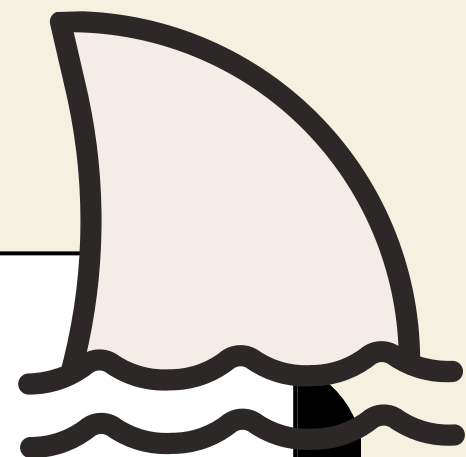


WIRESHARK



01

O QUE É



05

FILTRO 3



02

IMPORTÂNCIA



06

FILTRO 4



03

FILTRO 1



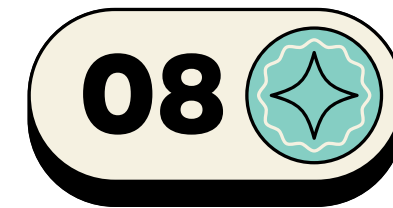
07

FILTRO 5



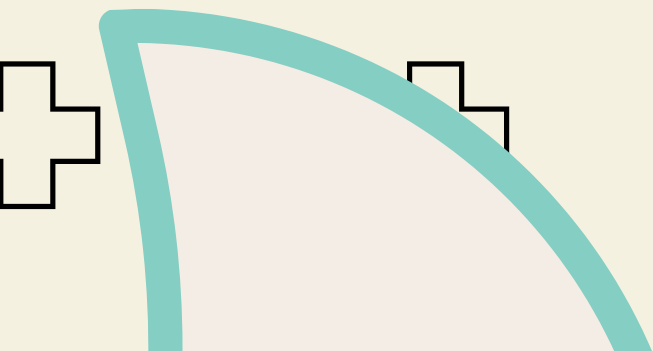
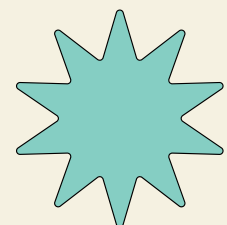
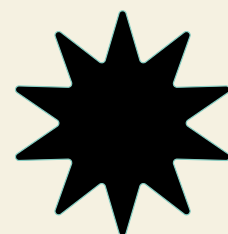
04

FILTRO 2



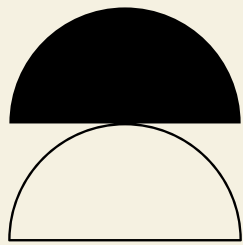
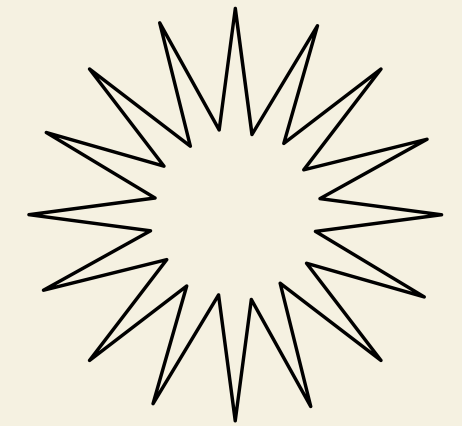
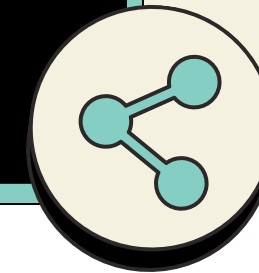
08

FILTRO 6





O QUE É

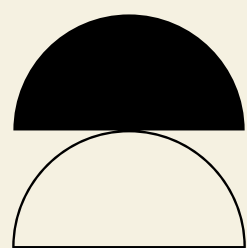
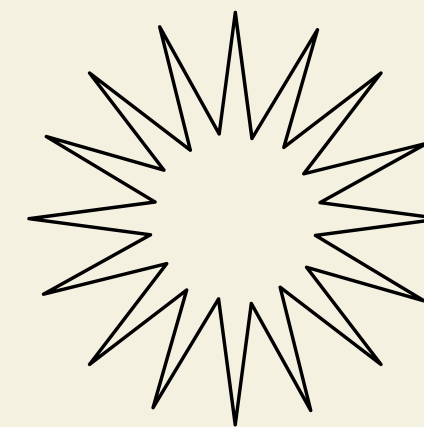


**Ferramenta de análise de rede que captura e
exibe os dados trafegando na rede em tempo
real. É utilizado para monitorar, analisar,
solucionar problemas e entender o
comportamento da rede.**





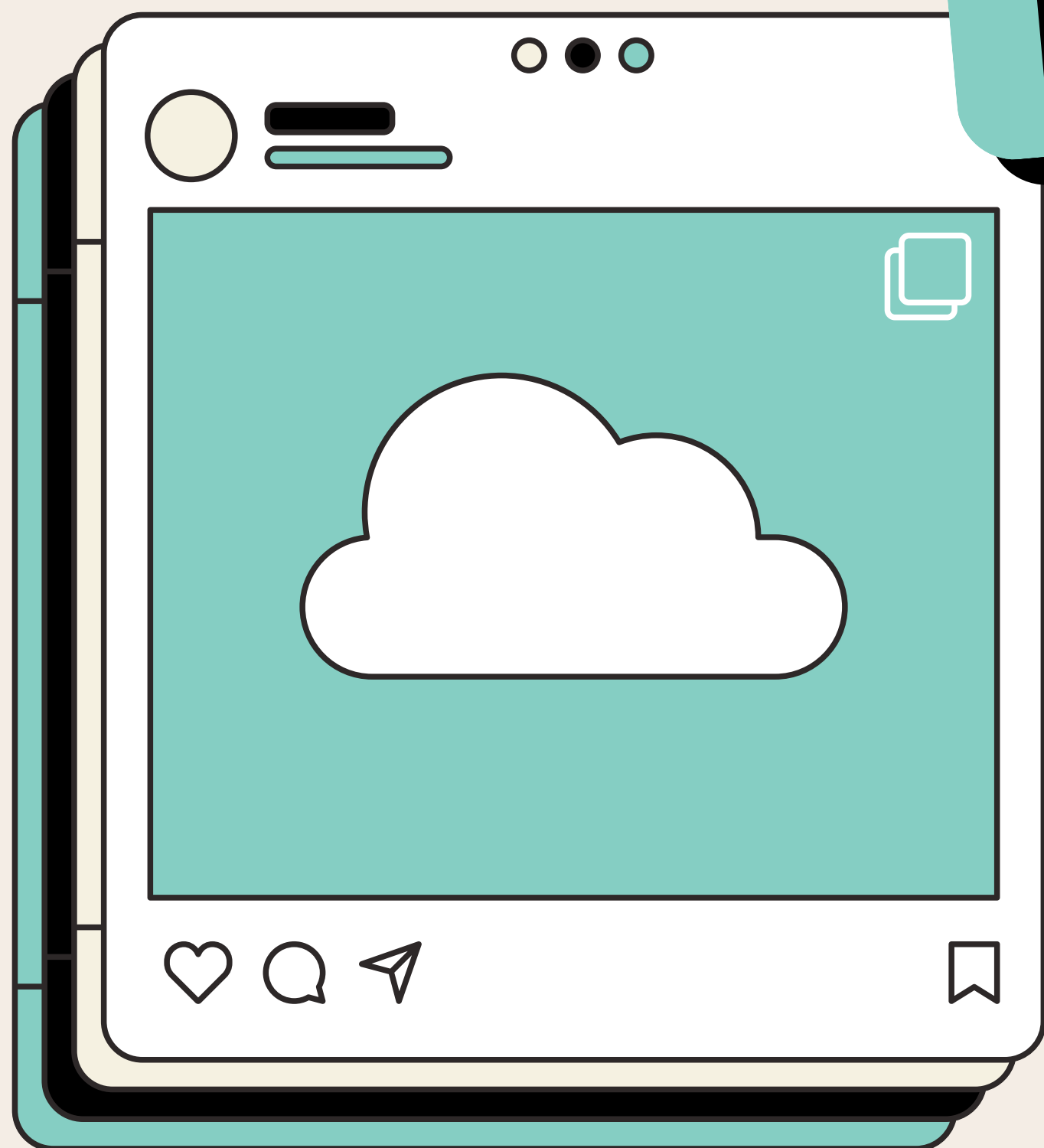
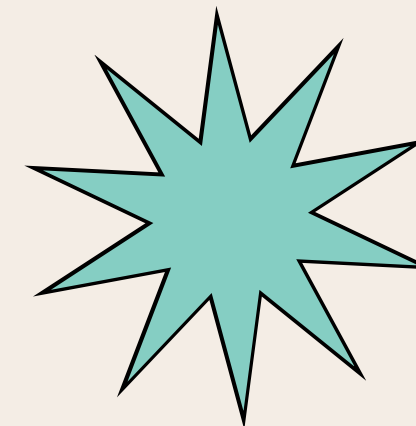
IMPORTÂNCIA



É essencial para administradores de rede, engenheiros de segurança e desenvolvedores para detectar problemas de rede, monitorar tráfego e verificar se há atividades suspeitas ou maliciosas.



FILTROS



01



tcp.analysis.flags

02

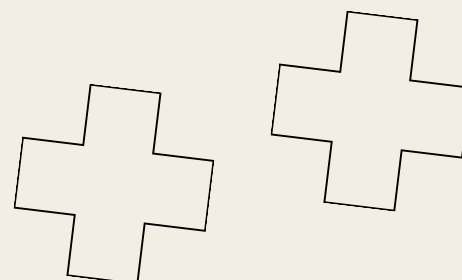


dns or http

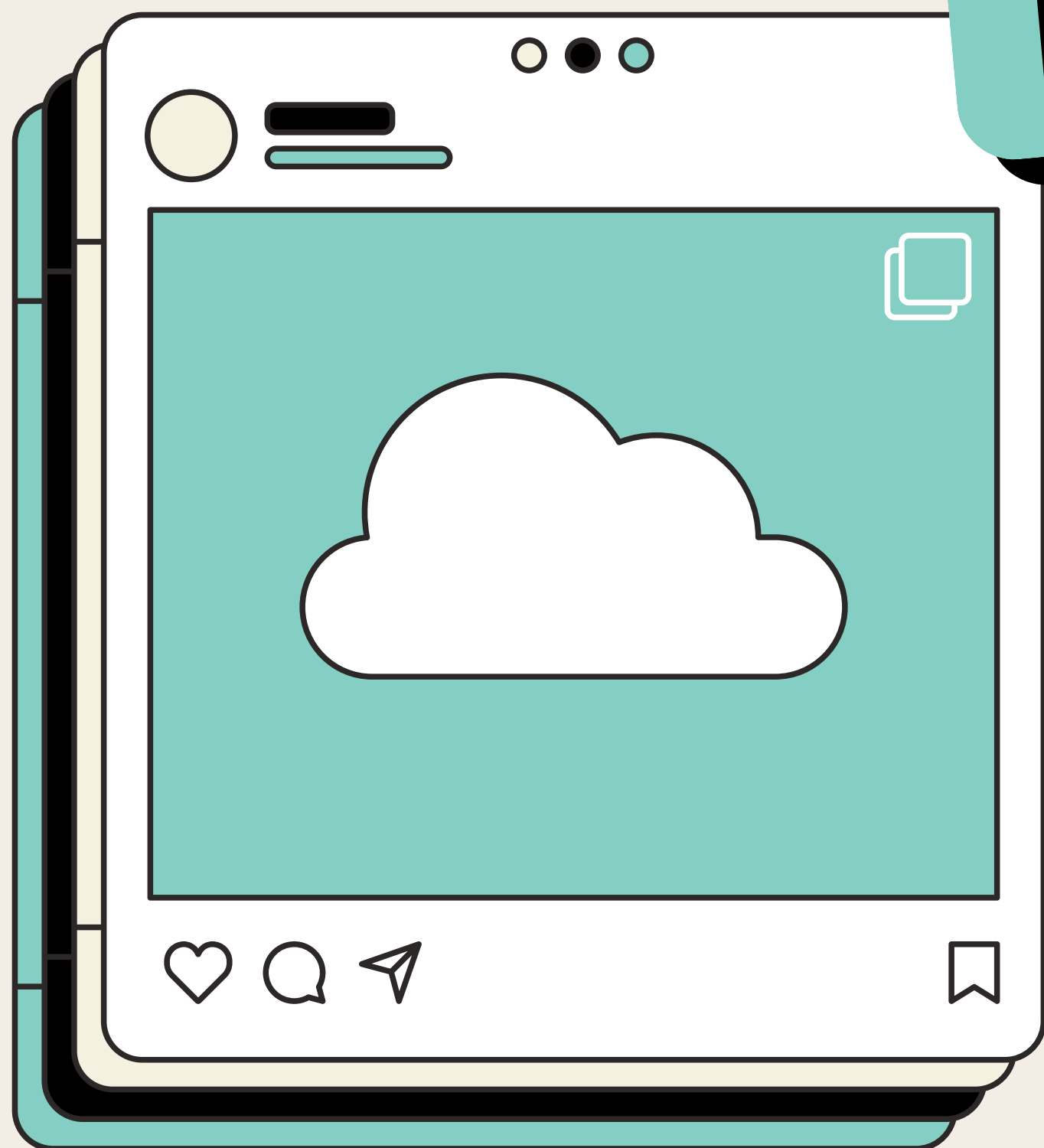
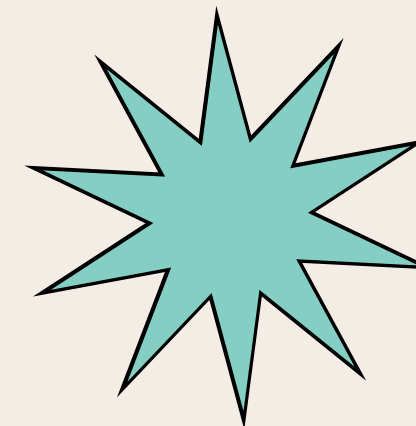
03



**!(arp or icmp
or dns)**



FILTROS



04



tcp.analysis.retransmission

05

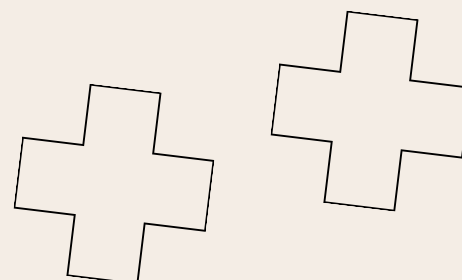


**frame contains "attachment"
or frame contains "pdf"**

06



**ip.addr==127.0.
0.1**



FILTRO 1



TCP.ANALYSIS.FLAGS

Protocolo TCP:

- **Transmission Control Protocol (TCP)** é um protocolo de transporte confiável que garante a entrega ordenada de pacotes de dados entre um remetente e um destinatário.
- **TCP** utiliza um mecanismo de controle de fluxo e de erro para garantir que os dados sejam entregues corretamente.
- **TCP** estabelece uma conexão através do processo conhecido como "three-way handshake" e termina a conexão de forma ordenada.

TEORIA

FILTRO 1



TCP.ANALYSIS.FLAGS

- **Flags TCP:** Os flags são bits de controle no cabeçalho TCP que indicam o estado ou o propósito de um pacote TCP.
 - **SYN:** Sinaliza o início de uma conexão.
 - **ACK:** Confirma o recebimento de pacotes.
 - **FIN:** Sinaliza o término de uma conexão.
 - **RST:** Reinicia a conexão.
 - **PSH:** Indica que os dados devem ser empurrados imediatamente ao aplicativo.
 - **URG:** Indica dados urgentes.

FUNCIONAMENTO

FILTRO 1



TCP.ANALYSIS.FLAGS

Esse filtro é utilizado para **identificar pacotes TCP que têm flags específicos** definidos, indicando eventos ou estados importantes na comunicação TCP, como retransmissões, pacotes fora de ordem, e outros problemas ou condições especiais.

FILTRO 1



TCP.ANALYSIS.FLAGS

```
62 2804:1b2:8183:902:b... TCP 98 [TCP Retransmission] 443 → 62570 [FIN, PSH, ACK] Seq=
:: 2804:1b2:8183:902:b... TCP 74 [TCP Retransmission] 443 → 62568 [FIN, ACK] Seq=4411
02:b... 2600:1901:1:7c5:: TCP 74 [TCP ZeroWindow] 62568 → 443 [ACK] Seq=4673 Ack=4412
:: 2804:1b2:8183:902:b... TCP 74 [TCP Retransmission] 443 → 62575 [FIN, ACK] Seq=1081
02:b... 2600:1901:1:7c5:: TCP 74 [TCP ZeroWindow] 62575 → 443 [ACK] Seq=12337 Ack=1081
0:6670 2804:1b2:8183:902:b... TCP 74 [TCP Dup ACK 3175#1] 443 → 62579 [ACK] Seq=1 Ack=747
```

EXEMPLO

FILTRO 2



DNS OR HTTP

Protocolo DNS:

- **Domain Name System (DNS)** é responsável pela resolução de nomes de domínio para endereços IP.
- **As consultas DNS transformam nomes de domínio amigáveis (como www.exemplo.com) em endereços IP necessários para o roteamento de rede.**

TEORIA

FILTRO 2



DNS OR HTTP

Protocolo HTTP:

- **HyperText Transfer Protocol (HTTP) é utilizado para comunicação entre navegadores web e servidores.**
- **Funciona no modelo de requisição e resposta, onde o cliente faz uma requisição e o servidor responde com os dados solicitados.**

TEORIA

FILTRO 2



DNS OR HTTP

DNS:

- **Consulta:** Um cliente envia uma consulta DNS para resolver um nome de domínio.
- **Resposta:** O servidor DNS responde com o endereço IP correspondente.

HTTP:

- **Requisição:** O cliente envia uma requisição HTTP (GET, POST, etc.).
- **Resposta:** O servidor responde com o conteúdo solicitado (página web, dados, etc.).

FUNCIONAMENTO

FILTRO 2



DNS OR HTTP

Esse filtro é usado para **exibir apenas** os pacotes DNS e HTTP na captura, facilitando a análise do tráfego de navegação web e resolução de nomes de domínio.

USO

FILTRO 2



DNS OR HTTP

192.168.15.1	DNS	77 Standard query 0xc829 AAAA www.instagram.com
192.168.15.1	DNS	77 Standard query 0x96f5 A www.instagram.com
192.168.15.1	DNS	77 Standard query 0x7611 HTTPS www.instagram.com
192.168.15.17	DNS	140 Standard query response 0xc829 AAAA www.instagram.com CNAME z-p42-ins
192.168.15.17	DNS	128 Standard query response 0x96f5 A www.instagram.com CNAME z-p42-instag
192.168.15.17	DNS	166 Standard query response 0x7611 HTTPS www.instagram.com CNAME z-p42-in
192.168.15.1	DNS	101 Standard query 0x60b4 AAAA video-akpcw-cdn-spotify-com.akamaized.net
192.168.15.1	DNS	101 Standard query 0xe549 A video-akpcw-cdn-spotify-com.akamaized.net

EXEMPLO

FILTRO 3



!(ARP OR ICMP OR DNS)

Protocolo ARP:

- **Address Resolution Protocol (ARP)** é usado para mapear endereços IP para endereços MAC (Media Access Control) em redes locais.

Protocolo ICMP:

- **Internet Control Message Protocol (ICMP)** é utilizado para enviar mensagens de erro e operações de diagnóstico (como ping) entre dispositivos de rede.

TEORIA

FILTRO 3



!(ARP OR ICMP OR DNS)

ARP:

- **Solicitação ARP:** Um dispositivo solicita o endereço MAC correspondente a um endereço IP.
- **Resposta ARP:** O dispositivo com o endereço IP correspondente responde com seu endereço MAC.

ICMP:

- **Echo Request/Reply:** Utilizado para verificar a conectividade (ping).
- **Mensagens de Erro:** Indicando problemas de roteamento ou entrega de pacotes.

FUNCIONAMENTO

FILTRO 3



!(ARP OR ICMP OR DNS)

Esse filtro é usado para **excluir** pacotes ARP, ICMP e DNS, permitindo a análise de outros tipos de tráfego na rede.

FILTRO 3



!(ARP OR ICMP OR DNS)

:902:b...	2600:1901:1:7c5::	TCP	74 62834 → 443 [ACK] Seq=2170 Ack=799 Win=131072
c5::	2804:1b2:8183:902:b...	QUIC	86 Protected Payload (KP0)
	191.219.21.14	DTLSv1...	103 Application Data
	191.219.21.14	STUN	142 Binding Request user: 4b+E:wUvA
	192.168.15.17	STUN	106 Binding Success Response XOR-MAPPED-ADDRESS: 1
	157.240.12.52	TLSv1.2	124 Application Data

EXEMPLO

FILTRO 4



TCP.ANALYSIS.RETRANSMISSION

- O TCP é um protocolo de transporte confiável que garante a entrega de dados na ordem correta. Para fazer isso, ele usa um mecanismo de confirmação (ACK) para garantir que os pacotes foram recebidos corretamente.
- Quando um pacote é enviado, o remetente aguarda um ACK do destinatário. Se o ACK não for recebido dentro de um tempo especificado (timeout), o remetente presume que o pacote foi perdido ou corrompido e retransmite o pacote.

FUNCIONAMENTO

FILTRO 4



TCP.ANALYSIS.RETRANSMISSION

- **Wireshark identifica essas retransmissões usando o filtro `tcp.analysis.retransmission`.**

FUNCIONAMENTO

FILTRO 4



TCP.ANALYSIS.RETRANSMISSION

- **Este filtro é uma ferramenta poderosa no Wireshark para identificar e analisar retransmissões de pacotes TCP. Compreender o funcionamento do protocolo TCP e a importância das retransmissões ajuda a diagnosticar problemas de rede, melhorar o desempenho e garantir a segurança da rede.**

FILTRO 4



TCP.ANALYSIS.RETRANSMISSION

```
TCP      66 [TCP Retransmission] 6307 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6304 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6303 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6305 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6306 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6307 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6304 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      66 [TCP Retransmission] 6303 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
TCP      412 [TCP Retransmission] 6311 → 443 [PSH, ACK] Seq=288 Ack=100 Win=132352 Len=358
TCP      118 [TCP Retransmission] 6304 → 443 [PSH, ACK] Seq=680 Ack=245 Win=131584 Len=64
TCP      118 [TCP Retransmission] 6303 → 443 [PSH, ACK] Seq=648 Ack=245 Win=132096 Len=64
TCP     1494 [TCP Retransmission] 443 → 6311 [PSH, ACK] Seq=3802 Ack=646 Win=4194048 Len=1440
TLSv1.3 1494 [TCP Fast Retransmission] , Server Hello
TLSv1.3 1494 [TCP Fast Retransmission] , Server Hello
```

EXEMPLO

FILTRO 5



FRAME CONTAINS "ATTACHMENT" OR FRAME CONTAINS "PDF"

O filtro frame contains não se limita a um protocolo específico. Ele procura por uma string de texto dentro do conteúdo de qualquer quadro (frame) capturado. Isso significa que pode ser aplicado a qualquer pacote de rede, independentemente do protocolo.

TEORIA

FILTRO 5



FRAME CONTAINS "ATTACHMENT" OR FRAME CONTAINS "PDF"

- **Wireshark captura todo o tráfego de rede que passa pela interface de rede selecionada.**
- **O filtro frame contains procura dentro do conteúdo de cada quadro capturado para verificar se ele contém a string especificada.**

FUNCIONAMENTO

FILTRO 5



FRAME CONTAINS "ATTACHMENT" OR FRAME CONTAINS "PDF"

Este filtro pode ser usado para monitorar anexos de email, transferências de arquivos PDF e para fins de segurança da rede. Ao aplicá-lo, você pode obter insights valiosos sobre o tráfego de rede e detectar atividades específicas relacionadas a anexos e arquivos PDF.

USO

FILTRO 5



FRAME CONTAINS "ATTACHMENT" OR FRAME CONTAINS "PDF"

```
TCP      1494 4428 → 443 [ACK] Seq=78264244 Ack=41282 Win=65536 Len=1440 [TCP segment of a reassembled PDU]
TCP      1494 4428 → 443 [ACK] Seq=85915376 Ack=41282 Win=65536 Len=1440 [TCP segment of a reassembled PDU]
TCP      1494 4428 → 443 [ACK] Seq=108578076 Ack=42576 Win=66048 Len=1440 [TCP segment of a reassembled PDU]
HTTP     643 HTTP/1.1 200 OK
TCP      1486 [TCP Retransmission] 80 → 4512 [ACK] Seq=590 Ack=871 Win=67840 Len=1432
HTTP     894 HTTP/1.1 206 Partial Content
TCP      894 [TCP Retransmission] 80 → 4512 [PSH, ACK] Seq=2346 Ack=1345 Win=68864 Len=840
TCP      1486 80 → 4512 [ACK] Seq=3186 Ack=1819 Win=69888 Len=1432 [TCP segment of a reassembled PDU]
TCP      1486 [TCP Retransmission] 80 → 4512 [ACK] Seq=5650 Ack=2293 Win=70912 Len=1432
```

EXEMPLO

FILTRO 6



IP.ADDR==127.0.0.1

Utilizado no Wireshark para identificar pacotes cujo endereço IP de origem ou destino é 127.0.0.1, também conhecido como o endereço de loopback ou **localhost. Este endereço é usado para testes e diagnósticos dentro do próprio dispositivo.**

FUNCIONAMENTO

FILTRO 6



IP.ADDR==127.0.0.1

Abrir e visualizar portas abertas com NMAP

- 1. Baixar e instalar o Nmap (Network Mapper);**
- 2. Executar o Nmap para escanear as portas do localhost;**
- 3. Abrir uma nova porta no localhost;**
- 4. Fechar essa porta criada;**
- 5. Verificar o Wireshark, com o filtro 6.**

TESTE

FILTRO 6



1. BAIXAR E INSTALAR O NMAP

ACESSAR <https://nmap.org/download.html>

BAIXAR A VERSÃO COMPATÍVEL COM SEU SISTEMA OPERACIONAL

INSTALAR DE ACORDO COM AS INSTRUÇÕES DO SITE OFICIAL

TESTE

FILTRO 6



2.EXECUTAR O NMAP

ABRIR O TERMINAL COMO ADMINISTADOR

EXECUTAR O SEGUINTE COMANDO PARA ESCANEAR AS PORTAS DO

LOCALHOST: `nmap 127.0.0.1`

TESTE

FILTRO 6



2.EXECUTAR O NMAP

```
Starting Nmap 7.95 ( https://nmap.org ) at 2024-07-17 00:22 Hora oficial do Brasil
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00015s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
445/tcp    open  microsoft-ds
1123/tcp   open  murray
5432/tcp   open  postgresql
6881/tcp   open  bittorrent-tracker
49152/tcp  open  unknown
```

TESTE

FILTRO 6



3.ABRIR UMA NOVA PORTA NO LOCALHOST;

ABRIR OUTRA ABA DO TERMINAL, COMO ADMINISTADOR

EXECUTAR O SEGUINTE COMANDO PARA CRIAR UMA NOVA PORTA NO

LOCALHOST: **ncat -l n°_da_porta**

TESTE

FILTRO 6



3.ABRIR UMA NOVA PORTA NO LOCALHOST;

```
C:\Windows\System32>ncat -l 1001
```

TESTE

FILTRO 6



4.FECHAR A PORTA CRIADA

NA ABA DO TERMINAL QUE USAMOS O NMAP, EXECUTAR O SEGUINTE
COMANDO PARA CRIAR UMA NOVA PORTA NO LOCALHOST:

ncat 127.0.0.1 n°_da_porta

TESTE

FILTRO 6



5. VERIFICAR O WIRESHARK COM O FILTRO 6

A PORTA 1001 FOI USADA EM UM PROTOCOLO TCP!!

```
56 7977 → 1001 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
56 1001 → 7977 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
```

TESTE

FILTRO 6



IP.ADDR==127.0.0.1

O filtro `ip.addr == 127.0.0.1` no Wireshark é útil para capturar e analisar pacotes de loopback, permitindo que você examine a comunicação interna no dispositivo.

FILTRO EXTRA



TLS.HANDSHAKE.EXTENSIONS_SERVER_NAME CONTAINS "AMAZON.COM"

- **TLS (Transport Layer Security): É um protocolo criptográfico que fornece comunicação segura pela rede. É amplamente utilizado para proteger a comunicação na internet, incluindo HTTPS.**

TEORIA

FILTRO EXTRA



TLS.HANDSHAKE.EXTENSIONS_SERVER_NAME CONTAINS "AMAZON.COM"

- **Handshake TLS: O handshake é o processo inicial de estabelecimento de uma conexão TLS. Durante este processo, o cliente e o servidor negociam parâmetros de segurança, trocam chaves e autenticam um ao outro.**

TEORIA

FILTRO EXTRA



TLS.HANDSHAKE.EXTENSIONS_SERVER_NAME CONTAINS "AMAZON.COM"

- **Extensão SNI (Server Name Indication):** A SNI é uma extensão do protocolo TLS que permite que o cliente informe ao servidor o nome do host ao qual ele está tentando se conectar no início do handshake TLS. Isso permite que o servidor utilize certificados diferentes para diferentes nomes de host.

TEORIA

FILTRO EXTRA



Este filtro funciona analisando os pacotes de handshake do protocolo TLS (Transport Layer Security) e verificando se a extensão SNI (Server Name Indication) contém a string "amazon.com". Durante o processo de handshake, o cliente inclui a extensão SNI na mensagem ClientHello para informar ao servidor o nome do host desejado. O Wireshark utiliza este filtro para exibir apenas os pacotes que contêm "amazon.com" na extensão SNI, permitindo a identificação de conexões TLS estabelecidas com o servidor específico.

FUNCIONAMENTO

FILTRO EXTRA



O filtro `tls.handshake.extensions_server_name contains "amazon.com"` no Wireshark é uma ferramenta poderosa para identificar e analisar pacotes de handshake TLS que utilizam a extensão SNI para se conectar a um servidor específico. Este filtro permite que você monitore e audite conexões seguras em uma rede, fornecendo insights valiosos sobre o tráfego HTTPS e a segurança da comunicação.

FILTRO EXTRA



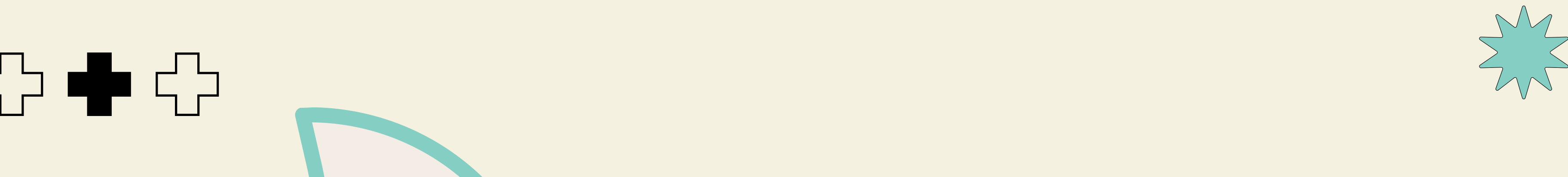
TLS.HANDSHAKE.EXTENSIONS_SERVER_NAME CONTAINS "AMAZON.COM"

QUIC	1282	Initial, DCID=491a7efe243e22be, PKN: 1, CRYPTO
TLSv1.2	646	Client Hello (SNI=m.media-amazon.com)
TLSv1	588	Client Hello (SNI=completion.amazon.com.br)

EXEMPLO

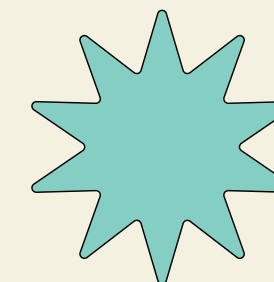
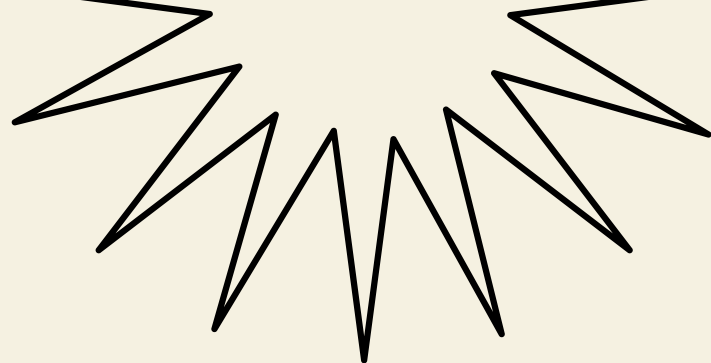


CONCLUSÃO

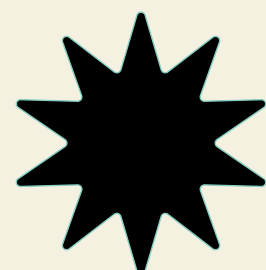
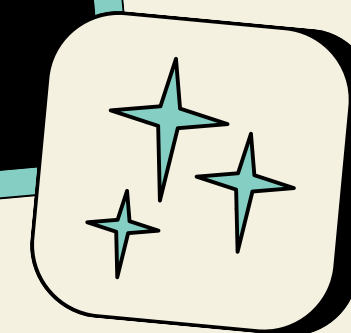
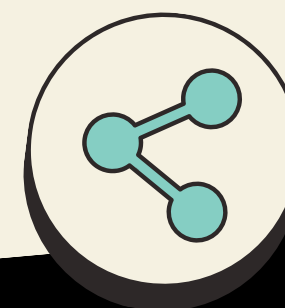
- **Eficiência na Análise:** Utilizar filtros no Wireshark é crucial para focar em pacotes específicos de interesse, facilitando a análise e resolução de problemas de rede.
 - **Compreensão Detalhada:** Os filtros permitem uma compreensão mais profunda do comportamento dos protocolos, ajudando a identificar e solucionar questões de desempenho e segurança.
 - **Economia de Tempo:** Filtros ajudam a navegar rapidamente através de grandes volumes de dados capturados, economizando tempo e esforços durante a análise.
- 



GIT HUB



OBRIGADO



ALUNOS: EMILAINÉ DO PRADO CORREIA
JOÃO VITOR DE SOUZA RIBEIRO
VINÍCIUS FERREIRA COUTO