Filter Maniacs

Goodies about display and capture filter

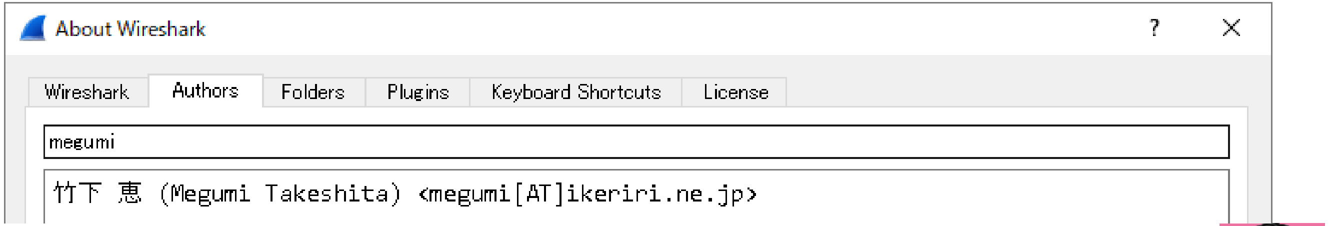
Megumi Takeshita

Packet Otaku, ikeriri network service

Sample trace and configuration
https://www.ikeriri.ne.jp/sharkfest

Megumi Takeshita, ikeriri network servic

- Founder, ikeriri network service co.,ltd
- Wrote 10+ books about Wireshark
- Reseller of Riverbed Technology (former CACE technologies) in Japan
- Attending all Sharkfest
- Translator of QT Wireshark into Japanese



megumi

竹下 恵 (Megumi Takeshita) <megumi[AT]ikeriri.ne.jp>

Filter Maniacs

TIPS and techniques about Wireshark display filters and WinPcap/libpcap capture filters. Wireshark has flexible and strong functions to filter packets, display filter by Wireshark, and capture filter by WinPcap/libpcap. We can capture the only packets you want and reduce trace file size using capture filters, and we can show the series of packets by display filter in a trace file.

This session Megumi shows practical TIPS and convenient techniques to use both filter using actual filter strings and trace files. You can utilize them in trace file and get the packet you need.

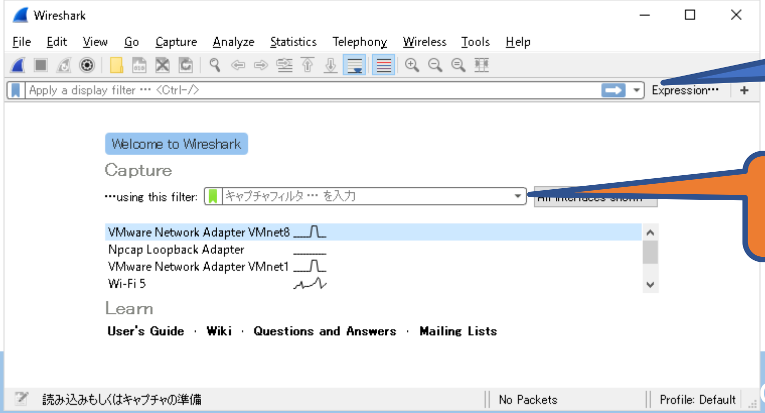
Agenda

- Capture and Display Filter Basics
- Capture Filter TIPS
- Display Filter TIPS
- Display filter Techniques
- Q & A

- Capture and Display Filter Basics

Capture filter and Display Filter

- Capture filter is used by WinPcap/libpcap/Npcap and other capture drivers to filter packet data
- Display filter is used by Wireshark/tshark/dumpcap to filter display information of packet list pane
- Each text box is able to use auto complete


Wireshark
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+
Welcome to Wireshark
Capture
…using this filter:
キャプチャフィルタ … を入力
VMware Network Adapter VMnet8
Npcap Loopback Adapter
VMware Network Adapter VMnet1
Wi-Fi 5
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
読み込みもしくはキャプチャの準備
No Packets
Profile: Default

Display Filter

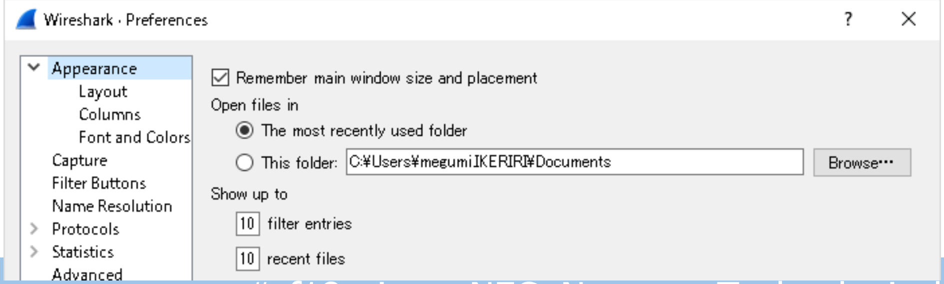
Capture Filter

Difference between Capture and Display filter

	Capture Filter	Display Filter
Set by	WinPcap/libpcap/Npcap and packet capture driver	Wireshark
Applies to	Each interface	Each trace file
Syntax	Tcpdump, pcap_compile(), and pf()	Wireshark protocol.field.subfield
Layer	Under layer 4 based on tcpdump, pcap_compile()	All layer based on the fields of the Wireshark's dissector
Pcap file size	Reduced	No change
Statistics	X Bad Ratio of packets is changed	○ Good Ratio of packets is the same

History of filter

- If you once set Capture / Display filter, the latest filter string is saved in filter list, and also saved in **recent_common** file in Personal configuration folder
- The number of history can be changed in settings



☑ Remember main window size and placement

Open files in

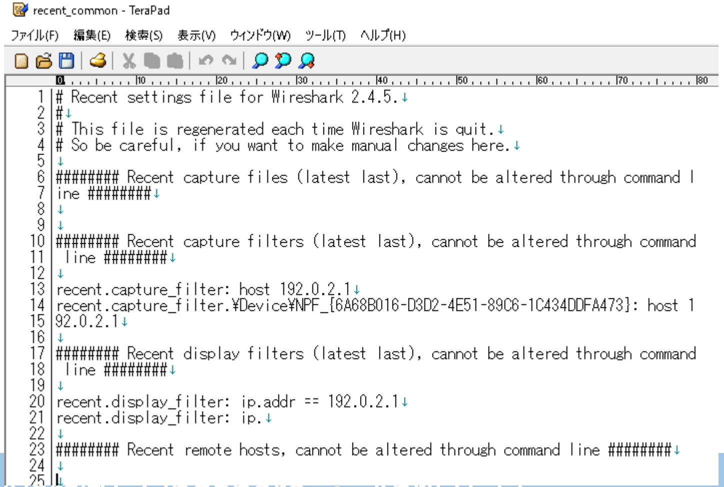
● The most recently used folder

○ This folder: C:¥Users¥megumiIKERIRI¥Documents

Show up to

10 filter entries

10 recent files


recent_common - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)

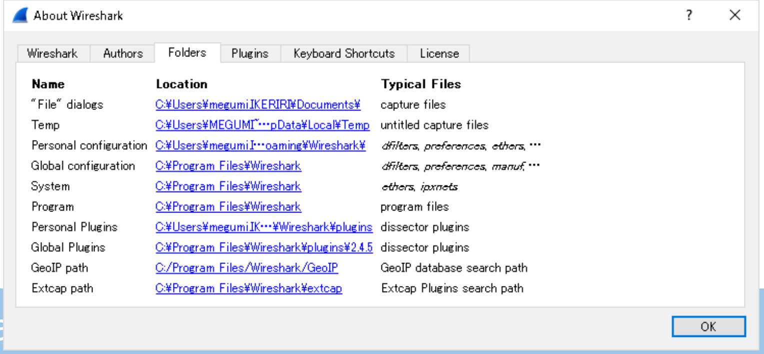
```
# Recent settings file for Wireshark 2.4.5.
#
# This file is regenerated each time Wireshark is quit.
# So be careful, if you want to make manual changes here.

######## Recent capture files (latest last), cannot be altered through command l
ine ########


######## Recent capture filters (latest last), cannot be altered through command
 line ########

recent.capture_filter: host 192.0.2.1
recent.capture_filter.¥Device¥NPF_{6A68B016-D3D2-4E51-89C6-1C434DDFA473}: host 1
92.0.2.1

######## Recent display filters (latest last), cannot be altered through command
 line ########

recent.display_filter: ip.addr == 192.0.2.1
recent.display_filter: ip.

######## Recent remote hosts, cannot be altered through command line ########

```

#1 check and test filter

- Open Wireshark, set Capture filter (host 8.8.8.8) and start capturing, and stop.
- Open another Wireshark, set Display filter (ip.addr==8.8.8.8) and start capturing and stop.
- Check the difference of each Syntax, file size and statistics
- Open Personal Configuration folder by Help>About>Folders and open recent_common file.
- Check the number of history by Edit>Preferences (show up to XX filter entries, XX recent files)

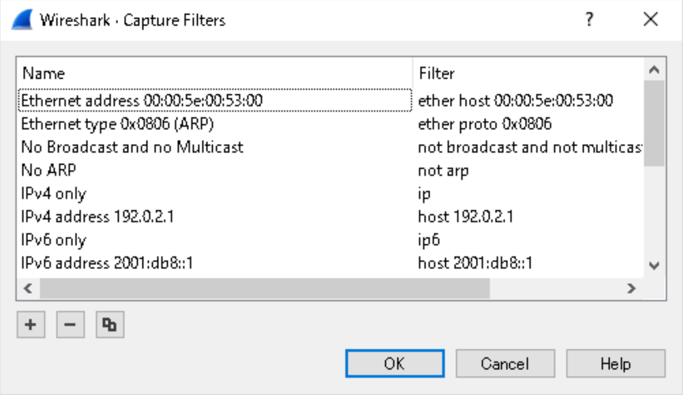
Configuration files of each filter

- Open Help>About Wireshark>Folders
- We can edit dfilters (Display filter template) and cfilters (Capture filter template) in Global configuration and Personal configuration (filter format using UTF8N and LF in Windows)
- You can also edit filters using filter dialog box Capture>Capture Filter…
Analyze>Display Filter



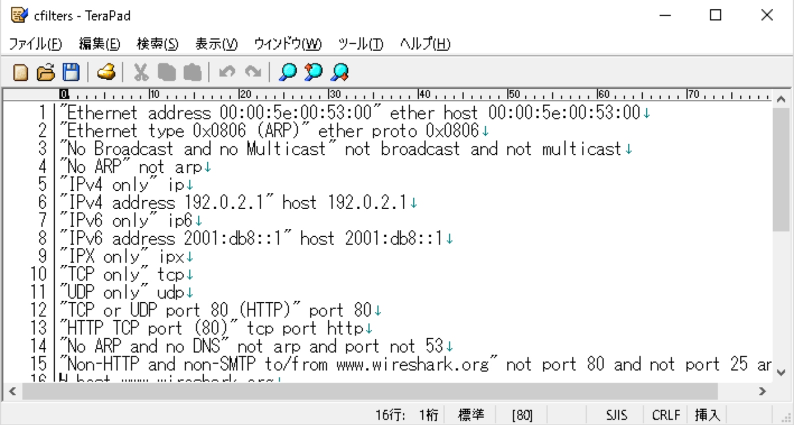
Name	Location	Typical Files
"File" dialogs	C:¥Users¥megumiIKERIRI¥Documents¥	capture files
Temp	C:¥Users¥MEGUMI˜…pData¥Local¥Temp	untitled capture files
Personal configuration	C:¥Users¥megumiI…oaming¥Wireshark¥	*dfilters, preferences, ethers, …*
Global configuration	C:¥Program Files¥Wireshark	*dfilters, preferences, manuf, …*
System	C:¥Program Files¥Wireshark	*ethers, ipxnets*
Program	C:¥Program Files¥Wireshark	program files
Personal Plugins	C:¥Users¥megumiIK…¥Wireshark¥plugins	dissector plugins
Global Plugins	C:¥Program Files¥Wireshark¥plugins¥2.4.5	dissector plugins
GeoIP path	C:/Program Files/Wireshark/GeoIP	GeoIP database search path
Extcap path	C:¥Program Files¥Wireshark¥extcap	Extcap Plugins search path

Check in Global configuration



Wireshark · Capture Filters

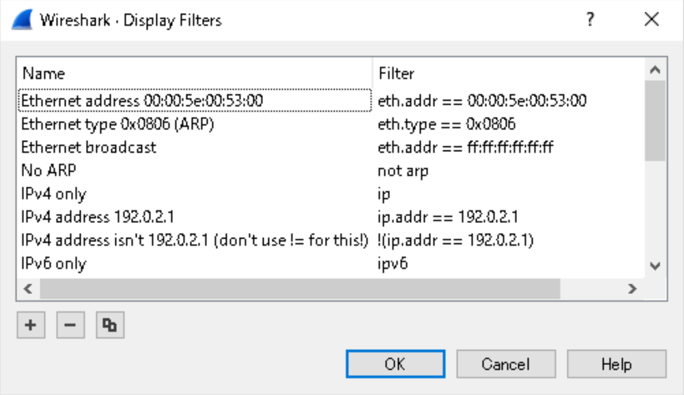
Name	Filter
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicas
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1



cfilters - TeraPad

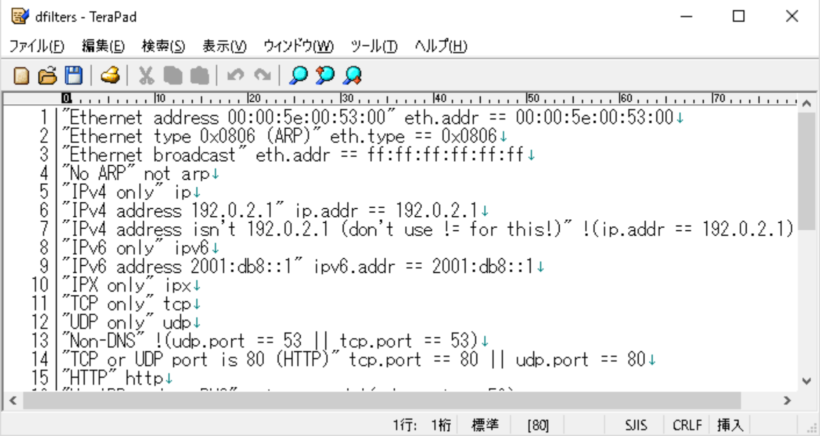
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)

```
"Ethernet address 00:00:5e:00:53:00" ether host 00:00:5e:00:53:00
"Ethernet type 0x0806 (ARP)" ether proto 0x0806
"No Broadcast and no Multicast" not broadcast and not multicast
"No ARP" not arp
"IPv4 only" ip
"IPv4 address 192.0.2.1" host 192.0.2.1
"IPv6 only" ip6
"IPv6 address 2001:db8::1" host 2001:db8::1
"IPX only" ipx
"TCP only" tcp
"UDP only" udp
"TCP or UDP port 80 (HTTP)" port 80
"HTTP TCP port (80)" tcp port http
"No ARP and no DNS" not arp and port not 53
"Non-HTTP and non-SMTP to/from www.wireshark.org" not port 80 and not port 25 ar
```



Wireshark · Display Filters

Name	Filter
Ethernet address 00:00:5e:00:53:00	eth.addr == 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	eth.type == 0x0806
Ethernet broadcast	eth.addr == ff:ff:ff:ff:ff:ff
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 address isn't 192.0.2.1 (don't use != for this!)	!(ip.addr == 192.0.2.1)
IPv6 only	ipv6



dfilters - TeraPad

ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)

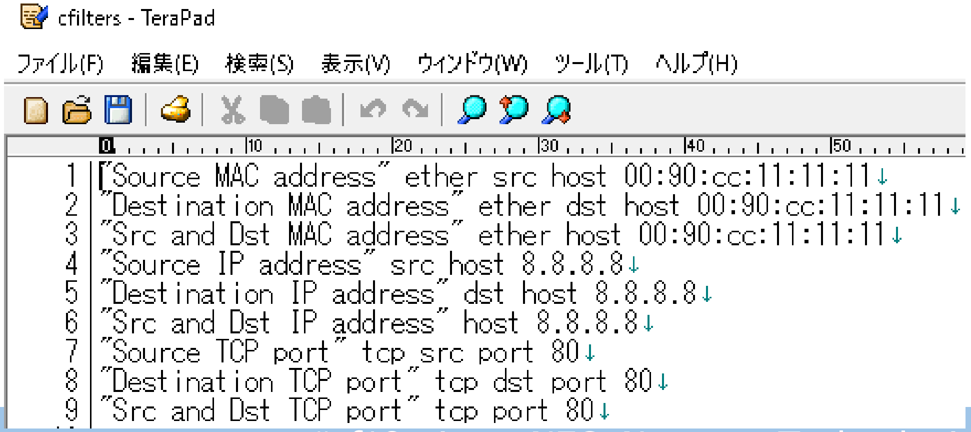
```
"Ethernet address 00:00:5e:00:53:00" eth.addr == 00:00:5e:00:53:00
"Ethernet type 0x0806 (ARP)" eth.type == 0x0806
"Ethernet broadcast" eth.addr == ff:ff:ff:ff:ff:ff
"No ARP" not arp
"IPv4 only" ip
"IPv4 address 192.0.2.1" ip.addr == 192.0.2.1
"IPv4 address isn't 192.0.2.1 (don't use != for this!)" !(ip.addr == 192.0.2.1)
"IPv6 only" ipv6
"IPv6 address 2001:db8::1" ipv6.addr == 2001:db8::1
"IPX only" ipx
"TCP only" tcp
"UDP only" udp
"Non-DNS" !(udp.port == 53 || tcp.port == 53)
"TCP or UDP port is 80 (HTTP)" tcp.port == 80 || udp.port == 80
"HTTP" http
```

Common example of Capture and Display filters

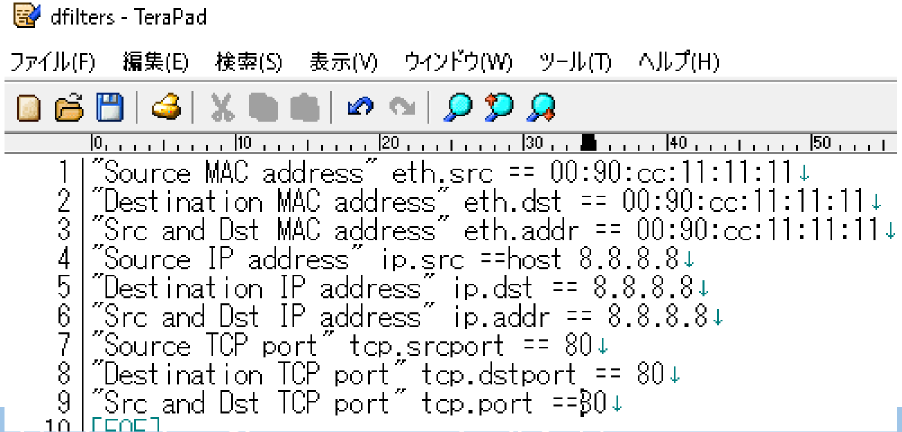
Address/port	Capture filter	Display filter
Source MAC address	ether src host	eth.src
Destination MAC address	ether dst host	eth.dst
Src and Dst MAC address	ether host	eth.addr
Source IPv4 address	src host	ip.src
Destination IPv4 address	dst host	ip.dst
Src and Dst IPv4 address	host	ip.addr
Source TCP port	tcp src port	tcp.srcport
Destination TCP port	tcp dst port	tcp.dstport
Src and Dst TCP port	tcp port	tcp.port

#2 Create your own filter template

- Create your own cfilters and dfilters and copy them into personal configuration from cfilter1 and dfilter1 and history
- Restart Wireshark and check each filter



```
"Source MAC address" ether src host 00:90:cc:11:11:11
"Destination MAC address" ether dst host 00:90:cc:11:11:11
"Src and Dst MAC address" ether host 00:90:cc:11:11:11
"Source IP address" src host 8.8.8.8
"Destination IP address" dst host 8.8.8.8
"Src and Dst IP address" host 8.8.8.8
"Source TCP port" tcp src port 80
"Destination TCP port" tcp dst port 80
"Src and Dst TCP port" tcp port 80
```

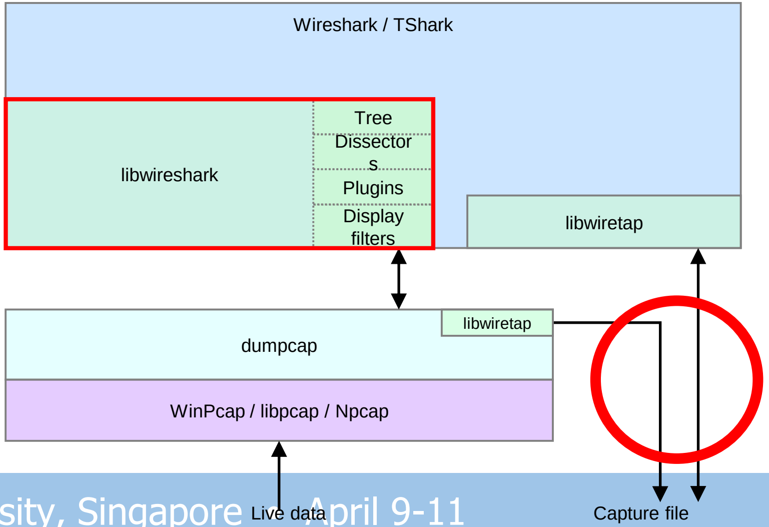


```
"Source MAC address" eth.src == 00:90:cc:11:11:11
"Destination MAC address" eth.dst == 00:90:cc:11:11:11
"Src and Dst MAC address" eth.addr == 00:90:cc:11:11:11
"Source IP address" ip.src ==host 8.8.8.8
"Destination IP address" ip.dst == 8.8.8.8
"Src and Dst IP address" ip.addr == 8.8.8.8
"Source TCP port" tcp.srcport == 80
"Destination TCP port" tcp.dstport == 80
"Src and Dst TCP port" tcp.port ==80
[EOF]
```

Capture filter TIPS

Capture Filter Syntax

- Capture filter is set on each interface, Select interface then put filter string
- Capture filter syntax is derived from tcpdump, pcap_compile() and pf() firewall
- Capture filter is concerned about under Transport layer header information (radio, ether, wlan, ppp, ip, ipv6, arp, rarp, tcp,udp, icmp)
- Different from Display filter



name and network and port

- You can use hostname in capture filter string
  [src | dst] host www.ikeriri.ne.jp
- network address in capture filter string
  [src | dst] net 172.16
  [src | dst] net 192.168 mask 255.255.255.0
- Broadcast and Multicast
  [ip] broadcast and multicast
- The port from 0 – 1023
  [tcp | udp] portrange 0-1023

Examples

"Source host name" src host www.ikeriri.ne.jp

"Destination host name" dst host www.ikeriri.ne.jp

"Src and Dst host name" host www.ikeriri.ne.jp

"Src and Dst Network 172.16.0.0" net 172.16

"Src and Dst Network 192.168.0.0/24" net 192.168 mask 255.255.255.0

"Src Network 172.16.0.0" src net 172.16

"Dst Network 192.168.0.0/24" dst net 192.168 mask 255.255.255.0

"Ethernet broadcast and multicast" broadcast and multicast

"IP broadcast and multicast" ip broadcast and ip multicast

"Well known TCP port" tcp portrange 0-1023

"Well known UDP port" udp portrange 0-1023

Byte value, Frame size and VLAN/WLAN

- Set frame size using less or grater
  less 100 means capture only under 100bytes frame
  greater 1000 means capture only over 1000bytes frame
- VLAN traffic
  vlan [vlanid] (check name resolution setting)
- WLAN management, control, and data frame
  type [mgt | ctl | data]
- WLAN traffic
  wlan [host | src | dst]
- WLAN subtype (Beacon, Probe Request, Probe Response, Authentication, Association Request, Association Response, ACK, RTS, CTS, Deauthentication and Disassociation WITH AirPcap and other wireless capture devices)
  subtype [beacon | probereq | proberesp | auth | assocreq | assocresp | ack | rts | cts | deauth | disassoc]

Examples

"Frame size is under 100" less 100

"Frame size is over 1000" greater 1000

"IEEE802.1Q vlan frame" vlan

"VLAN ID is 10" vlan 10

"IEEE802.11 Wireless lan" wlan

"IEEE802.11 MAC address 00:90:cc:11:11:11" wlan host 00:90:cc:11:11:11

"IEEE802.11 Souce address 00:90:cc:11:11:11" wlan src 00:90:cc:11:11:11

"IEEE802.11 Destination address 00:90:cc:11:11:11" wlan dst 00:90:cc:11:11:11

"IEEE802.11 Management frame" type mgt

"IEEE802.11 Control frame" type ctl

"IEEE802.11 Data frame" type mgt

Examples

"IEEE802.11 Beacon frame" subtype beacon

"IEEE802.11 Probe Request frame" subtype probereq

"IEEE802.11 Probe Response" subtype proberesp

"IEEE802.11 Authentication" subtype auth

"IEEE802.11 Association Request" subtype assocreq

"IEEE802.11 Association Response" subtype assocresp

"IEEE802.11 ACK frame" subtype ack

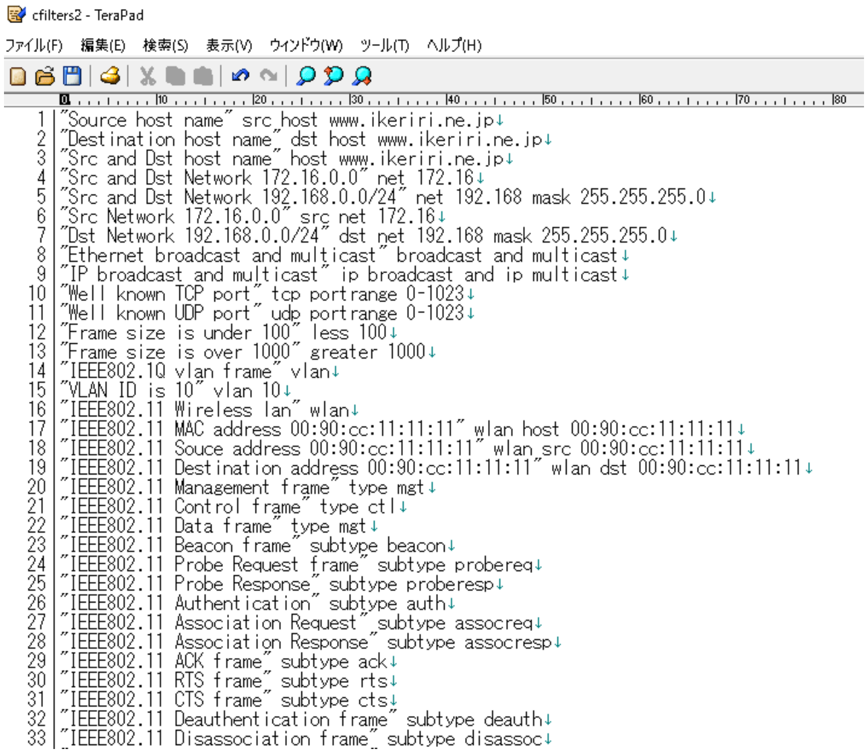
"IEEE802.11 RTS frame" subtype rts

"IEEE802.11 CTS frame" subtype cts

"IEEE802.11 Deauthentication frame" subtype deauth

"IEEE802.11 Disassociation frame" subtype disassoc

#3 collect only Wi-Fi connection


cfilters2 - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
0 10 20 30 40 50 60 70 80

```
"Source host name" src host www.ikeriri.ne.jp
"Destination host name" dst host www.ikeriri.ne.jp
"Src and Dst host name" host www.ikeriri.ne.jp
"Src and Dst Network 172.16.0.0" net 172.16
"Src and Dst Network 192.168.0.0/24" net 192.168 mask 255.255.255.0
"Src Network 172.16.0.0" src net 172.16
"Dst Network 192.168.0.0/24" dst net 192.168 mask 255.255.255.0
"Ethernet broadcast and multicast" broadcast and multicast
"IP broadcast and multicast" ip broadcast and ip multicast
"Well known TCP port" tcp portrange 0-1023
"Well known UDP port" udp portrange 0-1023
"Frame size is under 100" less 100
"Frame size is over 1000" greater 1000
"IEEE802.1Q vlan frame" vlan
"VLAN ID is 10" vlan 10
"IEEE802.11 Wireless lan" wlan
"IEEE802.11 MAC address 00:90:cc:11:11:11" wlan host 00:90:cc:11:11:11
"IEEE802.11 Souce address 00:90:cc:11:11:11" wlan src 00:90:cc:11:11:11
"IEEE802.11 Destination address 00:90:cc:11:11:11" wlan dst 00:90:cc:11:11:11
"IEEE802.11 Management frame" type mgt
"IEEE802.11 Control frame" type ctl
"IEEE802.11 Data frame" type mgt
"IEEE802.11 Beacon frame" subtype beacon
"IEEE802.11 Probe Request frame" subtype probereq
"IEEE802.11 Probe Response" subtype proberesp
"IEEE802.11 Authentication" subtype auth
"IEEE802.11 Association Request" subtype assocreq
"IEEE802.11 Association Response" subtype assocresp
"IEEE802.11 ACK frame" subtype ack
"IEEE802.11 RTS frame" subtype rts
"IEEE802.11 CTS frame" subtype cts
"IEEE802.11 Deauthentication frame" subtype deauth
"IEEE802.11 Disassociation frame" subtype disassoc
```

- Modify capture filter in Personal configuration using cfilter2 and set "collect only WiFi connection"
  Note: it needs IEEE802.11 wireless capture driver

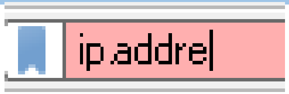
Display filter TIPS

Display Filter Syntax

- Filter syntax is Protocol.field.subfield style
- Display filter is set **on each capture file**, set filter string in text box of display Filter toolbar
- Display filter syntax is derived from each protocol dissector of Wireshark, look at each field of **packet detail pane and status bar**.
- Display filter is concerned about **all layer and generated fields** such as GeoIP, Expert info, time

Color of filter text box

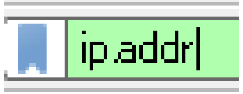
- Red means Error



ip.addre

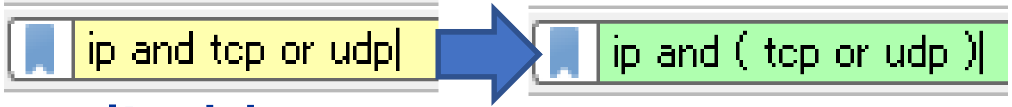
Filter string is not applied.

- Green means OK.



ip.addr

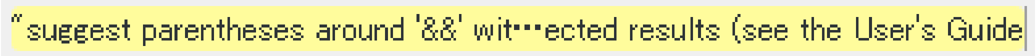
- the filter string can be applied
- Yellow means Warning



ip and tcp or udp

ip and (tcp or udp)

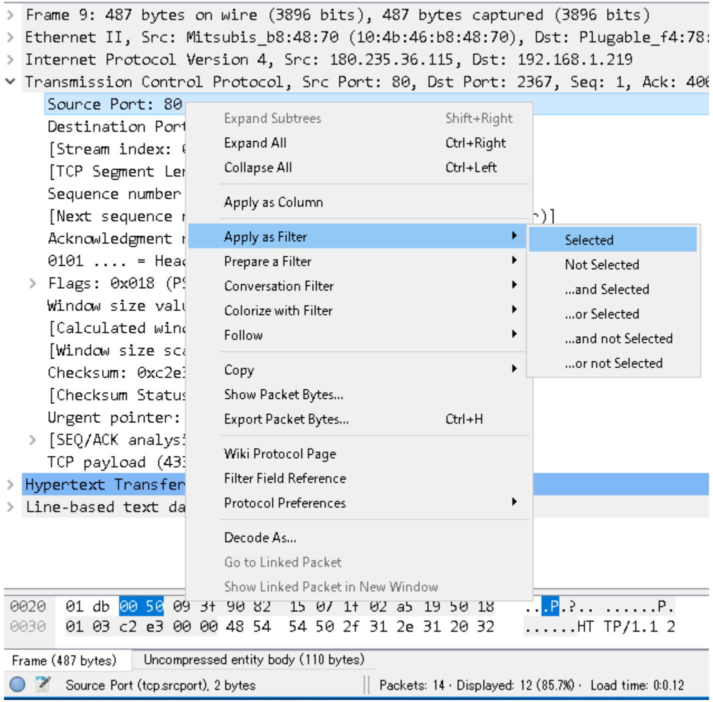
the filter string can be applied but
there are some ambiguous or contradiction
look status bar and USE BRACKET to fix



"suggest parentheses around '&&' wit···ected results (see the User's Guide

Cannot remember Filter String, select the field to right click

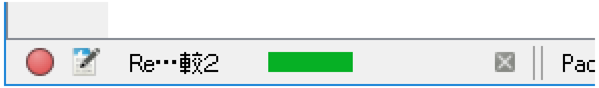
- If you cannot remember filter string, select each field of Packet detail pane.
- Wireshark display filter is derived from protocol dissectors, so look status bar.
- Select the field in Packet Detail Pane, Just right click to [Apply | Prepare] Filter > [Selected | Not Selected | …and Selected | …or Selected | … and not Selected | …or not Selected



```
> Frame 9: 487 bytes on wire (3896 bits), 487 bytes captured (3896 bits)
> Ethernet II, Src: Mitsubis_b8:48:70 (10:4b:46:b8:48:70), Dst: Plugable_f4:78:
> Internet Protocol Version 4, Src: 180.235.36.115, Dst: 192.168.1.219
v Transmission Control Protocol, Src Port: 80, Dst Port: 2367, Seq: 1, Ack: 40
    Source Port: 80
    Destination Port
    [Stream index:
    [TCP Segment Len
    Sequence number
    [Next sequence                                                    )]
    Acknowledgment
    0101 .... = Hea
  > Flags: 0x018 (P
    Window size val
    [Calculated win
    [Window size sca
    Checksum: 0xc2e
    [Checksum Statu
    Urgent pointer:
  > [SEQ/ACK analys
    TCP payload (43
> Hypertext Transfer
> Line-based text da

0020   01 db 00 50 09 3f 90 82  15 07 1f 02 a5 19 50 18   ...P.?.. ......P.
0030   01 03 c2 e3 00 00 48 54  54 50 2f 31 2e 31 20 32   ......HT TP/1.1 2

Frame (487 bytes)   Uncompressed entity body (110 bytes)
Source Port (tcp.srcport), 2 bytes        Packets: 14 · Displayed: 12 (85.7%) · Load time: 0:0.12
```

Which one is good for Display Filter ?
Apply or Prepare, try Prepare !

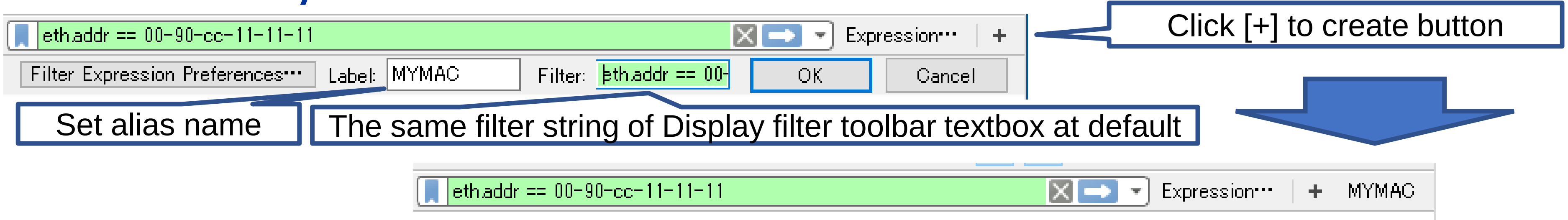
- If you create display filter in huge pcap/pcapng file, please try “Prepare Filter”, you can edit and check Display Filter string in Filter textbox.
- You can also add another filter string using “Prepare Filter”



- “Apply Filter” works immediately, so it may take several minutes to finish.

Create Display Filter Button

- It is good idea to create Display Filter Button in case of commonly use such as device MAC address.



eth.addr == 00-90-cc-11-11-11

Label: MYMAC

Filter: eth.addr == 00-

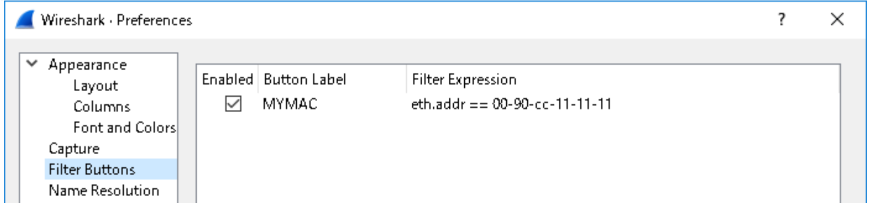
Click [+] to create button

Set alias name

The same filter string of Display filter toolbar textbox at default

eth.addr == 00-90-cc-11-11-11

- You can add/del/edit Filter Button Edit>Preference>Filter Buttons

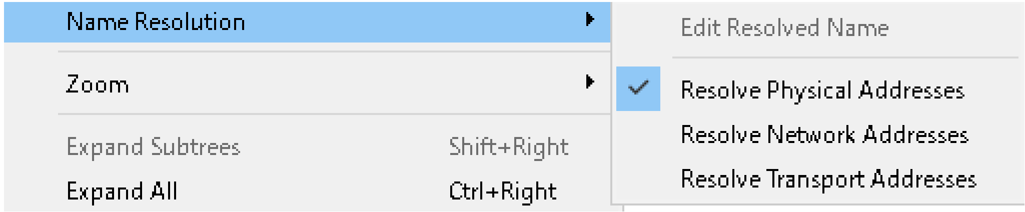


Wireshark · Preferences

Enabled	Button Label	Filter Expression
☑	MYMAC	eth.addr == 00-90-cc-11-11-11

Name Resolution

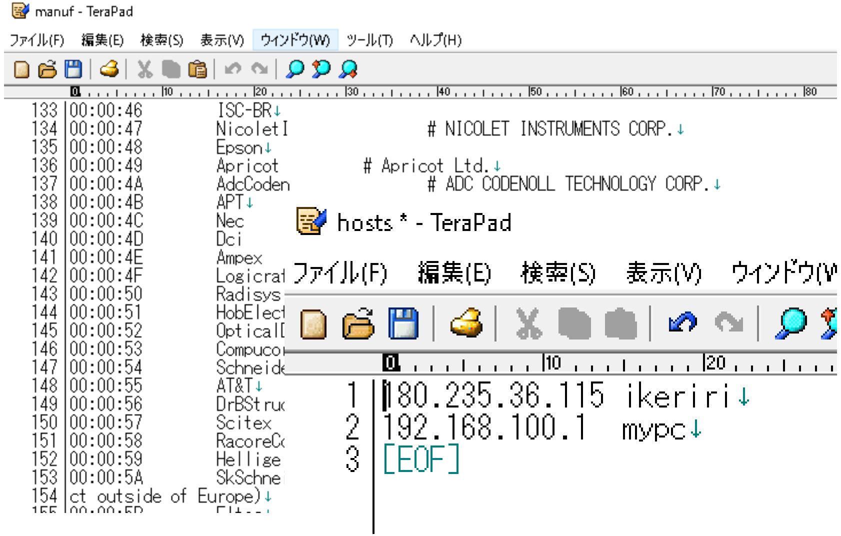
- Only Physical Address can be resolved at Default.
- You need to check “Resolve Network Address” in View>Name Resolution to use host name.



- Wireshark use manuf, hosts, services files in Global Configuration.
- You can also refer external DNS and DNS packet information to resolve name if you configure.

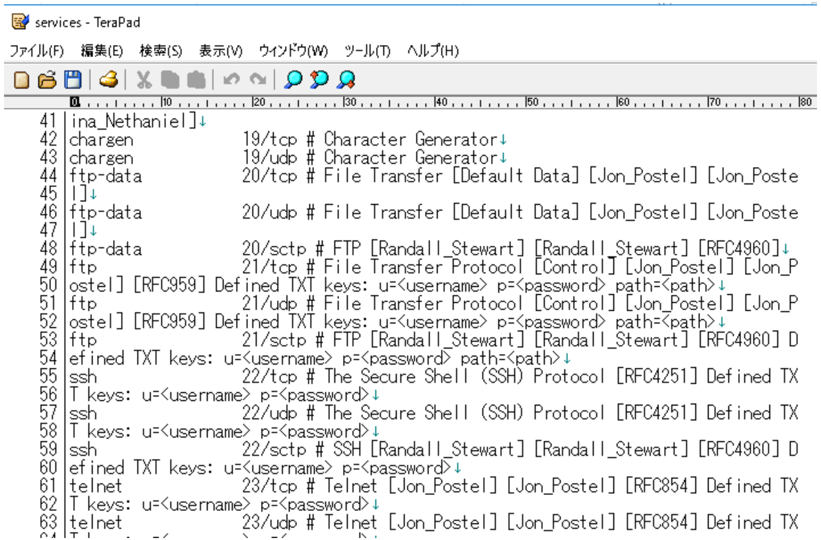
manuf, hosts, services

- You can edit manuf, hosts, services files to add your custom Name resolution aliases


manuf - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
hosts * - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W

```
00:00:46        ISC-BR↓
00:00:47        NicoletI                # NICOLET INSTRUMENTS CORP.↓
00:00:48        Epson↓
00:00:49        Apricot         # Apricot Ltd.↓
00:00:4A        AdcCoden                # ADC CODENOLL TECHNOLOGY CORP.↓
00:00:4B        APT↓
00:00:4C        Nec
00:00:4D        Dci
00:00:4E        Ampex
00:00:4F        Logicrai
00:00:50        Radisys
00:00:51        HobElect
00:00:52        OpticalI
00:00:53        Compucor
00:00:54        Schneide
00:00:55        AT&T↓
00:00:56        DrBStruc
00:00:57        Scitex
00:00:58        RacoreCo
00:00:59        Hellige
00:00:5A        SkSchne
ct outside of Europe)↓
```

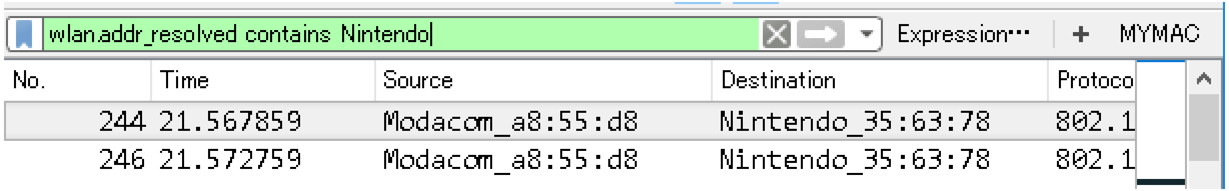
```
180.235.36.115 ikeriri↓
192.168.100.1   mypc↓
[EOF]
```


services - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)

```
ina_Nethaniel]↓
chargen             19/tcp # Character Generator↓
chargen             19/udp # Character Generator↓
ftp-data            20/tcp # File Transfer [Default Data] [Jon_Postel] [Jon_Poste
l]↓
ftp-data            20/udp # File Transfer [Default Data] [Jon_Postel] [Jon_Poste
l]↓
ftp-data            20/sctp # FTP [Randall_Stewart] [Randall_Stewart] [RFC4960]↓
ftp                 21/tcp # File Transfer Protocol [Control] [Jon_Postel] [Jon_P
ostel] [RFC959] Defined TXT keys: u=<username> p=<password> path=<path>↓
ftp                 21/udp # File Transfer Protocol [Control] [Jon_Postel] [Jon_P
ostel] [RFC959] Defined TXT keys: u=<username> p=<password> path=<path>↓
ftp                 21/sctp # FTP [Randall_Stewart] [Randall_Stewart] [RFC4960] D
efined TXT keys: u=<username> p=<password> path=<path>↓
ssh                 22/tcp # The Secure Shell (SSH) Protocol [RFC4251] Defined TX
T keys: u=<username> p=<password>↓
ssh                 22/udp # The Secure Shell (SSH) Protocol [RFC4251] Defined TX
T keys: u=<username> p=<password>↓
ssh                 22/sctp # SSH [Randall_Stewart] [Randall_Stewart] [RFC4960] D
efined TXT keys: u=<username> p=<password>↓
telnet              23/tcp # Telnet [Jon_Postel] [Jon_Postel] [RFC854] Defined TX
T keys: u=<username> p=<password>↓
telnet              23/udp # Telnet [Jon_Postel] [Jon_Postel] [RFC854] Defined TX
```

MAC Address Resolution

- You can use alias name of MAC address
  eth.addr_resolved (wlan.addr_resolved)
  eth.src_resolved (wlan.sa_resolved)
  eth.dst_resolved (wlan.da_resolved)
- If you want to look for Nintendo Switch type “wlan.addr_resolved contains Nintendo”

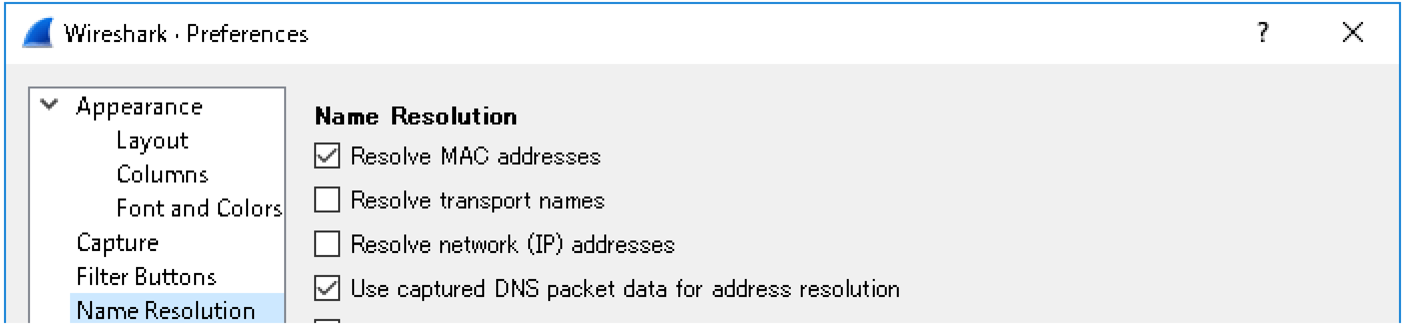


wlan.addr_resolved contains Nintendo

```
No.        Time       Source              Destination           Protoco
       244 21.567859  Modacom_a8:55:d8    Nintendo_35:63:78     802.1
       246 21.572759  Modacom_a8:55:d8    Nintendo_35:63:78     802.1
```

Host Name Resolution

- You can use host name in Display Filter
  ip.host ip.src_host ip.dst_host
  (View>Name Resolution> Resolve Network Address)
- You also need to refer Edit>Preference>Name Resolution



Wireshark · Preferences

Name Resolution

☑ Resolve MAC addresses

□ Resolve transport names

□ Resolve network (IP) addresses

☑ Use captured DNS packet data for address resolution

Examples (dfilter2)

“Sony’s MAC address” eth.addr_resolved contains Sony

“source MAC address of Sony” eth.src_resolved contains Sony

“destination MAC address of Sony” eth.dst_resolved contains Sony

“Nintendo’s wireless MAC address” wlan.addr_resolved contains Nintendo

“source wireless MAC address of Nintendo” wlan.sa_resolved contains Nintendo

“destination wireless MAC address of Nintendo” wlan.da_resolved contains Nintendo

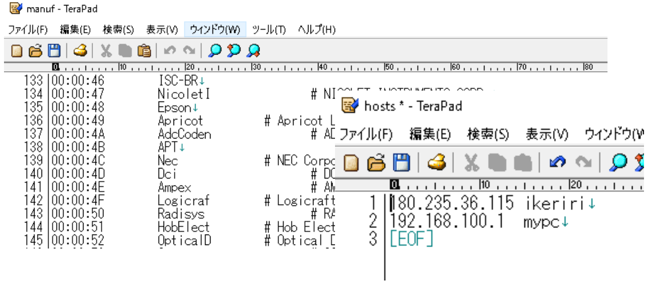
“Japan domain host” ip host contains jp

“source host of ikeriri” ip.src_host contains ikeriri

“destination host of ikeriri” ip.dst_host contains ikeriri

#4 edit your own alias

- Edit manuf and add alias of your own MAC address in Global configuration
- Edit hosts and add alias of your IP address too
- Check Resolve Network Address
- Restart Wireshark and start capturing

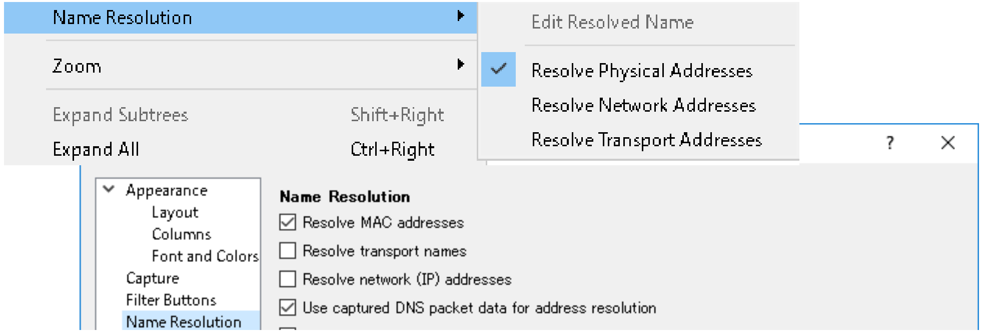


manuf - TeraPad

```
00:00:46        ISC-BR
00:00:47        NicoletI                # NI
00:00:48        Epson
00:00:49        Apricot         # Apricot L
00:00:4A        AdcCoden                # AD
00:00:4B        APT
00:00:4C        Nec             # NEC Corpo
00:00:4D        Dci                     # DC
00:00:4E        Ampex                   # AM
00:00:4F        Logicraf        # Logicraft
00:00:50        Radisys                 # RA
00:00:51        HobElect        # Hob Elect
00:00:52        OpticalD        # Optical D
```

hosts * - TeraPad

```
180.235.36.115 ikeriri
192.168.100.1  mypc
[EOF]
```



Name Resolution

☑ Resolve MAC addresses

☐ Resolve transport names

☐ Resolve network (IP) addresses

☑ Use captured DNS packet data for address resolution

Display filter Techniques

Multiple address and port

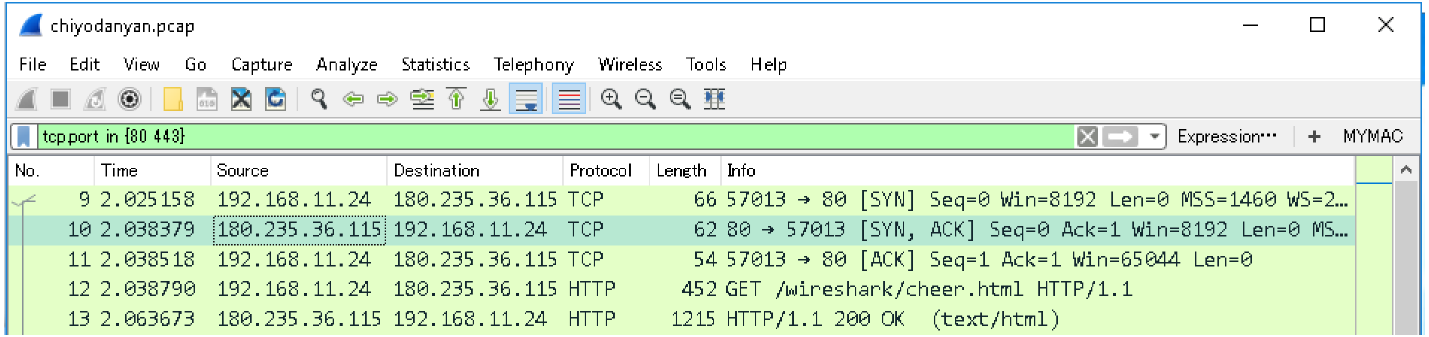
- If you want to grab the range of IP address and multiple port, there are some ways to filter packets.
- Filter IP Network
  ip.src>=192.168.100.0 and ip.src<=192.168.100.255
  ip.addr==192.168.100.0/24
- Filter HTTP and SSL port
  tcp.port == 80 or tcp.port == 443
  tcp.port in {80 443}

Examples (dfilter3)

“all address of network 192.168.100.0” ip.addr==192.168.100.0/24

“the range from 192.168.100.10 to 20” ip.src>=192.168.100.0 and ip.src<=192.168.100.255

“TCP HTTP and SSL port” tcp.port in {80 443}



tcp.port in {80 443}

```
No.  Time      Source          Destination     Protocol Length Info
   9 2.025158  192.168.11.24   180.235.36.115  TCP        66 57013 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2…
  10 2.038379  180.235.36.115  192.168.11.24   TCP        62 80 → 57013 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MS…
  11 2.038518  192.168.11.24   180.235.36.115  TCP        54 57013 → 80 [ACK] Seq=1 Ack=1 Win=65044 Len=0
  12 2.038790  192.168.11.24   180.235.36.115  HTTP      452 GET /wireshark/cheer.html HTTP/1.1
  13 2.063673  180.235.36.115  192.168.11.24   HTTP     1215 HTTP/1.1 200 OK  (text/html)
```

Slices [] in Display Filter

- You can match hex value using slices []
  typically used with eth, eth.src, eth.dst, ip, tcp, udp and other header (sometimes may not work as you expected)
- [start byte index : length]
  eth.src[0:3] first 3 bytes in ethernet source address
- [start index – end index]
  eth.dst[1-2] second, third bytes of ethernet destination
- [: size]
  ip[:2] first 2 bytes of IP header

Examples (dfilter4)

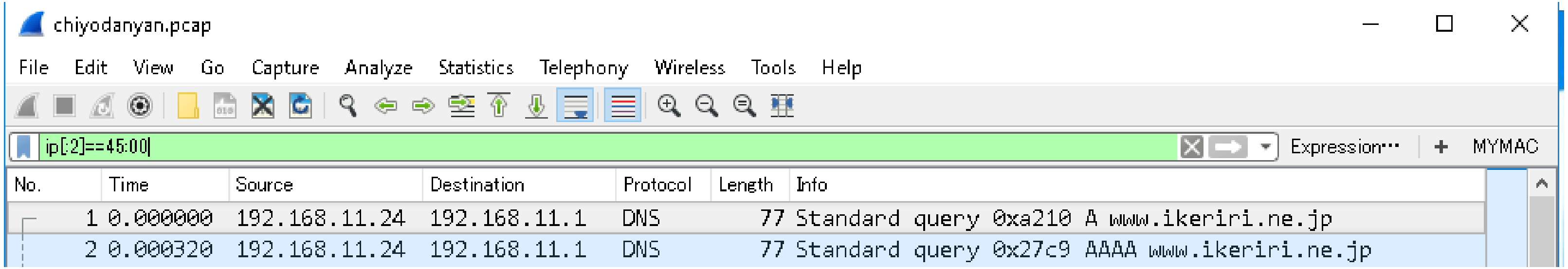
“OUI 00:D0:F1 (SEGA ENTERPRISES, LTD)” eth[0:3]==00:D0:F1

“second, third bytes of ethernet source is ff:ff” eth.src[1-2]==ff:ff

“second, third bytes of ethernet destination is ff:ff” eth.dst[1-2]==ff:ff

“IP version 4, length 20 TOS(DiffServ)=0 (first 2 bytes of IP header)” ip[:2]==45:00

“TCP destination port (from index 2 length 2 bytes) is 80(0x0050)” tcp[2:2]=00:50



ip[:2]==45:00

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.11.24	192.168.11.1	DNS	77	Standard query 0xa210 A www.ikeriri.ne.jp
2	0.000320	192.168.11.24	192.168.11.1	DNS	77	Standard query 0x27c9 AAAA www.ikeriri.ne.jp

Relation (contains / matches)

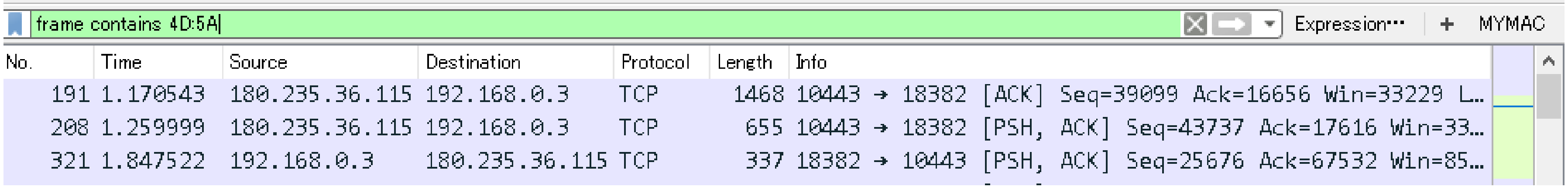
- Display filter string is commonly used with relation (eq(==), gt(>), lt(<), etc.)
- You can also use relation (and, or, not, xor)
- "**contains**" is convenient relation as wildcard of string value (ex. http.request.uri contains ikeriri
- "**matches**" is the relation of PCRE (Perl Compatible Regular Expressions)

Direct search of specified bytes value

- You can search specified bytes value in capture file using Display filters with “contains” relation
- Each file has a marker, the specified bytes value, for example JPEG file has a marker of Start (SOI: start of image) as “FF D8 FF”
- You can look for frame, tcp.segment, and more example frame contains ff:d8:ff

Examples (dfilter5)

“all frames that contains JPEG file SOI marker” frame contains FF-D8-FF
“all frames that contains PNG file signature (png.signature)” frame contains 89:50:4e:47:0d:0a:1a:0a
“find suspicious packets of Windows Executables (MZ marker)” frame contains 4D:5A
“find suspicious packets of Uboat RAT (remote access trojan) malware” frame contains 34:38:38



```
frame contains 4D:5A
```

```
No.  Time      Source          Destination     Protocol  Length Info
 191 1.170543  180.235.36.115  192.168.0.3     TCP         1468 10443 → 18382 [ACK] Seq=39099 Ack=16656 Win=33229 L…
 208 1.259999  180.235.36.115  192.168.0.3     TCP          655 10443 → 18382 [PSH, ACK] Seq=43737 Ack=17616 Win=33…
 321 1.847522  192.168.0.3     180.235.36.115  TCP          337 18382 → 10443 [PSH, ACK] Seq=25676 Ack=67532 Win=85…
```

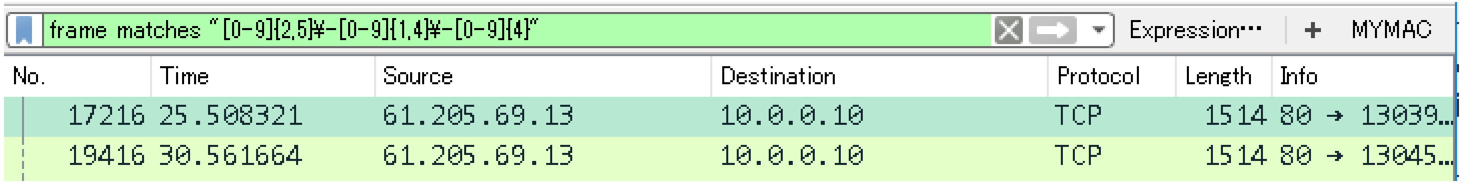
#5 sample use of PCRE

- Search Japanese local phone number in packets
  xx-xxxx-xxxx (first digits appears 2-5 times, second digits appears 1-4 times, and last digits appears 4 times)
  frame matches "[0-9]{2,5}¥-[0-9]{1,4}¥-[0-9]{4}”
  (Note: in a single byte environment escape character “¥” should be removed.) ¥ is backslash in Japanese keyboard map
- Search email address (any composite of alphabet, number and ._%+- ,@, any composite of subdomain and top level domain)
  frame matches “[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+[.][a-zA-Z]{2,4}”

Examples (dfilter6)

"Japanese local phone number in packets" frame matches "[0-9]{2,5}¥-[0-9]{1,4}¥-[0-9]{4}"

"Search email address" frame matches "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+[.][a-zA-Z]{2,4}"

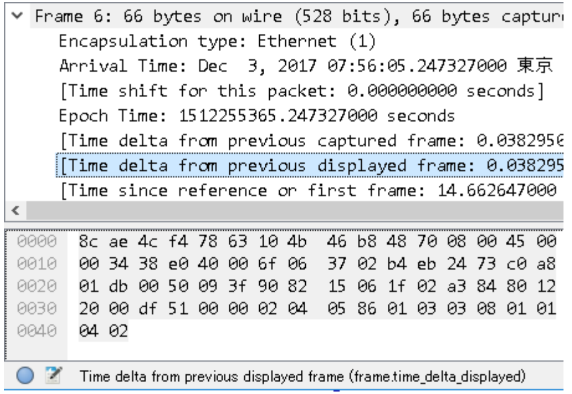


frame matches “[0-9]{2,5}¥-[0-9]{1,4}¥-[0-9]{4}”

No.	Time	Source	Destination	Protocol	Length	Info
17216	25.508321	61.205.69.13	10.0.0.10	TCP	1514	80 → 13039…
19416	30.561664	61.205.69.13	10.0.0.10	TCP	1514	80 → 13045…

Use Wireshark generated fields

- Display filter can refer generated fields as well as actual field of the dissectors
- You can use time and duration value of Wireshark generated field
  ex. icmp.resptime > 1
  ex. http.time > 1 or dns.time > 0.5
  ex. tcp.analysis.initial_rtt > 0.03
  ex. frame.time_delta_displayed > 1
- Please refer to dfilter7



```
⌄ Frame 6: 66 bytes on wire (528 bits), 66 bytes captur
    Encapsulation type: Ethernet (1)
    Arrival Time: Dec  3, 2017 07:56:05.247327000 東京
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1512255365.247327000 seconds
    [Time delta from previous captured frame: 0.0382956
    [Time delta from previous displayed frame: 0.038295
    [Time since reference or first frame: 14.662647000

0000  8c ae 4c f4 78 63 10 4b  46 b8 48 70 08 00 45 00
0010  00 34 38 e0 40 00 6f 06  37 02 b4 eb 24 73 c0 a8
0020  01 db 00 50 09 3f 90 82  15 06 1f 02 a3 84 80 12
0030  20 00 df 51 00 00 02 04  05 86 01 03 03 08 01 01
0040  04 02
```

Time delta from previous displayed frame (frame.time_delta_displayed)

Examples (dfilter7)

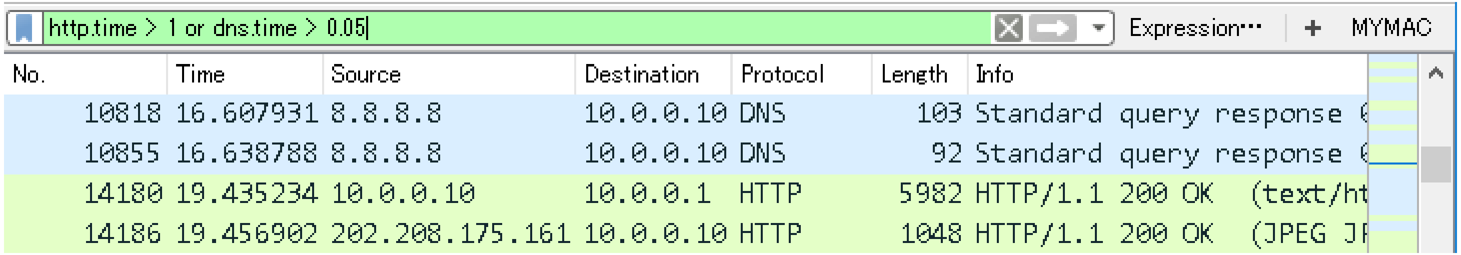
"Any frame that Ping responds in more than 1 second" icmp.resptime > 1

"Any frame that HTTP responds in more than 1 second" http.time > 1

"Any frame that DNS responds in less than 0.5 second" dns.time < 0.5

"Any frame that TCP initial Round Trip Time is more than 0.03 seconds" tcp.analysis.initial_rtt > 0.03

"Any frame that the time duration from previous displayed packet is more than 1 second" frame.time_delta_displayed > 1

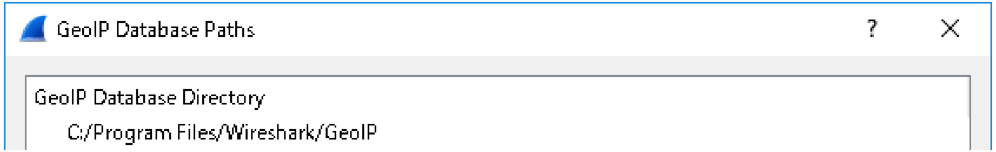


```
http.time > 1 or dns.time > 0.05
```

```
No.    Time      Source          Destination Protocol  Length Info
 10818 16.607931 8.8.8.8         10.0.0.10   DNS          103 Standard query response
 10855 16.638788 8.8.8.8         10.0.0.10   DNS           92 Standard query response
 14180 19.435234 10.0.0.10       10.0.0.1    HTTP        5982 HTTP/1.1 200 OK  (text/ht
 14186 19.456902 202.208.175.161 10.0.0.10   HTTP        1048 HTTP/1.1 200 OK  (JPEG JF
```

#6 use GeoIP

- Set GeoIP database directories in Name Resolution of Preferences > Name Resolution



GeoIP Database Paths

GeoIP Database Directory

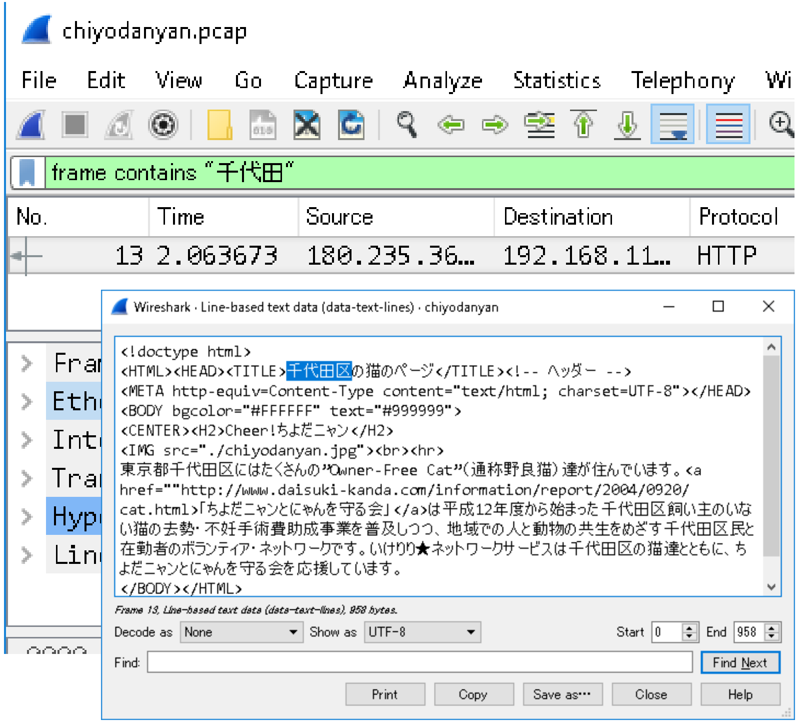
C:/Program Files/Wireshark/GeoIP

- Try to capture packets of Japanese website
- Try to filter packets using ip.geoip.country contains Japan or ipv6.geoip.country contains Japan

At last use multibytes

chiyodanyan.pcap

- open chiyodanyan.pcapng
- Try to use (if you have a multibytes character environment)
  frame contains "千代田“
- Wireshark can use UTF-8 characters including our CJK multibytes !



frame contains “千代田“

```
13 2.063673 180.235.36… 192.168.11… HTTP
```

```
<!doctype html>
<HTML><HEAD><TITLE>千代田区の猫のページ</TITLE><!-- ヘッダー -->
<META http-equiv=Content-Type content="text/html; charset=UTF-8"></HEAD>
<BODY bgcolor="#FFFFFF" text="#999999">
<CENTER><H2>Cheer!ちよだニャン</H2>
<IMG src="./chiyodanyan.jpg"><br><hr>
東京都千代田区にはたくさんの”Owner-Free Cat”(通称野良猫)達が住んでいます。<a
href=""http://www.daisuki-kanda.com/information/report/2004/0920/
cat.html>「ちよだニャンとにゃんを守る会」</a>は平成12年度から始まった千代田区飼い主のいな
い猫の去勢・不妊手術費助成事業を普及しつつ、地域での人と動物の共生をめざす千代田区民と
在勤者のボランティア・ネットワークです。いけりり★ネットワークサービスは千代田区の猫達とともに、ち
よだニャンとにゃんを守る会を応援しています。
</BODY></HTML>
```

Frame 13, Line-based text data (data-text-lines), 958 bytes.

Decode as: None ▾ Show as: UTF-8 ▾ Start: 0 End: 958

Find:

Q&A

Appendix Manpage and reference
Capture Filter https://www.tcpdump.org/manpages/pcap-filter.7.html
Display Filter https://www.wireshark.org/docs/man-pages/wireshark-filter.html
Display Filter references https://www.wireshark.org/docs/dfref/

USE Wireshark

Thank you very much !!

ワイヤーシャークを使おう！
どうもありがとうございました！ ❤