

# SharkFest '16

## Advanced Wireshark Display Filters: How to Zoom in on the 10 Packets You Actually Need

Download files from [tinyurl.com/tcptraces](http://tinyurl.com/tcptraces)

Wednesday, June 15, 2016



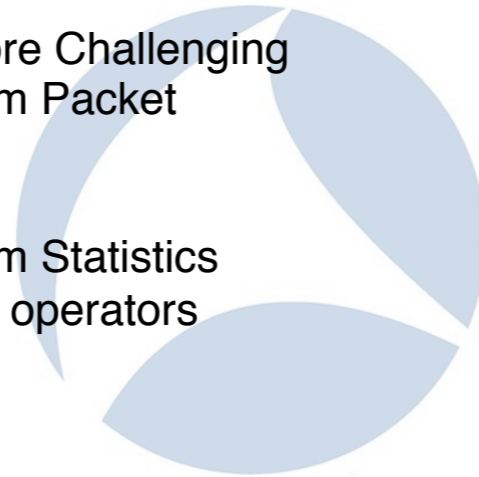
Betty DuBois, [betty@netdetect.co](mailto:betty@netdetect.co)

Chief Detective | Network Detectives, LLC

#SharkFest16 • Computer History Museum • June 13-16, 2016

# Agenda

- Saving
- Syntax
- From Easy to More Challenging
  - Right-Click from Packet
    - Simple
    - Compound
  - Right-Click from Statistics
  - Using different operators
    - Contains
    - Matches
    - In



## Where to Save Display Filters?

- Bookmark Display Filters
  - Saved in the "dfilters" file
    - Easy to share just that file of your profile
    - Have to "apply", extra click
- Filter Expression Buttons
  - More user-friendly
  - Saved in "preferences" file
    - More steps in sharing



#SharkFest16 • Computer History Museum • June 13-16, 2016

Start with WiresharkClassAfterNAT  
Create a dns filter

## Capture vs. Display Syntax

- Why are they different?
  - Capture filters are done by WinPcap/libpcap and follow the BPF syntax
  - Display filters are done by Wireshark and follow its syntax

Eth	IP	TCP	HTTP
DA SA ET	Pr SA DA	SP DP	http.host

#SharkFest16 • Computer History Museum • June 13-16, 2016

The green fields are available for capture filters. They are the ONLY ones. Limited power because you do not want to risk dropping packets.

The black field is a display filter field, any and every field is available for display filters. Much more powerful because you have all the time in the world.

Syntax difference:

Capture: host 192.168.10.10 and tcp port 80

Display: ip.addr==192.168.10.10 and tcp.port==80

## Display Filter Syntax

- Display filter syntax is “where what when”
- Where **is required** - Where to look
  - Frame
  - Protocol Name
  - Protocol Header
  - Field Name
- What **is optional** - What to do
  - Operator - &&, ||, ==, !=, <, <=, >, >=
  - Operator - and, or, eq, not, ne, lt, le, gt, ge
  - Operator - contains, matches, in
- When **is optional** - When the string/value is .....



#SharkFest16 • Computer History Museum • June 13-16, 2016

Add to the dns filter "dns or http"

## Case Matters

- Tcp and tcp are not the same
  - All protocol and operators should be input in lower case
  - **Exceptions:** Some VoIP call setup protocols use camel case
    - SIP
    - H.323 suite



## When in Doubt - Right-Click

- Easiest method to build filters
- Use Find to locate a packet of interest and r-click
  - Apply means; **Now**
  - Prepare means; **Stand by**, I'm going to edit
  - Compound filters can also be created with ...
- Use Statistics to find something of interest and r-click

#SharkFest16 • Computer History Museum • June 13-16, 2016

Use Find to find [talk.google.com](http://talk.google.com) DNS query response in packet 1649, then create the filter for the ip address with right-click  
Now find a retransmission and add it to the filter to determine if the percentage is too high. Nope, only 9 packets - that's good.

## Spaces – Do They Matter?

- **Yes**
  - Inside the double quotes only
  - `http.host == " bettyland.com"` is different from `http.host == "bettyland.com"`
- **No**
  - Outside the double quotes
    - `http.host=="bettyland.com"`
    - `http.host == "bettyland.com"`
    - `http.host == "bettyland.com"`
    - Are all equal





## Not vs. Not Equal

- `!(ip.dst == 31.13.77.58)`
  - Show me packets that do not have an IPv4 destination of 31.13.77.58
  - Packets that don't have an IP header **will flow through the filter** - STP for example. The IPv4 destination isn't equal to 31.13.77.58, because that field isn't even in the packet!

## Not vs. Not Equal

- `ip.dst != 31.13.77.58`
  - Show me packets that have IPv4 destinations, **unless** that destination(s) is 31.13.77.58 - get rid of those
  - Beware of bi-directional fields with this filter!
    - `ip.addr` is really `ip.src` or `ip.dst`
    - Therefore, `ip.addr != 31.13.77.58` is really  
`ip.src != 31.13.77.58 or ip.dst != 31.13.77.58`
    - Which shows all packets **for** 31.13.77.58 the exact opposite of what you expected/wanted
    - Yet, `ip and !ip.addr != 192.168.0.0/24` is a great way to show me packets where hosts in the subnet are talking to each other

## Hidden Fields

- What if I want to filter on text I see in the detail, but there isn't a field for me to right-click on?
  - Sometimes those are "hidden fields"
  - Type the protocol name then dot to see a list of available fields.
  - If you are unsure of which to use, display the hidden fields. Edit | Preferences | Protocols | check hidden fields

#SharkFest16 • Computer History Museum • June 13-16, 2016

Example in wiresharkclassafternat, eth.addr\_resolved contains Dell

## Using Contains

- Sometimes you don't know what field or even what protocol the data you are interested in might be
- Perfect time for the **contains** operator
- Example: **frame contains password**
  - How many are in WiresharkClassAfterNAT.pcap?
  - Counts are in the status bar at the bottom
- Contains filters are case-sensitive
  - It is really a hex filter, if there is no colon after the first byte, it becomes an ascii filter
- Works with frame, protocol, or fields that are character based, but not numeric based fields



## Using Matches

- Sometimes you know the character string you are looking for, but you don't know the protocol, or you need a wildcard in the string, or the position of the string matters
- Perfect time for the **matches** operator
  - Works with frame, protocol, or fields that are character based, but not numeric based
  - Matches uses Perl-Compatible Regular Expressions (PCRE)
  - Defaults to ascii characters
    - Hex can be denoted with \x - for example **frame**  
**matches "\x0d\x0aGET"** would find packets where the GET method appears in an unexpected position

## Using Matches

- Be aware:
  - “Some people, when confronted with a problem, think *“I know, I’ll use regular expressions.”* Now they have two problems.” Quote by Jamie Zawinski at <http://regex.info/blog/2006-09-15/247>

# RegEx Metacharacters

	\ ^ \$ * . ? +   ( ) []
\.	Means really <b>dot</b> not RegEx dot
“^string”	Means string must be at the <b>beginning</b> of field
“string\$”	Means string must be at <b>end</b> of field
“(?)string”	Means case <b>insensitive</b>
“str*ng”	The <b>asterisk</b> is a wildcard, zero or more characters can appear
“str.ng”	The <b>dot</b> is a wildcard, any single character can appear except new line
“stri?ng”	The <b>question mark</b> is a wildcard, the single letter <b>i</b> can appear or not appear
“stri+ng”	The plus is a wildcard, the letter <b>i</b> must appear one or many times
(string yarn)	Means either <b>string</b> or <b>yarn</b> must appear
()	Means it is a group of strings which may also include additional metacharacters
[]	Means it is a set of characters, and you may match on one of several

#SharkFest16 • Computer History Museum • June 13-16, 2016

## Byte-Offset Filters

- What if I can't right click on the field because my stuff is proprietary with no dissector??
- What if I only want the string if it occurs in a specific position??

```
data [113:10] == "lang=en-us"
```

data – starting offset

113 – number of bytes past the data field

10 – number of bytes to compare



#SharkFest16 • Computer History Museum • June 13-16, 2016



## Using Matches

- Try it on `WiresharkClassAfterNAT.pcap`
- How many DNS packets are there for `www.wireshark.org`?
- How many HTTP packets are there for `www.wireshark.org`?
- What fields does `www.wireshark.org` appear in?

## Slice Operator

- Sometimes you don't want the whole field
- Examples
  - `eth.src [0:3]==00:13:60` Start:Length
  - `eth.src [0-1]==00:13` Start:End (inclusive)
  - `eth.src [2]==13` Start:1Byte
  - `eth.src [:1]==00` Start0:Length
  - `eth.src [3:]==ff:f3:80` Start:EndOfField

**00:13:60:ff:f3:80**

**1 2 3 4 5 6**

#SharkFest16 • Computer History Museum • June 13-16, 2016



## Using In

- Sometimes you are ready to use matches, only to discover the field of interest is an integer
- Starting in 2.x, you can use a membership operator i.e., is the value of interest in this list?
- Works with both
  - `http.request.method` in {GET POST PUT}
  - `icmp.type` in {13 15 17}
- Think of it as an **or** for a field/header/frame of interest without having repeat the field/header/frame

## Converting Snort Rule to Display Filter

- # alert tcp \$HOME\_NET any -> \$EXTERNAL\_NET \$HTTP\_PORTS  
(msg:"MALWARE-CNC RDN Banker Strange Google Traffic";  
flow:to\_server,established; urilen:30; content:"User-Agent: Mozilla/4.0  
(compatible|3B| Win32|3B| WinHttp.WinHttpRequest.5)"; fast\_pattern:only;  
http\_header; content:"Host: www.google.com"; http\_header;  
metadata:impact\_flag red, policy security-ips drop, ruleset community,  
service http; reference:url,www.virustotal.com/en/file/  
1a23f27b046af92b7dd2c4a8f8349c9fd9582ad91b5a61556470c58b15af3b  
26/analysis/1369251144/; classtype:trojan-activity; sid:26836; rev:1;)
- http.user\_agent matches "Mozilla/4\.0 \ (compatible|3B| Win32|  
3B| WinHttp\.WinHttpRequest\.5\)" && http.host==[www.google.com](http://www.google.com)

## Where Else Can I Use Display Filter Syntax?

- Limit to Display Filter
    - Conversations
    - Endpoints
    - I/O Graph
    - DHCP/DNS/HTTP/IPv4/IPv6/SIP Statistics
  - Find
  - Expert
  - Coloring Rules
- 

## What about tshark?

- -Y -R are the display filter switches
  - -Y for a single pass filter
  - -2 -R for a double pass filter (where packets must be compared to each other)
- `tshark -r BreakingPoint.pcapng -2 -R "http.time gt .1" -w slow.pcapng`



#SharkFest16 • Computer History Museum • June 13-16, 2016

Any field that uses another packet to calculate it, requires a 2nd pass ie. the -R filter.



#SharkFest16 • Computer History Museum • June 13-16, 2016