | df | cb | 73 | 00 | 00 | 62 | | | | | | | | | | ...s..b |

Continuamos farejando esses pacotes até que tenhamos a senha completa, barrymanilow. Não só conseguimos capturar a senha do roteador, mas também aprendemos que o administrador da rede tem bom gosto musical!

Resumo

Neste ponto, nós temos tudo que precisamos para derrubar esta rede. Uma vez dentro da configuração do roteador, podemos eliminar sub-redes, trocar ips das interfaces, e fazer todas as sortes de outras coisas perniciosas que fará com que o administrador de rede tenha dores de cabeça severas.

O ponto deste cenário não é mostrar o que se pode fazer em um momento de raiva, mas sim demonstrar o poder que alguém com um pouco de conhecimento e um farejador de pacotes pode ter. Com o Wireshark e algumas outras ferramentas simples, nós efetivamente encontramos uma maneira de parar completamente todas as funções nesta rede.

10

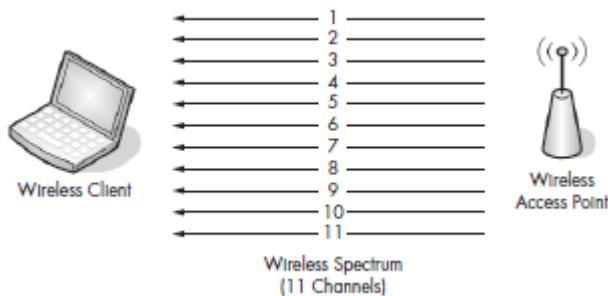
FAREJANDO PELO AR

O mundo das redes sem fio (wireless) é completamente diferente da tradicional rede (com fio). Quando consideramos a rede wireless precisamos levar em conta questões como freqüências, padrões e questões de segurança. Dadas estas considerações extras, você pode apostar que o processo de farejamento (sniffing) muda completamente.

Este capítulo é dedicado a explicar o processo de farejamento em redes sem fio no Windows. Enquanto discutimos o que faz o farejamento wireless original, veremos alguns exemplos do mesmo.

Farejando um Canal de Cada Vez

A primeira coisa a entender sobre o farejamento de tráfego em redes sem fio é que você só pode farejar um canal sem fio por vez. As redes sem fio nos Estados Unidos podem operar em um dos onze canais diferentes (mais informações estão disponíveis a nível internacional). Portanto, antes de capturar o tráfego de uma rede sem fio do cliente ou do ponto de acesso (WAP), você deve primeiro identificar qual o canal está transmitindo (figura abaixo).



A melhor maneira de descobrir qual o canal está sendo utilizado é utilizar o salto de canal. Quando você utiliza o salto de canais você simplesmente inicia uma captura de pacotes e alterna rapidamente de canal para canal até que você veja os dados relacionados ao que você está procurando. Embora o salto de canal não seja a melhor técnica ele funciona.

Interferência do Sinal em uma Rede Sem Fio

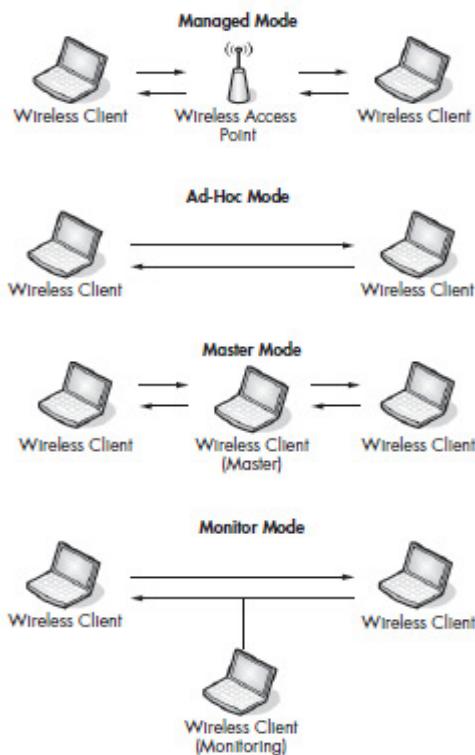
Infelizmente, às vezes não podemos confiar na integridade da comunicação sem fio. Porque os dados são enviados através do ar, é muito provável que alguma coisa vai interferir no sinal. As redes sem fio incluem recursos para lidar com interferências, mas eles não trabalham sempre. Portanto, estiver capturando pacotes sobre uma rede sem fio, preste muita atenção no seu ambiente para garantir que não existam fontes de interferência, como as grandes superfícies reflexivas, grandes objetos rígidos, microondas, aparelhos de 2,4 GHz sem fio, paredes grossas e superfícies de alta densidade.

Nessa mesma linha, tente chegar o mais perto possível do dispositivo que você está analisando. Você não pode esperar capturar todos os pacotes enviados por um dispositivo se você estiver em um andar acima dele.

Modos de Funcionamento de uma Placa de Rede Sem Fio

Antes do farejamento de pacotes em uma rede sem fio, é uma boa idéia se familiarizar com os diferentes modos em que uma placa de rede sem fio pode operar.

A maioria dos usuários utiliza cartões de rede sem fios gerenciados ou em modo ad-hoc, mas outros modos incluem o modo mestre e o modo monitor. Falarei sobre cada modalidade, uma representação gráfica da forma como cada um funciona é mostrado na figura abaixo.



Modo Gerenciado

O modo gerenciado é utilizado quando o cliente sem fio se conecta diretamente a um ponto de acesso sem fio (WAP). Nestes casos, o driver associado com a placa de rede sem fio depende do WAP para gerenciar as entradas do processo de comunicação.

Modo Ad-Hoc

O modo Ad-hoc é usado quando você tem uma configuração de rede sem fio em que dispositivos se conectam diretamente uns aos outros. Neste modo, dois clientes sem fio que desejam se comunicar um com o outro compartilham as responsabilidades da comunicação, coisas que um WAP normalmente lida.

Modo Master

Algumas placas de rede sem fio de ponta também suportam o modo Master. O modo Master permite que a placa de rede sem fio trabalha em conjunto com o software driver de modo a permitir que o computador funcione como um WAP para outros dispositivos.

Modo Monitor

Este é o modo mais importante para nossos propósitos. O modo Monitor é usado quando você quer que seu cliente sem fio não pare de transmitir e receber dados e só escutar os pacotes a ele destinados pelo ar. Para que o Wireshark possa capturar os pacotes em uma rede sem fio, sua placa de rede e o driver da mesma deve suportar o modo Monitor. Se você compra uma placa de rede sem fio para fins de análise, não se esqueça de verificar se ela suporta o modo Monitor (também conhecido como modo RFMON).

Farejando via Rede Sem Fios no Windows

Mesmo que você tenha uma placa de rede sem fio que suporta o modo Monitor, muitos driver dessas placas baseados no windows podem não permitir que você mude para este modo. Você precisa de um hardware extra para começar o trabalho a ser feito.

Configurando o AirPcap

AirPcap (da CACE Technologies <http://www.cacetech.com>) é projetado para superar as limitações que coloca Windows na análise de pacotes wireless. O AirPcap é um pequeno dispositivo USB (figura abaixo) que lembra um flash drive que foi concebido para capturar o tráfego sem fio. O AirPcap usa o driver WinPcap discutido no capítulo 3 e um utilitário de configuração especial no cliente.



O programa de configuração AirPcap é simples de usar, tem poucas opções configuráveis. Conforme mostrado na figura abaixo, o painel de controle do AirPcap oferece as seguintes opções:

Interface

Você pode selecionar o dispositivo que você está usando para a sua captura aqui. Algumas análise de cenários avançados podem exigir que você use mais de um dispositivo AirPcap para farejar simultaneamente em vários canais.

Blink Led

Clicando neste botão fará com que as luzes LED no dispositivo AirPcap pisquem. Isto é usado principalmente para identificação do adaptador específico que você está utilizando, se você estiver usando vários dispositivos AirPcap.

Channel

Neste campo, você seleciona o canal no AirPcap que deseja escutar.

Include 802.11 FCS in Frames

Por padrão, alguns sistemas reservam os últimos quatro bits de verificação dos pacotes wireless. Este check, conhecido como Frame Check Sequence (FCS), é usado para garantir que os pacotes não foram corrompidos durante a transmissão. A menos que você tenha um motivo específico para fazer o contrário, marque essa caixa para incluir o checksums FCS.

Capture Type

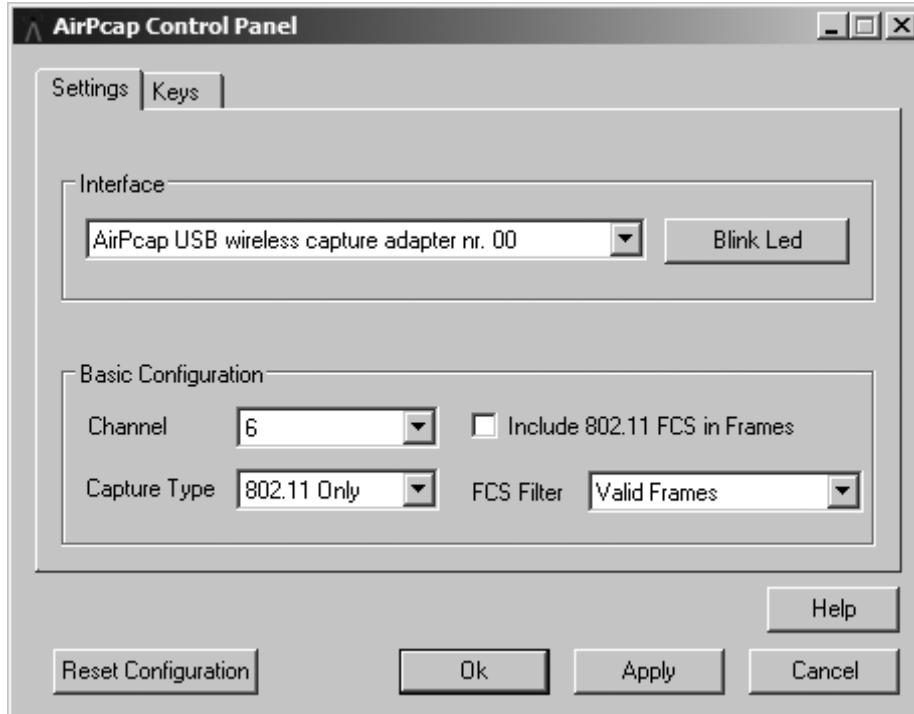
As duas opções aqui são 802.11 e 802.11 + Radio. Somente a opção 802.11 inclui o cabeçalho padrão 802.11 em todos os pacotes de captura. A opção 802.11 + Radio inclui este cabeçalho e também apresenta um cabeçalho radiotap, que contém informações adicionais sobre o pacote, como a taxa de dados, a freqüência, o nível do sinal e o nível de ruído. Escolha 802.11 + Radio para ver todas as informações de pacotes disponíveis.

FCS Filter

Mesmo que você desmarque a caixa ao lado das palavras Include 802.11 FCS in Frames, esta opção permite filtrar pacotes que o FCS determinou como corrompidos. Use a opção Valid Frames para mostrar apenas os pacotes que o FCS acha que foi recebido com êxito.

WEP Configuration

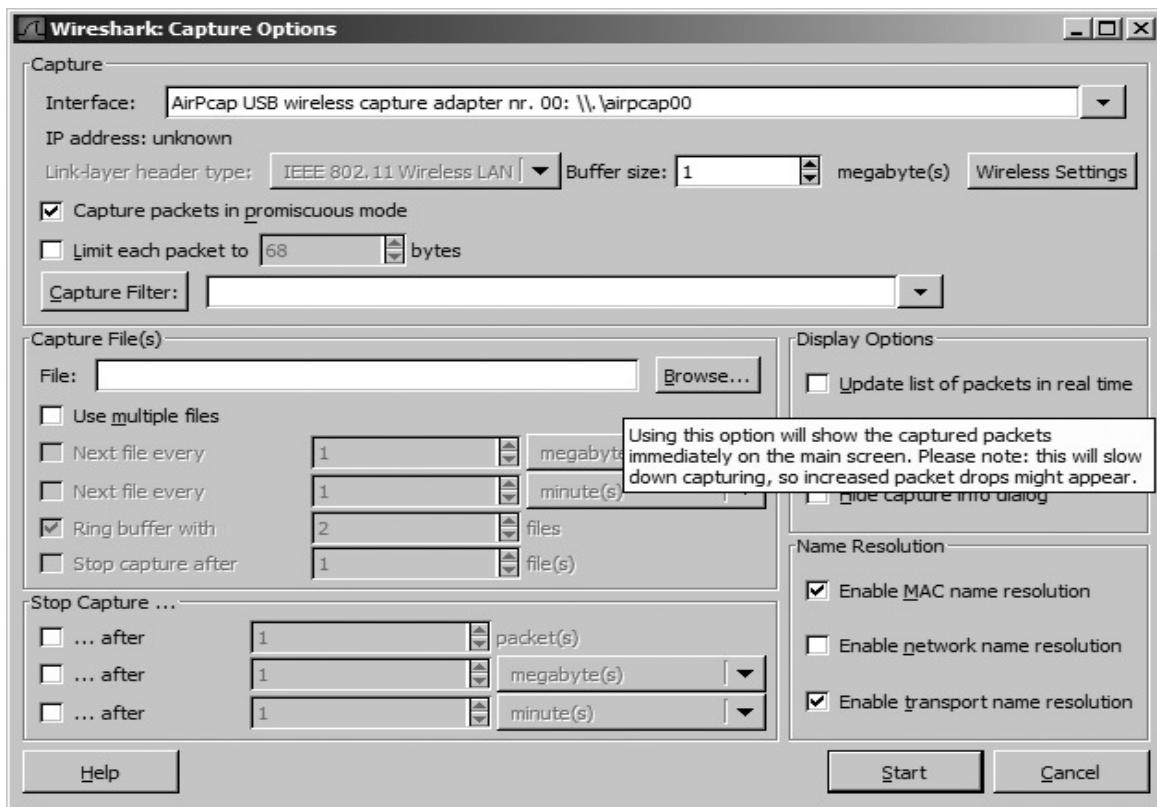
Esta área (acessível na guia Keys) permite que você insira as chaves de decriptografia WEP para as redes que você está farejando. Para ser capaz de interpretar os dados criptografados por WEP, você terá que inserir as chaves WEP correta para este campo.



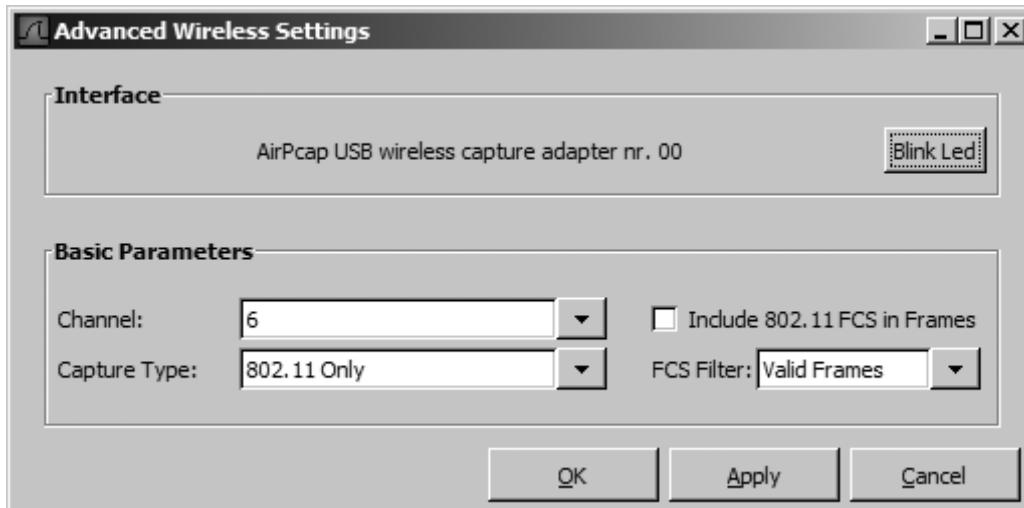
Capturando Tráfego com o AirPcap

Uma vez que você instalou e configurou o AirPcap, o processo de captura deverá ser familiar para você. Basta seguir estes passos:

1. No Wireshark, selecione **Capture > Options**.
2. Selecione o dispositivo AirPcap na caixa de seleção da interface, como mostrado na figura abaixo.



Tudo na tela deve ser familiar a você, exceto o botão de **Wireless Settings**. Clicando neste botão você obterá as mesmas opções que o utilitário AirPcap lhe deu, como mostrado na figura abaixo. Porque o Wireshark é totalmente integrado com o AirPcap, qualquer coisa configurada no utilitário cliente também pode ser configurado a partir do Wireshark.



3. Depois de ter tudo configurado ao seu gosto, comece a capturar os pacotes clicando no botão Iniciar.

Pacotes Extras 802.11

A principal diferença entre a estrutura do pacote de uma rede sem fio e de um pacote-padrão com fio é a adição de um cabeçalho 802.11. Este cabeçalho contém informações adicionais sobre o pacote e o meio utilizado para transmiti-lo, como mostrado na figura abaixo.

```

Frame 1 (132 bytes on wire, 132 bytes captured)
IEEE 802.11
    Type/Subtype: Beacon frame (8)
    Frame Control: 0x0080 (Normal)
        Duration: 0
        Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
        Source address: D-Link_0b:22:ba (00:13:46:0b:22:ba)
        BSS Id: D-Link_0b:22:ba (00:13:46:0b:22:ba)
        Fragment number: 0
        Sequence number: 1352
IEEE 802.11 wireless LAN management frame

```

Para examinar o pacote mostrado na figura acima mais de perto, abra o arquivo de exemplo 80211traffic.pcap. Vejamos alguns dos itens interessantes neste cabeçalho:

Type/Subtype Especifica o tipo ou subtipo do pacote 802.11 mostrado. O tipo pode ser de gestão, dados ou controle. Cada tipo pode ter um subtipo. Por exemplo, o subtipo de pacotes de gestão pode ser indicação de frame, solicitação de autenticação, ou aviso de encerramento.

Destination Address, Source Address, and BSS Id Esses campos contêm a origem, destino e endereços BSS do pacote.

Fragment Number and Sequence Number Esses números são usados para colocar os pacotes da rede sem fio em devida ordem, semelhante à maneira como o TCP organiza os fluxos de dados.

802.11 Flags

O cabeçalho do pacote de 802.11 também contém uma seção de **Flags** com ainda mais informações específicas, como mostrado na figura abaixo.

```

Flags: 0x0
    DS status: Not leaving DS or network is operating in AD-HOC mode
        .... .0.. = More Fragments: This is the last fragment
        .... 0... = Retry: Frame is not being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = Protected flag: Data is not protected
        0.... .... = Order flag: Not strictly ordered

```

A seção Flags inclui estes campos:

DS Status O campo Distribution Status (DS) é utilizado para determinar a forma que o pacote está sendo enviado. Se **From DS** é 1 e **To DS** é 0, então o pacote está sendo enviado do WAP para o cliente sem fio. Se os valores são o inverso, o pacote está viajando a partir do cliente sem fio para o WAP. Se ambos os números são 0, geralmente significa que o pacote está sendo difundido via broadcast do WAP.

More Fragments Este campo é utilizado quando os pacotes adicionais são necessários para ler o pacote que está sendo enviado.

Retry A opção Repetir indica se o pacote que está sendo transmitido a partir da tentativa original de transmissão (0) ou uma retransmissão (1).

PWR MGT Este campo indica se um cliente está entrando ou não em um estado de economia de energia.

More Data Este campo é utilizado por um WAP para informar o cliente que mais pacotes estão esperando para serem enviados a ele.

Protected Flag Este campo é usado para mostrar se o pacote está utilizando criptografia de dados.

Order Flag O campo Ordem é usado para informar ao destinatário que o pacote deve ser mantido em uma determinada ordem, que impede o receptor de reorganizar os pacotes a fim de aumentar o desempenho de transferência.

O Quadro Beacon

O quadro (frame) Beacon é um dos pacotes mais informativos em uma transmissão de uma rede sem fio. Um quadro Beacon é enviado como um pacote de difusão WAP através de um canal da rede sem fio notificando a todos os clientes sem fio que está a escutar informando que o WAP está disponível e definir os parâmetros a serem definidos quando se conectar a ele. Portanto, esse tipo de pacote de transmissão contém uma grande quantidade de informações úteis, como mostrado na figura abaixo.

The screenshot shows the Wireshark interface with a selected packet labeled "Frame 1 (132 bytes on wire, 132 bytes captured)". The "IEEE 802.11" section contains the following details:

- Type/Subtype: Beacon frame (8)
- Frame Control: 0x0080 (Normal)
- Duration: 0
- Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
- Source address: D-Link_0b:22:ba (00:13:46:0b:22:ba)
- BSS Id: D-Link_0b:22:ba (00:13:46:0b:22:ba)
- Fragment number: 0
- Sequence number: 1352

Below this, the "IEEE 802.11 wireless LAN management frame" section is also visible.

Algumas informações que você vê em um quadro beacon inclui o seguinte:

SSID parameter set Este é o SSID que o WAP está transmitindo.

Supported rates Este lista as taxas de throughput de dados suportados fornecidas pelo WAP e especifica se o protocolo utilizado é 802.11b ou 802.11g.

DS parameter set Este mostra o canal que o WAP está transmitindo.

Extended supported rates Este mostra outras taxas de throughput suportadas pelo WAP.

Vendor-specific information Esta seção mostra as informações específicas do fornecedor sobre o WAP, incluindo o fabricante do chipset, número da etiqueta,e o comprimento tag. (note que o fabricante do chipset não é sempre o mesmo que o fabricante WAP).

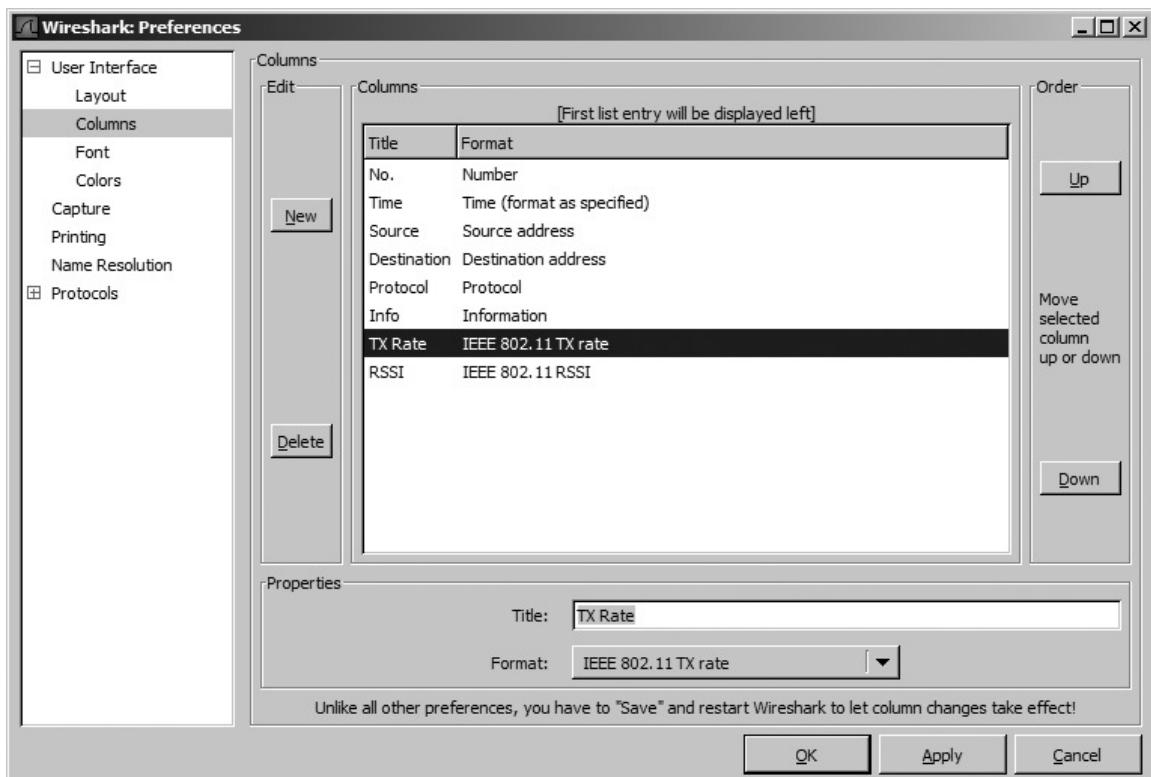
Colunas Wireless Específicas

O Wireshark tipicamente mostra seis colunas individuais no painel **Packet List**, todas devem ser familiares a você. No entanto, devido à sobrecarga adicionada ao analisar e interpretar os pacotes de uma rede sem fio, o Wireshark mostra as duas colunas mais úteis: **RSSI** e **TX Rate**. A coluna Received Signal Strength Indication (RSSI) mostra a freqüência de rádio (RF) a força do sinal de um pacote capturado, enquanto a coluna **TX Rate** mostra a taxa de dados de um pacote capturado, como mostrado na figura abaixo. Ambos os indicadores podem ser de grande ajuda quando você está solucionando problemas de conexões sem fio. De fato, mesmo se o seu software cliente wireless diz que você tem a força de sinal excelente, fazendo uma captação com essas colunas habilitadas os números mostrados podem não corresponder com essa informação.



Para adicionar essas colunas no painel **Packet List**, siga estes passos:

1. Escolha **Edit > Preferences**.
2. Navegue até a seção **Columns** e clique em **New**.
3. Digite RSSI no campo de **Title**, e selecione IEEE 802.11 RSSI no formato drop-down box.
4. Repita esse processo novamente para a coluna **TX Rate**, intitulando-a adequadamente e selecionando IEEE 802.11 TX Rate na seção **Format**. A figura abaixo mostra uma janela semelhante depois de você ter adicionado as informações para ambas as colunas.
5. Clique em **OK** na janela de **Preferences** para salvar suas alterações.
6. Reinicie o Wireshark para mostrar as novas colunas.



Filtros Wireless Específicos

Discutimos os benefícios dos filtros de captura no Capítulo 4. Em uma infra-estrutura com fio é muito mais fácil filtrar o tráfego que você deseja capturar, pois cada dispositivo tem o seu próprio cabo dedicado. Em uma rede sem fios, no entanto, todo o tráfego gerado por clientes sem fios coexiste em canais compartilhados, o que significa que uma captura de qualquer canal pode conter o tráfego de dezenas de clientes. Esta seção é dedicada a alguns filtros de pacotes que podem ser usados para ajudar a encontrar o tráfego que você deseja.

Filtrando o Tráfego de um Especifico BSS Id

Cada WAP em uma rede tem um nome de identificação exclusivo, denominado **Basic Service Set Identifier** (BSS Id). Este nome é enviado em cada quadro de gerenciamento e dados quando o ponto de acesso transmite. (Veja "Pacotes Extras 802.11 Packet").

Depois que você souber o nome do **BSS Id** que pretende examinar, tudo o que você realmente tem a fazer é encontrar um pacote que tenha sido enviado a partir desse WAP particular. O Wireshark mostra a transmissão WAP na coluna **Info** do painel **Packet List**, para encontrar esta informação normalmente é muito fácil.

Uma vez que você tem um pacote de um WAP em particular que você quer, encontre o campo do seu **BSS Id** no cabeçalho do 802.11, como mostrado anteriormente. Este é o endereço que o seu filtro se baseará.

Depois de ter encontrado o BSS Id MAC endereço (listados no painel **Packet Details**), você pode usar o filtro **wlan.bssid.eq 00:11:23:44:55:66** para mostrar apenas o tráfego que flui através desse WAP particular.

Filtrando Tipos de Pacotes Wireless Específicos

No início deste capítulo, discutimos sobre diversos tipos de pacotes sem fio que você pode ver em uma rede. Você muitas vezes precisará fazer filtros com base nesses tipos e subtipos. Use a Tabela abaixo como referência para ajudar a construir os filtros que você precisa.

Filtrando Tipos Específicos de Dados

Embora os pacotes de gerenciamento sem fio sejam muito importantes para alguns tipos de análise, a nossa análise exige que somente olhemos para os dados que são transmitidos através do ar, por exemplo, se precisamos rastrear os clientes sem fio desonestos ou identificar a possibilidade de divulgação de informações indesejáveis sobre a rede sem fios. Portanto, precisamos saber como filtrar esses pacotes de dados.

Para filtrar todos os pacotes de dados em uma captura de arquivo, use a captura filtro wlan.fc.type eq 2. (Se você faz referência a tabela abaixo, você verá que um quadro tipo 2 irá mostrar-nos de todos os dados relativos aos quadros de dados.)

A única desvantagem de usar esse filtro é que ele ainda permite a exibição de pacotes de dados nulos. Estes pacotes são usados pelo WAP em determinadas placas de rede sem fios para alertar a rede que eles estão prestes a mudar de canal. Se você não tem a necessidade de ver esses pacotes nulos, filtre-os através da expansão do filtro, criado anteriormente e eliminando o subtipo pacote **NUL**. O filtro ficará parecido com o mostrado abaixo:

(wlan.fc.type eq 2) and !(wlan.fc_subtype eq 4).

Diferenciar os dados não criptografados/criptografados é uma ótima maneira de identificar pacotes WAP clandestino em uma rede ou determinar se as informações sensíveis estão sendo enviadas em texto legível.

| Frame Type/Subtype | Filter Syntax |
|----------------------------|----------------------------|
| Management frames | wlan.fc.type eq 0 |
| Control frames | wlan.fc.type eq 1 |
| Data frames | wlan.fc.type eq 2 |
| Association request | wlan.fc.type_subtype eq 0 |
| Association response | wlan.fc.type_subtype eq 1 |
| Reassociation request | wlan.fc.type_subtype eq 2 |
| Reassociation response | wlan.fc.type_subtype eq 3 |
| Probe request | wlan.fc.type_subtype eq 4 |
| Probe response | wlan.fc.type_subtype eq 5 |
| Beacon | wlan.fc.type_subtype eq 8 |
| Disassociate | wlan.fc.type_subtype eq 10 |
| Authentication | wlan.fc.type_subtype eq 11 |
| Deauthentication | wlan.fc.type_subtype eq 12 |
| Action frames | wlan.fc.type_subtype eq 13 |
| Block ACK requests | wlan.fc.type_subtype eq 24 |
| Block ACK | wlan.fc.type_subtype eq 25 |
| Power save poll | wlan.fc.type_subtype eq 26 |
| Request to send | wlan.fc.type_subtype eq 27 |
| Clear to send | wlan.fc.type_subtype eq 28 |
| ACK | wlan.fc.type_subtype eq 29 |
| Contention free period end | wlan.fc.type_subtype eq 30 |
| NULL data | wlan.fc.type_subtype eq 36 |
| QoS data | wlan.fc.type_subtype eq 40 |
| Null QoS data | wlan.fc.type_subtype eq 44 |

Lembre-se do Flag protegido da seção "802.11 Flags", visto anteriormente, que é o Flag usado para identificar um pacote como sendo criptografados ou não criptografados. Nós basearemos nosso filtro neste flag.

Lembre-se que o bit de flag protegido está definido para 0 quando não está sendo usada criptografia e é definido como 1 se o pacote é criptografado com um protocolo como o WEP, WPA, TKIP, e assim por diante. Portanto, o uso de um filtro

wlan.fc.protected eq 0

mostrará-nos todos os pacotes que não são criptografadas. Da mesma forma, um filtro

`wlan.fc.protected eq 1`

mostrará apenas o tráfego criptografado.

Há centenas de formas de filtrar o tráfego capturado sem fio. Você pode ver vários desses filtros de captura de uma rede sem fios no wiki Wireshark na <http://wiki.wireshark.org>.

Uma Tentativa Ruim de Conexão

Agora vamos dar uma olhada em um cenário específico relacionado à análise de pacotes em uma rede sem fios. Neste cenário, Justin está tentando configurar o seu laptop para acessar a rede sem fios em seu escritório. Infelizmente, ela não está funcionando.

O Que Sabemos

A rede que Justin está tentando se conectar usa o método de autenticação compartilhada com criptografia WEP de canal. Justin simplesmente deve ser capaz de inserir essas configurações em seu cliente sem fio para se conectar, mas quando o faz, a conexão falha.

Farejando Através dos Fios

Nesta situação, capturando os pacotes do ar exige o mesmo pensamento do processo de captura de pacotes em uma conexão com fio. Como o processo parece falhar quando Justin tenta se conectar à rede sem fios, vamos capturar os pacotes neste momento. A melhor maneira de fazer isso é usando o dispositivo AirPcap, configurado para um canal.

Análise

Uma vez que temos de olhar para uma captura em uma rede sem fios, não sabemos o que é uma autenticação bem-sucedida e como se parece essa seqüência. Vamos olhar para um arquivo de captura deste processo quando ele está funcionando corretamente, abra o exemplo de arquivo SuccessfulWEPAuth.pcap, que mostra uma seqüência bem-sucedida na rede de Justin.

A rede sem fio de Justin está configurada utilizando a chave WEP compartilhada de segurança. A chave **Wired Equivalent Privacy** (WEP) é um código alfanumérico ou hexadecimal que serve como um tipo de senha usada para criptografar a comunicação entre um WAP e um cliente sem fio (ou seja, o usuário que está tentando se conectar ao wireless de rede). Para se conectar a um WAP, o cliente sem fio deve primeiro completar o processo de descoberta e resposta com o WAP, a fim de verificar a correta chave WEP que está sendo usada. Esta tentativa de descoberta e resposta começa no pacote 4 do arquivo de captura, como mostrado na figura abaixo.

```
□ Challenge text
  Tag Number: 16 (Challenge text)
  Tag length: 128
  Tag interpretation: Challenge text: D4ABB116F5B6C6CF1EC74B95A5389E7D341CC3D87A2F9F95...
```

O WAP responde à tentativa de conexão através do envio de uma chave de descoberta para o cliente. Esta chave é uma string encriptada de texto que deve ser decifrada pelo cliente (com a chave WEP adequada) e, em seguida enviada de volta para o WAP, como mostrado na figura abaixo.

```
□ WEP parameters
  Initialization Vector: 0x0cf79e
  Key Index: 0
  WEP ICV: 0x409d2512 (not verified)
  Data (147 bytes)
```

No pacote de 6 o cliente sem fio envia de volta a chave desencriptada, e o WAP responde com uma mensagem indicando que o processo de autenticação foi bem-sucedido, como mostrado na figura abaixo.

```

    □ Fixed parameters (6 bytes)
        Authentication Algorithm: shared key (1)
        Authentication SEQ: 0x0004
        Status code: successful (0x0000)

```

Finalmente, após uma autenticação bem-sucedida, o cliente pode transmitir um pedido de associação, receber um aviso, e se conectar, como mostrado na figura abaixo.

| No. ▾ | Time | Source | Destination | Protocol | Info |
|-------|----------|-------------------|-------------------|----------|--|
| 10 | 0.145465 | GemtekTe_30:b0:af | Enterasy_6b:68:30 | IEEE 8 | Association Request, SN=44, FN=0, SSID: "DENVEROFFICE" |
| 11 | 0.145839 | | GemtekTe_30:b0:af | IEEE 8 | Acknowledgement |
| 12 | 0.148466 | Enterasy_6b:67:28 | Broadcast | IEEE 8 | Data, SN=1390, FN=0 |
| 13 | 0.149090 | Enterasy_6b:68:30 | GemtekTe_30:b0:af | IEEE 8 | Association Response, SN=1391, FN=0 |
| 14 | 0.149464 | | Enterasy_6b:68:30 | IEEE 8 | Acknowledgement |

Agora que sabemos como uma conexão WAP se parece, vamos olhar para o arquivo de captura da tentativa de conexão de Justin. Como podemos ver no pacote 3 (mostrado na figura abaixo), o WAP envia a chave de descoberta para o computador de Justin, assim sabemos que os dois dispositivos podem ver um ao outro.

```

    □ IEEE 802.11 wireless LAN management frame
        □ Fixed parameters (6 bytes)
            Authentication Algorithm: shared key (1)
            Authentication SEQ: 0x0002
            Status code: successful (0x0000)
        □ Tagged parameters (130 bytes)
            □ Challenge text
                Tag Number: 16 (Challenge text)
                Tag length: 128
                Tag interpretation: challenge text: DEFC7D3DDCBC57CC85FFCE1687FAC6E5528E4DD0619BF5B1...

```

O Pacote 5 (figura abaixo) mostra o cliente sem fio enviando a sua resposta ao servidor, o que nos mostra que estes dispositivos estão tentando se comunicar.

```

Destination address: Enterasy_6b:68:30 (00:11:88:6b:68:30)
Source address: GemtekTe_30:b0:af (00:14:a5:30:b0:af)
BSS Id: Enterasy_6b:68:30 (00:11:88:6b:68:30)
Fragment number: 0
Sequence number: 43
□ WEP parameters
    Initialization vector: 0x26709d
    Key Index: 0
    WEP ICV: 0xc800a5b7 (not verified)
Data (147 bytes)

```

Neste ponto de progressão, deveríamos ver uma resposta do WAP, confirmando que o processo de autenticação foi bem sucedido. Mas em vez disso, vemos outra coisa, como mostrado na figura abaixo. A autenticação falha.

```

    □ IEEE 802.11 wireless LAN management frame
        □ Fixed parameters (6 bytes)
            Authentication Algorithm: Unknown (58901)
            Authentication SEQ: 0x884c
            Status code: Received an Authentication frame with authentication sequence transaction sequence number out of expected sequence

```

A mensagem enviada a partir do WAP para computador de Justin nos diz exatamente o que está acontecendo: os números de seqüência estão fora de ordem. Isso significa que a resposta da chave de descoberta que o computador de Justin forneceu não foi correta - portanto, a chave WEP usada para decriptografar a chave de descoberta não foi enviada ou não foi digitada corretamente.

Resumo

A triste verdade sobre como solucionar problemas de rede sem fio é que o software wireless cliente normalmente não relata problemas específicos: quando o cliente conecta ou não. Felizmente, as técnicas de análise de pacotes em redes sem fios permitem-nos ver exatamente o que está acontecendo e solucionar esses problemas de forma mais eficiente em redes sem fio.

Considerações Finais

As redes sem fio estão se tornando um marco no ambiente corporativo. Como o foco está mudando para as redes sem fio, devemos ser capazes de solucionar problemas em redes com fios e sem fios. As competências e os conceitos ensinados neste capítulo devem ajudar você a entender as complexidades da resolução de problemas em uma rede sem fio através da análise de pacotes.