ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Computer Networks: Wireshark filtering

Luca Bedogni

Department of Computer Science and Engineering
University of Bologna

Wireshark filtering – a primer

- Wireshark empowers the user with powerful filtering expressions
  - Can filter based on the content of a packet
  - Can filter based on the IP of a packet
  - Can filter based on the protocol of a packet
  - Many others

- It supports comparison and boolean operators

- Also has a GUI

Filtering – comparison operators

English	Operator	Description	Example
eq	==	Equal	ip.src == 192.168.1.1
ne	!=	Not Equal	ip.src != 192.168.1.1
gt/lt/ge/le	> < >= >=	Greater/less/Greater or equal/Less or equal than	frame.len {>,<,>=,<=} 10
contains		Field contains a value	sip.To contains “a1762”
matches	~	Field matches a regexp	http.host matches “acme\.(org|com|net)”
bitwise_and	&	Compares bitfiled values	tcp.flags & 0x02

Filtering – combining operations

English	Operator	Description	Example
and	&&	Logical AND	ip.src == 192.168.1.1 and tcp.flags.fin
or	||	Logical OR	ip.src == 192.168.1.1 or tcp.flags.fin
xor	^^	Logical XOR	tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29
not	!	Logical Not	not llc
[…]		Slice Operator	See next slides
in		Membership operator	See next slides

Filtering – slice operator

- Used to select subsequences of a sequence
- Simply put brackets after a label
  - eth.src[0:3] == 00:00:83
  - eth.src[1-2] == 00:83
  - eth.src[:4] != 00:83:45:21
  - eth.src[4:] != 00:83
  - eth.src[4] != 00
  - eth.src[0:3,1-2,:4,4:,2] == 00:00:83:00:83:00:00:83:00:20:20:83

Filtering – Membership operator

- Used to test a field against a set of values
- Simply use ***in*** after a label and put the set inside {}
  - tcp.port in {80 443 8080}
    - This is equal to tcp.port == 80 or tcp.port == 443 or tcp.port == 8080

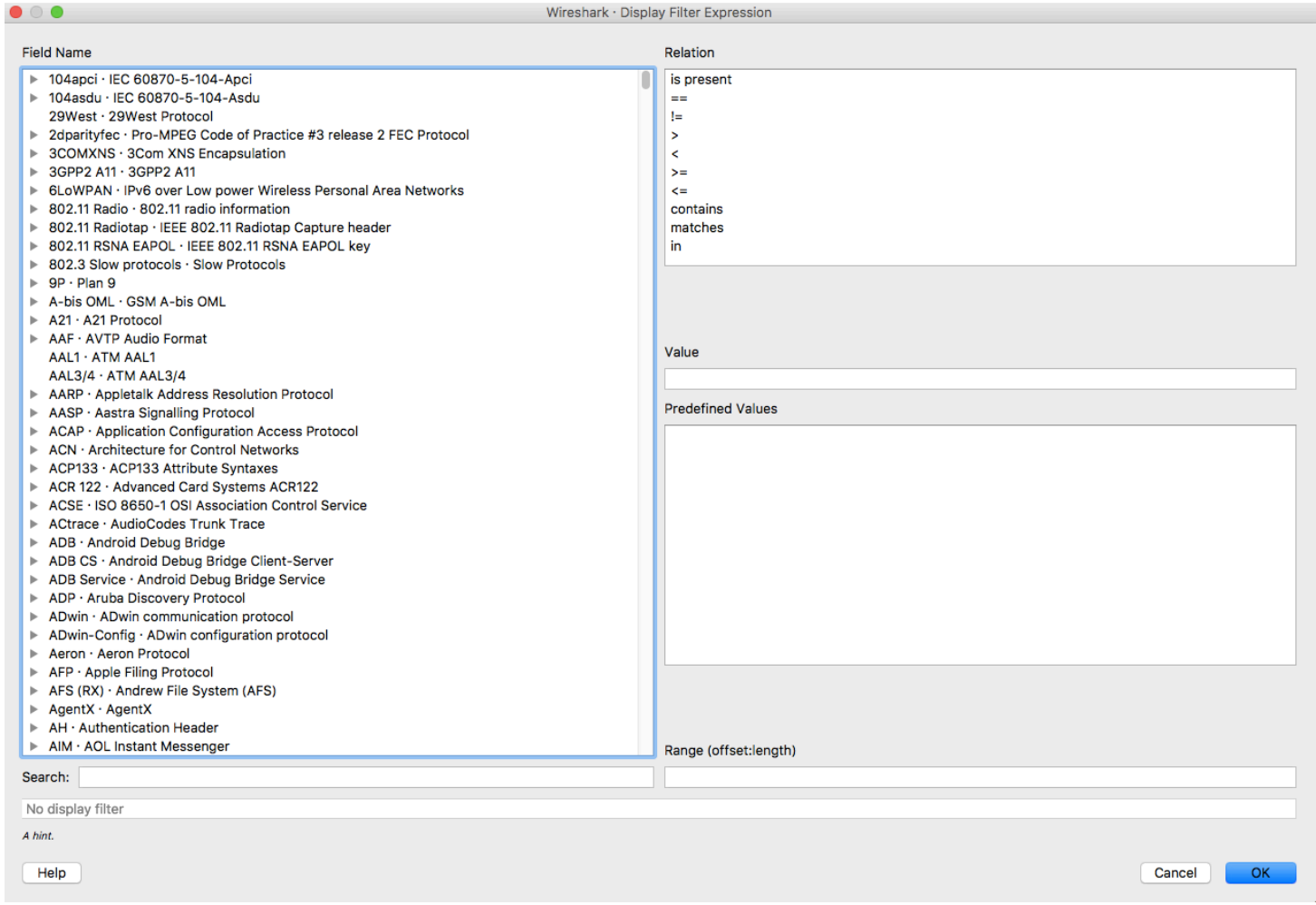
- You can also use ranges
  - tcp.port in {443 4430..4434}
    - This is equal to tcp.port == 443 or (tcp.port >= 4430 and tcp.port <= 4434)

- Something more complicated
  - http.request.method in {“HEAD” “GET”}
  - ip.addr in {10.0.0.5 .. 10.0.0.9 192.168.1.1 .. 192.168.1.9}
  - frame.time_delta in {10 .. 10.5}

- Wireshark also offers simple functions which can be helpful when dealing with packet content
  - upper/lower
  - len/count

Filtering – the filter expression dialog box

- Once you get used to wireshark, you will use it rarely
- At the beginning, it is an unvaluable tool to learn about wireshark's display strings
- It has a search function which makes easier to navigate through all the possible fields
- Also possible to define relations between fields and labels
- Finally, it is also possible to save filters for later use



Examples – Capture Filters

```
host 192.168.1.200
```

- Capture only packets coming from or going to host 192.168.1.200

```
ether host 00:00:5e:00:53:00
```

- Get all packets with source or destination MAC address equal to 00:00:5e:00:53:00

```
host google.it
```

- Get all the packets coming from or going to host www.google.it

```
not broadcast and not multicast
```

- Do not capture packets which are either broadcast or multicast

Examples – Display Filters

```
tcp.port == 80
```

- Get all the packets which have one port equal to 80

```
tcp.port == 80 or tcp.port == 443
```

```
tcp.port in {80 443}
```

- Get all the packets which have one port equal to 80 or 443

```
tcp.dstport == 80 and (tcp.srcport > 60000 and tcp.srcport < 64000)
```

- Get all the packets which are directed to port 80, and coming from a port number between 60000 and 64000

Reverse examples – Display Filters

- You are seeing an unusual http traffic from Ip 192.168.1.200, as it makes a lot of request to www.iamnotasafesite.danger. You are afraid that all its subnetwork (255.255.255.0) may be compromised. You want to identify all the hosts that are possibly compromised, what do you write?

  - Capture filters

```
src net 192.168.1.0/24
```

  - Display filters

```
http.host www.iamnotasafesite.danger
```

Reverse examples – Display Filters

- After analysing the previous attack, you discover that there are many sites which uses the domain .danger which may be harmful for your hosts. You also discover that there is a page, nowillstealallyourpersonalbelongings.html, which is the root cause of the problems. Identify all the hosts which visit such page.
  - Capture filters

```
src net 192.168.1.0/24
```

  - Display filters

```
http.host contains ".danger" and frame contains "nowillstealallyourpersonalbelongings.html"
```