<http://wfeng520.blog.hexun.com.tw/47990031_d.html>

下面是一張地址為192.168.1.2的計算機正在訪問“openmaniak.com”網站時的截圖。

|  |
| --- |
| [wireshark frontend](http://openmaniak.com/wireshark/wireshark_front.png) |

1. MENUS（菜單）

|  |
| --- |
| wireshark menus |

程序上方的8個菜單項用於對Wireshark進行配置：

|  |  |
| --- | --- |
| - "File"（文件） - "Edit" （編輯） - "View"（查看） - "Go" （轉到） - "Capture"（捕獲） - "Analyze"（分析） - "Statistics" （統計） - "Help" （幫助） | 打開或保存捕獲的信息。 查找或標記封包。進行全局設置。 設置Wireshark的視圖。 跳轉到捕獲的數據。 設置捕捉過濾器並開始捕捉。 設置分析選項。 查看Wireshark的統計信息。 查看本地或者在線支持。 |

3. DISPLAY FILTER（顯示過濾器）

|  |
| --- |
| [wireshark display filter](http://openmaniak.com/wireshark/wireshark_use_display_filter.png) |

顯示過濾器用於查找捕捉記錄中的內容。不要將捕捉過濾器和顯示過濾器的概念混淆。參考[Wireshark過濾器](http://openmaniak.com/cn/wireshark_filters.php)中的詳細內容。  
4. PACKET LIST PANE（封包列表）

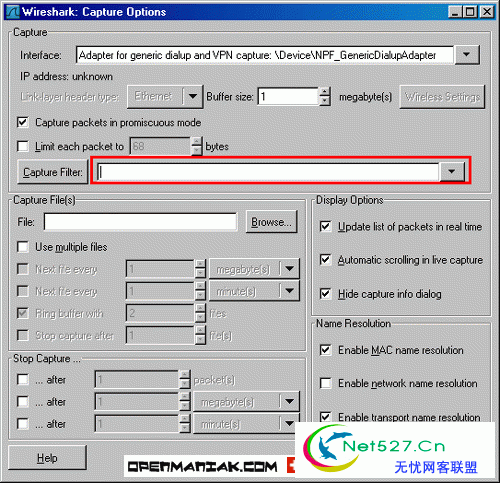
|  |
| --- |
| [wireshark packet filter pane](http://openmaniak.com/wireshark/wireshark_packet-list-pane.png) |

|  |
| --- |
| [wireshark packet filter pane](http://openmaniak.com/wireshark/wireshark_packet-list-pane2.png) |

封包列表中顯示所有已捕獲的封包。可看到發送或接收方的MAC/IP地址，TCP/UDP端口號，協議或者封包的內容。如果捕獲一個OSI layer 2的封包，在Source(來源)和Destination(目的地)列中看到的將是MAC地址，此時Port(端口)列將會為空。如果捕獲一個OSI layer 3或者更高層的封包，您在Source(來源)和Destination(目的地)列中看到的將是IP地址。Port(端口)列僅會在封包屬於第4或者更高層時才會顯示。

使用Wireshark時最常見的問題，是當您使用默認設置時，會得到大量冗余信息，以至於很難找到自己需要的部分。這就是為什麽過濾器會如此重要。它們可以幫助我們在龐雜的結果中迅速找到我們需要的信息。

|  |  |
| --- | --- |
| - - | 捕捉過濾器：用於決定將什麽樣的信息記錄在捕捉結果中。需要在開始捕捉前設置。 顯示過濾器：在捕捉結果中進行詳細查找。他們可以在得到捕捉結果後隨意修改。 |

那麽我應該使用哪一種過濾器呢？  
兩種過濾器的目的是不同的。  
捕捉過濾器是數據經過的第一層過濾器，它用於控制捕捉數據的數量，以避免產生過大的日誌文件。顯示過濾器是一種更為強大(復雜)的過濾器。它允許您在日誌文件中迅速準確地找到所需要的記錄。兩種過濾器使用的語法是完全不同的：  
http://www.net527.cn/uploads/allimg/100105/21124J532-11.gif1. 捕捉過濾器  
捕捉過濾器的語法與其它使用Lipcap(Linux)或者Winpcap(Windows)庫開發的軟件一樣，比如著名的[TCPdump](http://openmaniak.com/tcpdump.php)。捕捉過濾器必須在開始捕捉前設置完畢，這一點跟顯示過濾器是不同的。  
設置捕捉過濾器的步驟是：  
- 選擇 capture -> options。  
- 填寫"capture filter"欄或者點擊"capture filter"按鈕為過濾器起一個名字並保存，以便今後的捕捉中繼續用這個過濾器。  
- 點擊開始（Start）進行捕捉。  


|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 語法： |  | **Protocol** |  | **Direction** |  | **Host(s)** |  | **Value** |  | **Logical Operations** |  | **Other expression\_r** |
| 例子： |  | tcp |  | dst |  | 10.1.1.1 |  | 80 |  | and |  | tcp dst 10.2.2.2 3128 |

http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Protocol（協議）**:  
可能的值: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp.  
如果沒有特別指明是什麽協議，則默認使用所有支持的協議。  
http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Direction（方向）**:  
可能的值: src, dst, src and dst, src or dst  
如果沒有特別指明來源或目的地，則默認使用 "src or dst" 作為關鍵字。  
例如，"host 10.2.2.2"與"src or dst host 10.2.2.2"是一樣的。  
http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Host(s)**:  
可能的值： net, port, host, portrange.  
如果沒有指定此值，則默認使用"host"關鍵字。  
例如，"src 10.1.1.1"與"src host 10.1.1.1"相同。  
http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Logical Operations（邏輯運算）**:  
可能的值：not, and, or.  
否("not")具有最高的優先級。或("or")和與("and")具有相同的優先級，運算時從左至右進行。例如，  
"not tcp port 3128 and tcp port 23"與"(not tcp port 3128) and tcp port 23"相同。  
"not tcp port 3128 and tcp port 23"與"not (tcp port 3128 and tcp port 23)"不同。

例子：

|  |
| --- |
| **tcp dst port 3128** |

顯示目的TCP端口為3128的封包。

|  |
| --- |
| **ip src host 10.1.1.1** |

顯示來源IP地址為10.1.1.1的封包。

|  |
| --- |
| **host 10.1.2.3** |

顯示目的或來源IP地址為10.1.2.3的封包。

|  |
| --- |
| **src portrange 2000-2500** |

顯示來源為UDP或TCP，並且端口號在2000至2500範圍內的封包。

|  |
| --- |
| **not imcp** |

顯示除了icmp以外的所有封包。（icmp通常被ping工具使用）

|  |
| --- |
| **src host 10.7.2.12 and not dst net 10.200.0.0/16** |

顯示來源IP地址為10.7.2.12，但目的地不是10.200.0.0/16的封包。

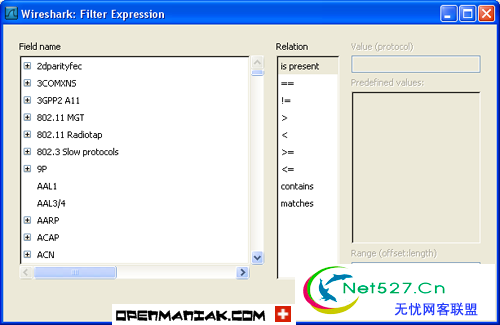
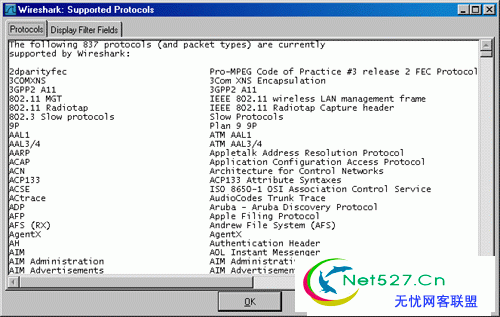
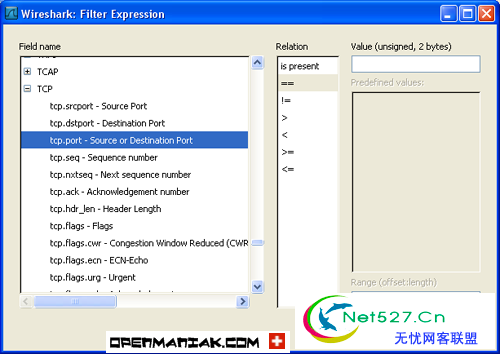
|  |
| --- |
| **(src host 10.4.1.12 or src net 10.6.0.0/16) and tcp dst portrange 200-10000 and dst net 10.0.0.0/8** |

顯示來源IP為10.4.1.12或者來源網絡為10.6.0.0/16，目的地TCP端口號在200至10000之間，並且目的位於網絡10.0.0.0/8內的所有封包。

註意事項：  
當使用關鍵字作為值時，需使用反斜杠“\”。  
"ether proto \ip" (與關鍵字"ip"相同).  
這樣寫將會以IP協議作為目標。  
  
"ip proto \icmp" (與關鍵字"icmp"相同).  
這樣寫將會以ping工具常用的icmp作為目標。  
  
可以在"ip"或"ether"後面使用"multicast"及"broadcast"關鍵字。  
當您想排除廣播請求時，"no broadcast"就會非常有用。

查看 [TCPdump的主頁](http://www.ethereal.com/docs/man-pages/tcpdump.8.html#_blank)以獲得更詳細的捕捉過濾器語法說明。  
在[Wiki Wireshark website](http://wiki.wireshark.org/CaptureFilters#_blank)上可以找到更多捕捉過濾器的例子。  
  
http://www.net527.cn/uploads/allimg/100105/21124J532-11.gif2. 顯示過濾器：  
  
通常經過捕捉過濾器過濾後的數據還是很復雜。此時您可以使用顯示過濾器進行更加細致的查找。  
它的功能比捕捉過濾器更為強大，而且在您想修改過濾器條件時，並不需要重新捕捉一次。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 語法： |  | **Protocol** | **.** | **String 1** | **.** | **String 2** |  | **Comparison operator** |  | **Value** |  | **Logical Operations** |  | **Other expression\_r** |
| 例子： |  | ftp |  | passive |  | ip |  | == |  | 10.2.3.4 |  | xor |  | icmp.type |

http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Protocol（協議）**:  
  
您可以使用大量位於OSI模型第2至7層的協議。點擊"Expression..."按鈕後，您可以看到它們。  
比如：IP，TCP，DNS，SSH  
  
  
  
[](http://openmaniak.com/wireshark/wireshark_display_filter_1.png)  
  
您同樣可以在如下所示位置找到所支持的協議：  
  
  
  
[](http://openmaniak.com/wireshark/wireshark_supported_protocols_window.gif)  
  
Wireshark的網站提供了對各種 [協議以及它們 子類的說明](http://www.wireshark.org/docs/dfref/#_blank)。  
  
http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**String1, String2** (可選項):  
  
協議的子類。  
點擊相關父類旁的"+"號，然後選擇其子類。  
  
[](http://openmaniak.com/wireshark/wireshark_display_filter_2.png)  
  
http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Comparison operators （比較運算符）**:  
  
可以使用6種比較運算符：

|  |  |  |
| --- | --- | --- |
| 英文寫法： | C語言寫法： | 含義： |
| eq | == | 等於 |
| ne | != | 不等於 |
| gt | > | 大於 |
| lt | < | 小於 |
| ge | >= | 大於等於 |
| le | <= | 小於等於 |

http://www.net527.cn/uploads/allimg/100105/21124MF7-14.gif**Logical expression\_rs（邏輯運算符）**:

|  |  |  |
| --- | --- | --- |
| 英文寫法： | C語言寫法： | 含義： |
| and | && | 邏輯與 |
| or | || | 邏輯或 |
| xor | ^^ | 邏輯異或 |
| not | ! | 邏輯非 |

被程序員們熟知的邏輯異或是一種排除性的或。當其被用在過濾器的兩個條件之間時，只有當且僅當其中的一個條件滿足時，這樣的結果才會被顯示在屏幕上。  
讓我們舉個例子：  
"tcp.dstport 80 xor tcp.dstport 1025"  
只有當目的TCP端口為80或者來源於端口1025（但又不能同時滿足這兩點）時，這樣的封包才會被顯示。

例子：

|  |  |
| --- | --- |
| **snmp || dns || icmp** | 顯示SNMP或DNS或ICMP封包。 |

|  |
| --- |
| **ip.addr == 10.1.1.1** |

顯示來源或目的IP地址為10.1.1.1的封包。

|  |
| --- |
| **ip.src != 10.1.2.3 or ip.dst != 10.4.5.6** |

顯示來源不為10.1.2.3或者目的不為10.4.5.6的封包。  
換句話說，顯示的封包將會為：  
來源IP：除了10.1.2.3以外任意；目的IP：任意  
以及  
來源IP：任意；目的IP：除了10.4.5.6以外任意

|  |
| --- |
| **ip.src != 10.1.2.3 and ip.dst != 10.4.5.6** |

顯示來源不為10.1.2.3並且目的IP不為10.4.5.6的封包。  
換句話說，顯示的封包將會為：  
來源IP：除了10.1.2.3以外任意；同時須滿足，目的IP：除了10.4.5.6以外任意

|  |  |
| --- | --- |
| **tcp.port == 25** | 顯示來源或目的TCP端口號為25的封包。 |

|  |  |
| --- | --- |
| **tcp.dstport == 25** | 顯示目的TCP端口號為25的封包。 |

|  |  |
| --- | --- |
| **tcp.flags** | 顯示包含TCP標誌的封包。 |

|  |  |
| --- | --- |
| **tcp.flags.syn == 0x02** | 顯示包含TCP SYN標誌的封包。 |

如果過濾器的語法是正確的，表達式的背景呈綠色。如果呈紅色，說明表達式有誤。

|  |  |
| --- | --- |
| wireshark display filter example | 表達式正確 |
| wireshark display filter example | 表達式錯誤 |