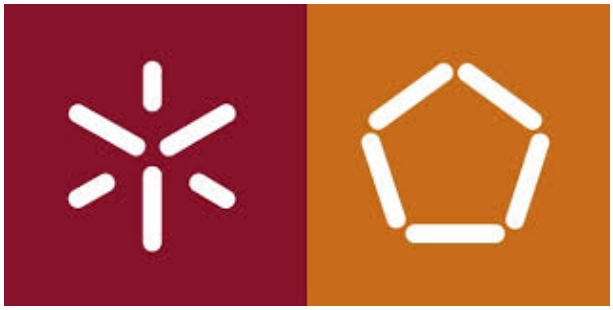
Universidade do Minho



Trabalho Prático

Análise de tráfego

Mestrado Integrado em Engenharia Informática

Segurança em Redes

(1º Semestre - 2018/2019)

a70565	Bruno Arieira
a73883	Cesário Perneta
a73974	Daniel Vieira
a78494	José Dias

1 de Dezembro de 2018

Resumo

"Over the last two decades the Internet protocol suite (also called the TCP/IP protocol suite) has come to be the most ubiquitous form of computer networking. Hence, the most widely used transport protocols today are TCP and its companion transport protocol, the User Datagram Protocol (UDP)."

Este trabalho prático foi realizado no âmbito da unidade curricular Segurança em Redes, e tem como principal objetivo a interação e experiência, usando a ferramenta *Wireshark*, com alguns conceitos e técnicas relevantes para **Análise de Tráfego**.

Conteúdo

1 Introdução

Neste quarto trabalho prático, temos como principal objetivo aplicar o conhecimento adquirido nas aulas de Seguranças em Redes, relativamente á matéria lecionada sobre *Análise de Tráfego* com principal objetivo de ganhar experiência com uma ferramenta de captura e análise tráfego (**Wireshark**), adquirir competências na definição e implementação de uma estratégia em relação á análise de tráfego em rede.

Em conformidade com o enunciado proposto, para a realização deste trabalho, inicialmente é necessária a instalação da ferramenta *Wireshark* e depois descarregar o exemplo do ficheiro de captura de tráfego, disponibilizado por o professor. Como nos foi pedido para este relatório ser claro e objetivo, decidimos explicitar todas as anomalias, descrevendo na secção Desenvolvimento alguns exemplos. Antes de começar a fazer o pedido no enunciado do trabalho, tivemos primeiramente que estudar e analisar o tráfego recolhido, usando as funções estatísticas da ferramenta, começando por identificar as secções.

Para o desenvolvimento deste trabalho, foi delineado que todos os elementos deviam analisar, estudar e entender os conceitos implícitos para a análise de tráfego por forma a facilitar a resolução do trabalho proposto. Com a devida consolidação dos termos indispensáveis, passamos á discussão e elaboração das tarefas propostas, onde todos os elementos trabalharam de forma uniforme.

2 Contextualização

Um **protocolo** pode ser definido como um conjunto regras que governam a sintaxe, semântica e sincronização da comunicação. Os protocolos podem ser implementados pelo hardware, software ou por uma combinação dos dois, possibilitando uma conexão, comunicação, transferência de dados entre dois sistemas.

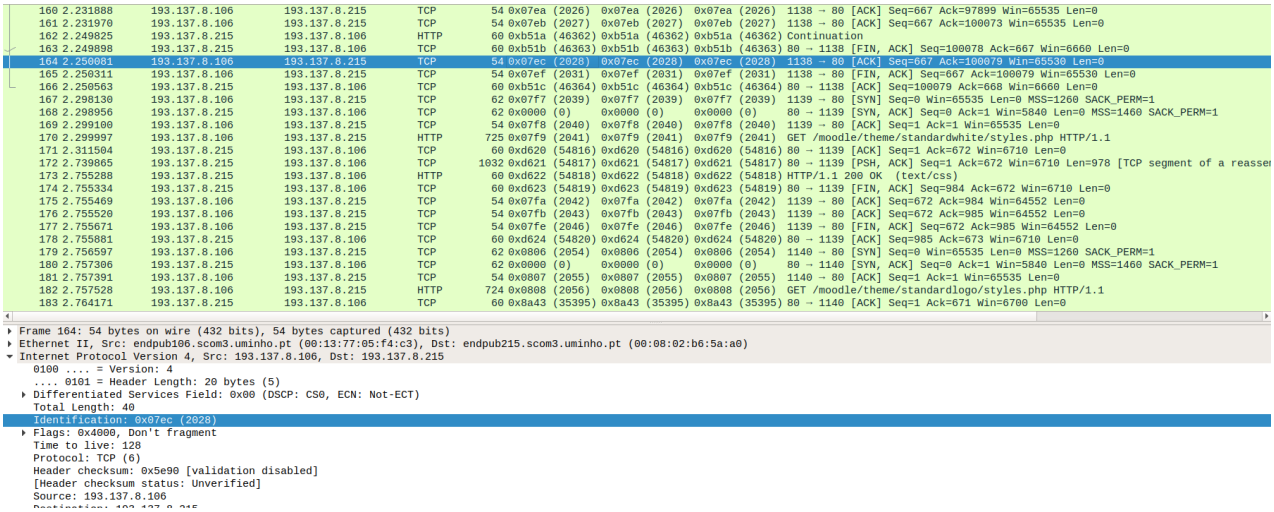
De seguida, apresentámos alguns protocolos definidos segundo as camadas do modelo OSI(**Open System Interconnection**) em que operam, sendo importante para as sessões que estão implícitas na captura do tráfego.

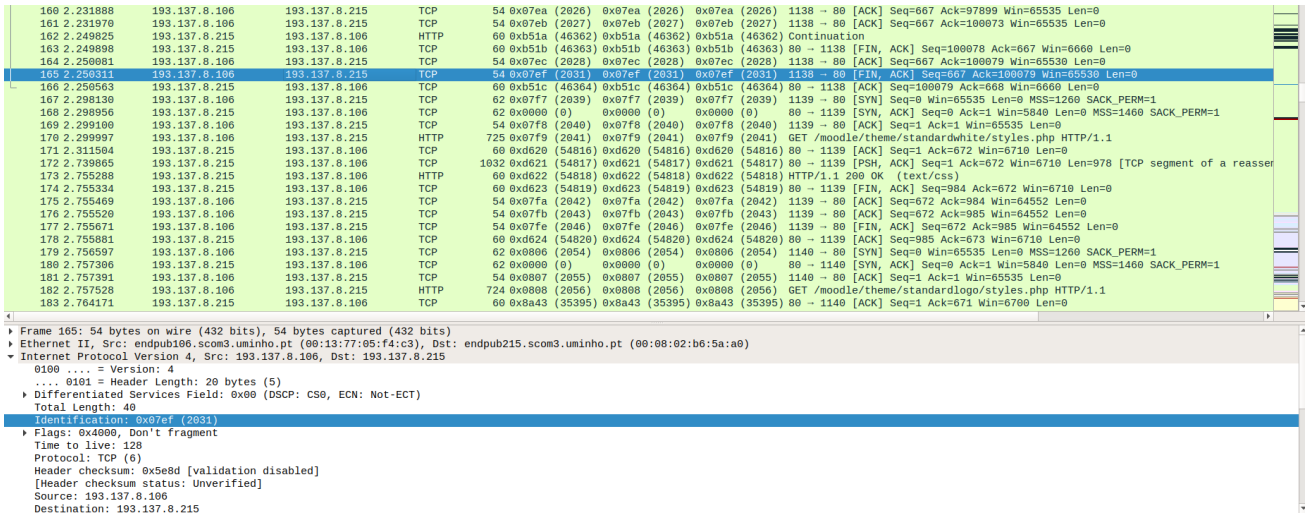
- **Camada da ligação de dados** constitui os protocolos de ligação lógica ou ligação de dados, constituem o primeiro nível de troca ordenada controlada e fiável de dados entre sistemas interligados por meio de uma ligação física, ou seja, também faz a deteção e, caso seja necessária, a correção de erros que possam acontecer na camada física.
  - **ATM (Asynchronous Transfer Mode)** é uma arquitetura de rede baseada na transferência de dados em pacotes de tamanho fixo;
  - **Ethernet (IEEE 802.3)** é arquitetura de interconexão que define cabeamento e sinais elétricos para a camada física, em formato de pacotes e protocolos para a subcamada Media Access Control (MAC).
  - **HDLC** é um protocolo da camada de ligação de dados síncronos orientados a bits que garante a transmissão de dados sem erros para os destinos adequados e controla a velocidade de transmissão de dados.
- **Camada de Rede** é responsável por o encaminhamento dos pacotes entre origem e destino, mesmo que estes tenham que passar por diversos nós intermediários durante o percurso, e pelo controlo de congestionamento e ainda a contabilização do número de pacotes utilizados pelo utilizador.
  - **IPv4** é um protocolo, sem conexão, de comunicação usado entre duas ou mais máquinas em rede para encaminhamento dos dados que utiliza endereços de 32 bits.
  - **IPv6** é a versão mais atualizada po protocolo IP em que são usados endereços de 128 bits.
- **Camada de Transporte** transporta e regula o fluxo de informações da origem até o destino, de forma confiável com a principal função de fornecer controlo fim-a-fim usando janelas e oferecendo fiabilidade nos números de sequência e nas confirmações.
  - **TCP (Transmission Control Protocol)** é um protocolo que fornece garantia na entrega de todos os pacotes entre um PC emissor e um PC recetor.
  - **UDP (User Datagram Protocol)** é um protocolo em que o emissor envia uma determinada informação e a máquina recetora recebe essa informação, não existindo qualquer confirmação dos pacotes recebidos.
  - **Telnet** é protocolo standard de Internet que permite a interface de terminais e de aplicações através da Internet.
- **Camada de Aplicação** é responsável por prover serviços para aplicações de modo a separar a existência de comunicação em rede entre processos de diferentes computadores.
  - **HTTP (Hypertext Transfer Protocol)**:é um protocolo subjacente usado pela *World Wide Web* e esse protocolo define como as mensagens são formatadas e transmitidas e quais ações os servidores da Web e os *browsers* devem executar em resposta a vários comandos.
  - **SNMP(Simple Network Management Protocol)** é um protocolo criado para facilitar a gestão e monitorização de dispositivos em redes IP.
  - **FTP (File Transfer Protocol)** é um protocolo de rede padrão usado para a transferência de ficheiros entre um cliente e um servidor em uma rede de computadores.

3 Desenvolvimento

De seguida apresentar-se-ão vários exemplos de *falhas* na rede que podem comprometê-la (estamos a usar o ficheiro **ExemploTrafego1.pcap**):

3.1 Falha 1





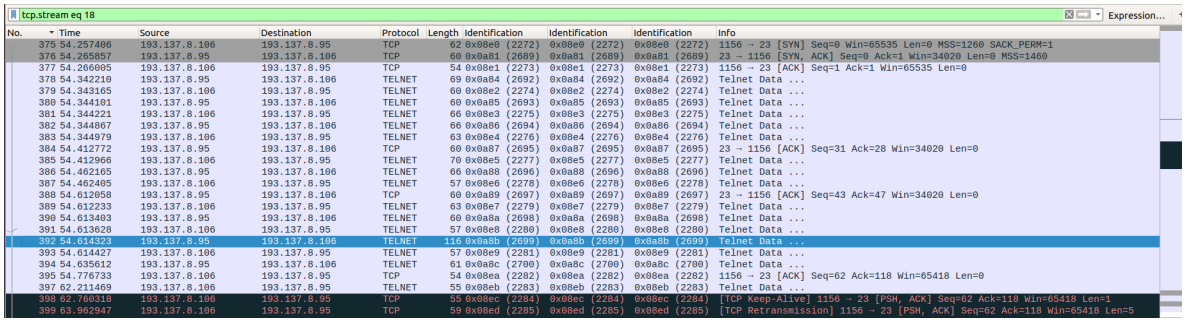
Como podemos verificar, nas 2 imagens anteriores, há uma possível perda de pacotes, visto que para o mesmo IP(193.137.8.106), o correspondente *identification* tem um interregno entre o **2028** e o **2031**. Portanto é de suspeitar que se perderam 2 pacotes. Para isso podemos verificar se existe um *"identification"* igual a 2029 e 2030 para o mesmo IP. Para tal, inserimos os seguintes filtros:**ip.id==0x07ed** e **ip.id==0x.07ee** sendo que estes números representam em hexadecimal os valores anteriores citados.

Como podemos ver nas figuras anteriores, há perda de pacotes, o que poderá levar a consequências nefastas para a rede como por exemplo:

- **Ataque de pacotes**: Caso haja um utilizador mal intencionado, é demasiado fácil controlar um *router* fazendo com que este descarte pacotes de um determinado fluxo.
- **Período sazonal de negócios**: Durante um período destes, uma rede pode não estar preparada para uma fluxo tão elevado, devido a um mau planeamento dela ou deficiência nos recursos. E é aqui que podem entrar os *hackers*. Eles têm tendência a entrar em grupo e caso uma rede não consiga priorizar o tráfego que é importante, um deles poderá entrar por um *backdoor* de baixa prioridade porque a rede mostra todas as informaçoes sem distinguir o grau de importância. Além disso, os *hackers* poderão inundar uma porta que tem uma largura de banda insuficiente, fazendo com que a rede aceite qualquer pacote, mesmo os infetados, visto que ela os irá aceitar porque está desesperada para manter o mesmo desempenho da sua rede como antes.

3.2 Falha 2

Sabendo que através do Wireshark é muito fácil descobrir uma palavra-passe de algum *login* que foi efetuado durante a captura do tráfego, caso a rede não tenha o mínimo exigido de segurança, optámos por procurar algum pacote que mostrasse tal.



Através da pesquisa do pacote com tal característica, foi relativamente fácil descobrir a palavra passe de um determinado utilizador. É uma grande vulnerabilidade que poderá ser colmatada usando criptografia e para isso poderia-se usar o **SSH** visto que este protocolo usa criptografia enquanto que o **Telnet** não, sendo que são 2 protocolos semelhantes, e têm como objetivo conseguir um acesso remoto a um servidor.

3.3 Notas do Desenvolvimento

Durante a pesquisa de anomalias tentamos verificar as várias referidas na aula, embora tenhamos apenas notado as possíveis falhas de segurança nas representadas acima. Algumas das anomalias que pesquisamos embora sem sucesso foram *ARP spoofing*, *ping flooding* e *SYN flooding*.

Com o objetivo de encontrar exemplos de ***ARP Spoofing*** no ficheiro referido pesquisamos por segmentos do *three way handshake* do `TCP`. Nestes comparamos o endereço da *Layer 2* do pacote com a *flag* `SYN` e o do pacote com a *flag* `ACK`. Pois em casos de *ARP Spoofing* estes mostrariam valores diferentes. Não tivemos sucesso em encontrar qualquer exemplo no ficheiro referido.

Para identificar ***SYN flooding*** é mais simples. Neste caso é apenas necessário pesquisar por grandes sequencias de pacotes `TCP` com a flag `SYN` *set* em que não é devolvido um `ACK` posteriormente, ou seja não completando o já referido *three way handshake* do `TCP`. Caso isto aconteça o servidor ficaria á espera do `ACK` da origem mas este nunca chegará gerando congestionamento na rede.

Já na identificação de ataques do tipo ***Ping flooding*** pesquisamos por sequencias de pacotes `ICMP` do tipo *Echo (ping) request* e *Echo (ping) reply* com tamanhos anormais, normalmente com tamanho máximo. O tamanho *standard* é 56 bytes para sistemas Linux e Mac, e no caso de sistemas Windows então 32 bytes, embora podendo ser ligeiramente superior devido á inclusão do *HEADER*. Neste caso embora tendo encontrado sequencias de pacotes dos tipos referidos estes demonstravam tamanhos normais para os sistemas operativos em causa.

Após estas verificações chegamos á conclusão que não foram efetuados ataques como os referidos acima á rede em questão.

4 Conclusão

Acabado este trabalho, há que refletir sobre os resultados obtidos e a forma de obter os mesmos. Devido ao conhecimento que adquirimos nas aulas bem como grandes pesquisas na Internet, o capítulo **Contextualização** em que falamos um bocado de cada protocolo, foi feito sem nenhum problema relevante.

Já em relação ao trabalho propriamente dito, que consistia em efetuar o *sniffing* a uma captura de tráfego dada pelo professor, inicialmente sentimos algumas dificuldades, pois não estávamos a conseguir relacionar os pacotes com a segurança do sistema. Depois de uma melhor compreensão do enunciado e das explicações do professor, conseguimos prosseguir sem problemas relevantes.

De modo a que o trabalho corra de forma fluída, dividimos o mesmo entre os elementos do grupo, sendo que no final nos reunimos para fazer uma apreciação global do mesmo e realizar as mudanças necessárias de modo a obter um resultado mais satisfatório.

Para finalizar, consideramos que este trabalho nos ajudou a aprofundar os conhecimentos técnicos que já possuíamos do Wireshark, permitindo-nos agora saber como procurar as vulnerabilidades de uma rede através desta poderosa ferramenta. Achamos que este trabalho atingiu os objetivos requeridos e estamos satisfeitos com o resultado final.

Referências