

Introduction to Layer 7-filter

**Presenter:
Andrzej Bober**

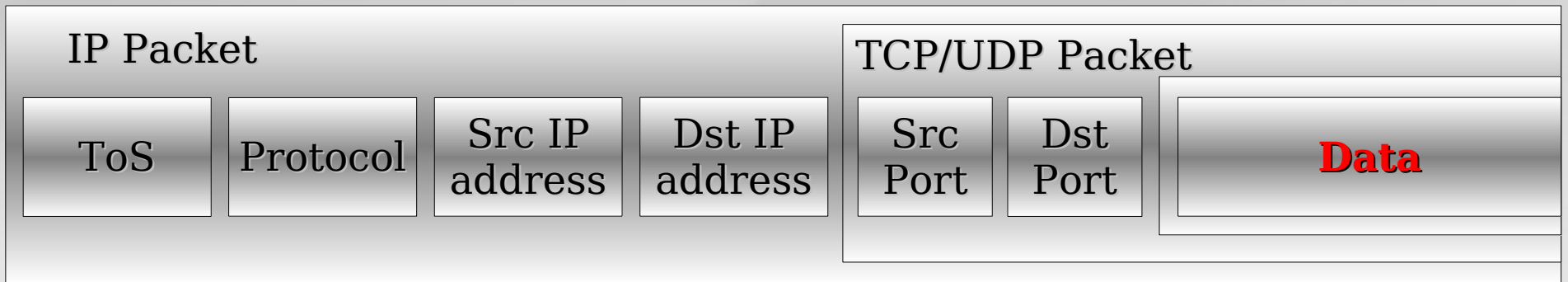
Questions

- What is and why we need L7 Filter
- How L7 works
- Regular expressions - example
- How to apply L7 on Mikrotik router
- Advantages and disadvantages

Traffic marking

Whatever we try to achieve, implement some security rules, bandwidth control or any accounting, first of all we need to define criteria on which we will classify our interesting traffic, from this first step depends the whole value of our configuration.

Traffic classification field

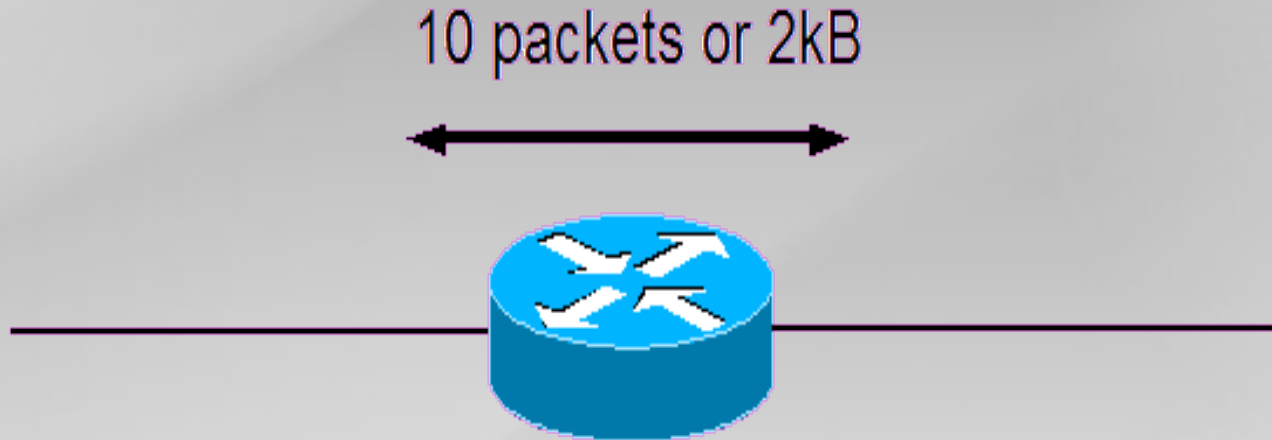


Analyse packet flow content

Recognise any traffic
Protocols
File types
Malware

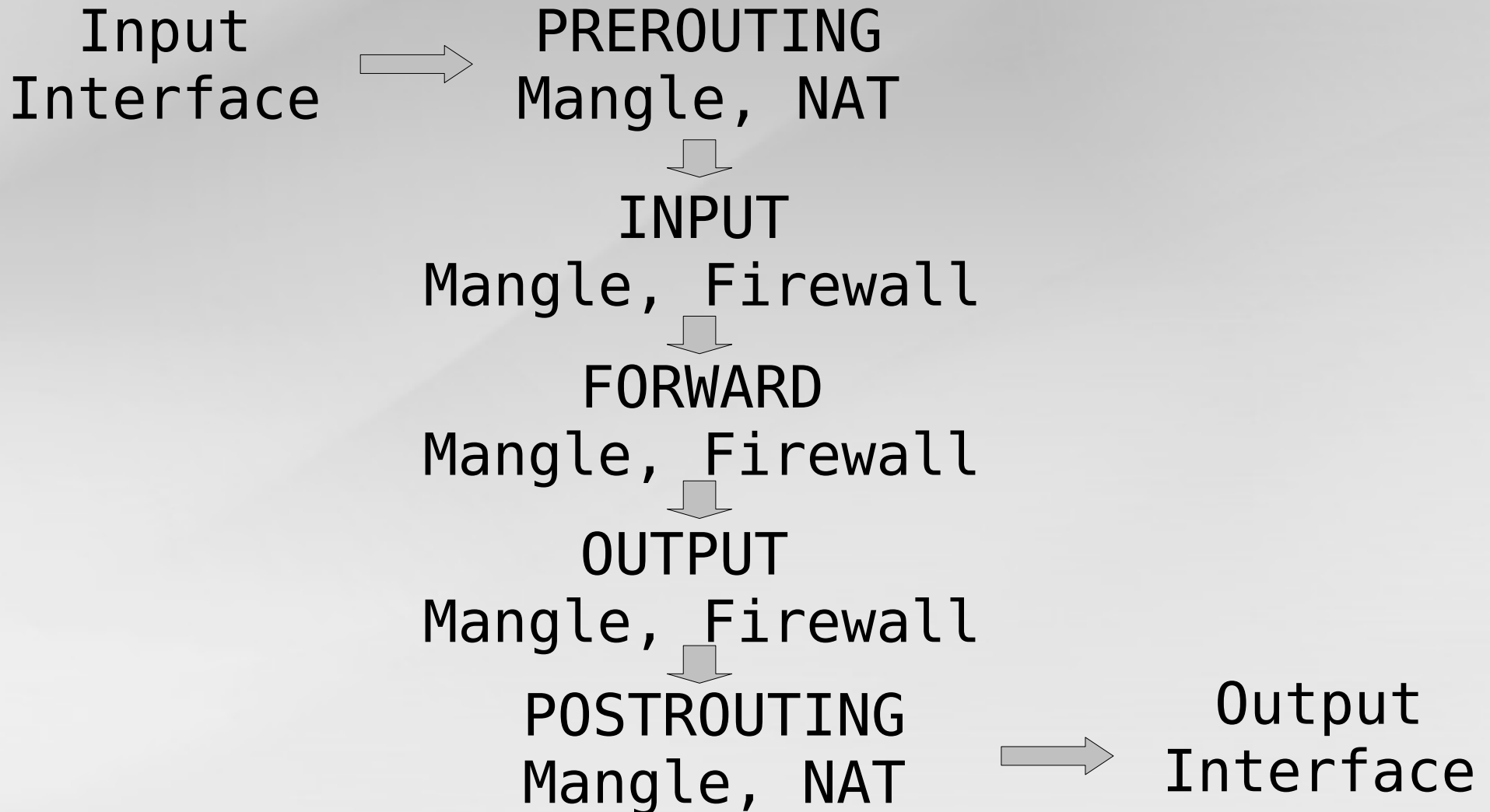
Almost 150 pattern ready

Analyse connections



L7 matcher, by default searching for pattern at the first 10 packets or 2kB, whichever is smaller (this value can not be changed). Any match made within this time is applied to the rest of the connection as well.

Traffic flow simplified diagram



Best place to apply

Most of patterns to correct work
need to see both sides of
connection, that is why the best
place to apply is on forwarding
chain.

Regular Expressions

L7 uses regular expressions to investigate the content within an individual connection.

Regular expression is text string for describing a search pattern.

Searching examples:

"hello" messages such as "220 ftp server ready", "* ok", or "HTTP/1.1 200 ok".

Regular Expressions

quick reference chart

„^“ (caret) Matches the beginning of input

„\$“ Matches the end of input

„.“ Matches any single character

„?“ 0 or 1 occurrences of preceding string

„*“ (star) 0 or more occurrences of preceding string

„[...]“ Matches any one of the enclosed characters e.g. ca[tr] matches cat and car

„|“ (pipe) Logical „or“, match either the part on the left side, or the part on the right side

Regular Expression

Useful things

`[\x09-\x0d -~]` printable characters,
including whitespace

`[\x09-\x0d]` any whitespace

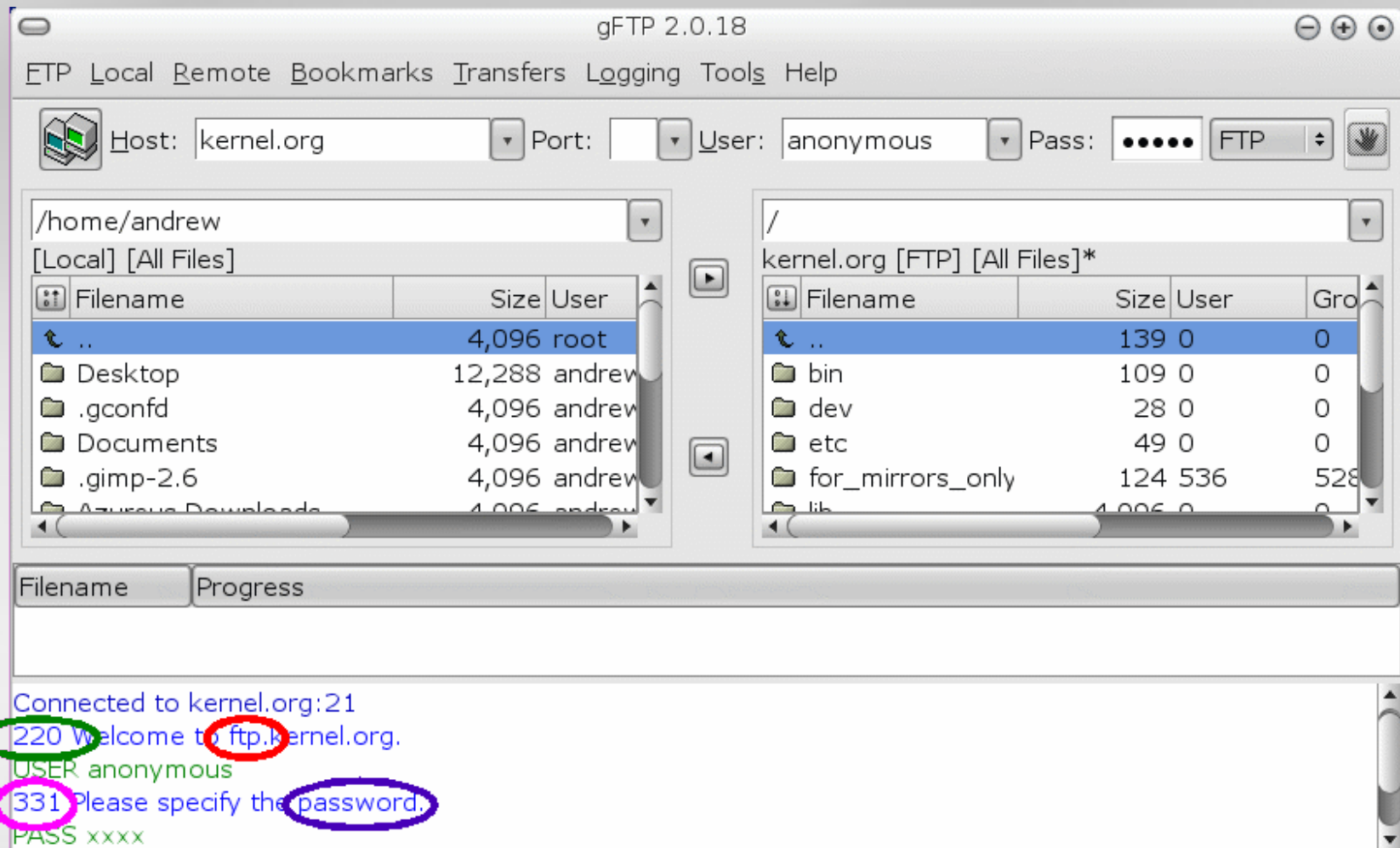
`[!-~]` non-whitespace printable characters

How to write own pattern

- Find and read specifications for the protocol you wish to match. For internet standard check RFCs, if it is proprietary protocol make general web searching for reverse-engineering specification.
- Use any network sniffer (e.g. Wireshark), to capture and watch packets of this protocol go by in a typical session of its use.
- Write a pattern that will reliably match one of the first few packets that are sent in your protocol. Test it.

FTP pattern

`^220[\x09-\x0d -~]*ftp|331[\x09-\x0d -~]*password`



FTP connection

The image shows a Wireshark capture of an FTP connection. The main pane displays a list of packets, with packet 4 selected. The packet list pane shows the following details:

| No. | Time | Source | Destination | Protocol | Info |
|-----|--------|----------------|----------------|----------|---|
| 1 | 0.0000 | 10.0.0.5 | 204.152.191.37 | TCP | 60528 > ftp [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=238718 TSER |
| 2 | 0.1884 | 204.152.191.37 | 10.0.0.5 | TCP | ftp > 60528 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV= |
| 3 | 0.1885 | 10.0.0.5 | 204.152.191.37 | TCP | 60528 > ftp [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=238775 TSER=20 |
| 4 | 0.3761 | 204.152.191.37 | 10.0.0.5 | FTP | Response: 220 Welcome to ftp.kernel.org. |
| 5 | 0.3762 | 10.0.0.5 | 204.152.191.37 | TCP | 60528 > ftp [ACK] Seq=1 Ack=33 Win=5888 Len=0 TSV=238831 TSER=2 |
| 6 | 0.3763 | 10.0.0.5 | 204.152.191.37 | FTP | Request: USER anonymous |
| 7 | 0.5621 | 204.152.191.37 | 10.0.0.5 | TCP | ftp > 60528 [ACK] Seq=33 Ack=17 Win=5888 Len=0 TSV=2096903795 T |
| 8 | 0.5626 | 204.152.191.37 | 10.0.0.5 | FTP | Response: 331 Please specify the password. |
| 9 | 0.5628 | 10.0.0.5 | 204.152.191.37 | FTP | Request: PASS andrew@localhost |
| 10 | 0.7500 | 204.152.191.37 | 10.0.0.5 | FTP | Response: 230-\t\t\t\t Welcome to the |
| 11 | 0.7503 | 204.152.191.37 | 10.0.0.5 | FTP | Response: 230- |
| 12 | 0.7504 | 10.0.0.5 | 204.152.191.37 | TCP | 60528 > ftp [ACK] Seq=40 Ack=100 Win=5888 Len=0 TSV=238943 TSER |

The packet details pane for packet 4 shows the following structure:

- Options: (12 bytes)
- [SEQ/ACK analysis]
- File Transfer Protocol (FTP)
- 220 Welcome to ftp.kernel.org.\r\n
 - Response code: Service ready for new user (220)
 - Response arg: Welcome to ftp.kernel.org.

The packet bytes pane shows the raw data for packet 4:

```
0000  00 10 0e cd 04 10 00 40 10 20 00 01 00 00 43 00  ..d...@ . . . .L.  
0010  00 54 11 50 40 00 38 06 9b 91 cc 98 bf 25 0a 00  .T.P@.8. ....%..  
0020  00 05 00 15 ec 70 3b 7e 24 e8 ef 95 cb 24 80 18  .....p;~ $.$.$.  
0030  00 2e 2b 82 00 00 01 01 08 0a 7c fc 35 b9 00 03  ..+..... ..|.5...  
0040  a4 b7 32 32 30 20 57 65 6c 63 6f 6d 65 20 74 6f  ..220 We lcome to  
0050  20 66 74 70 2e 6b 65 72 6e 65 6c 2e 6f 72 67 2e  ftp.ker nel.org.  
0060  0d 0a  ..
```

Looking for pattern?

Pattern libraries can be found on:

http://protocolinfo.org/wiki/Main_Page

<http://l7-filter.sourceforge.net/protocols>

Script for Mikrotik with common programs
list:

www.mikrotik.com/download/l7-protos.rsc

Layer 7 - CLI configuration

To define strings you will be looking for, add Regexp strings to the protocols menu.

```
/ip firewall layer7-protocol add=
```

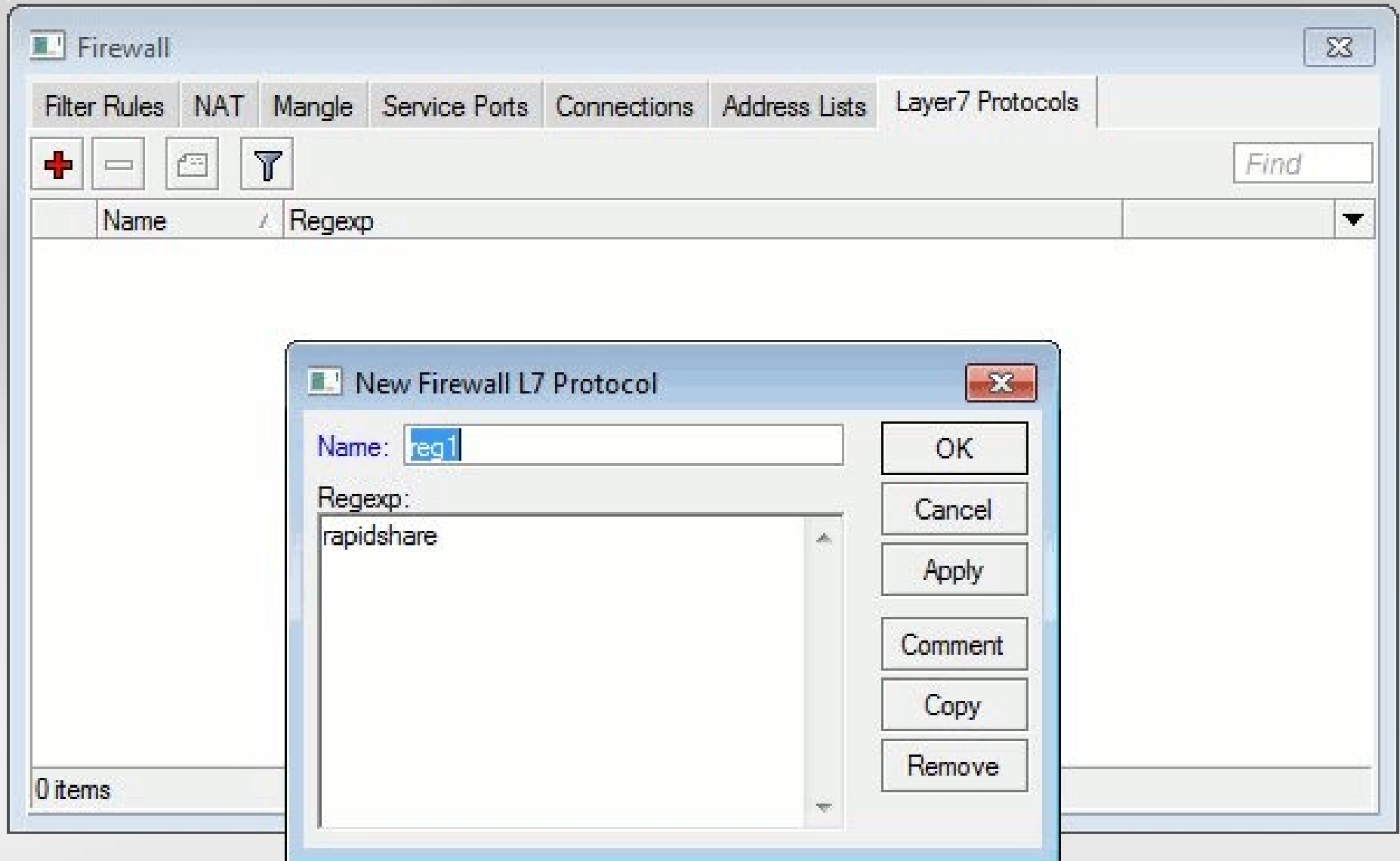
Then, apply defined protocols in firewall:

```
/ip firewall filter add layer7-protocol=
```

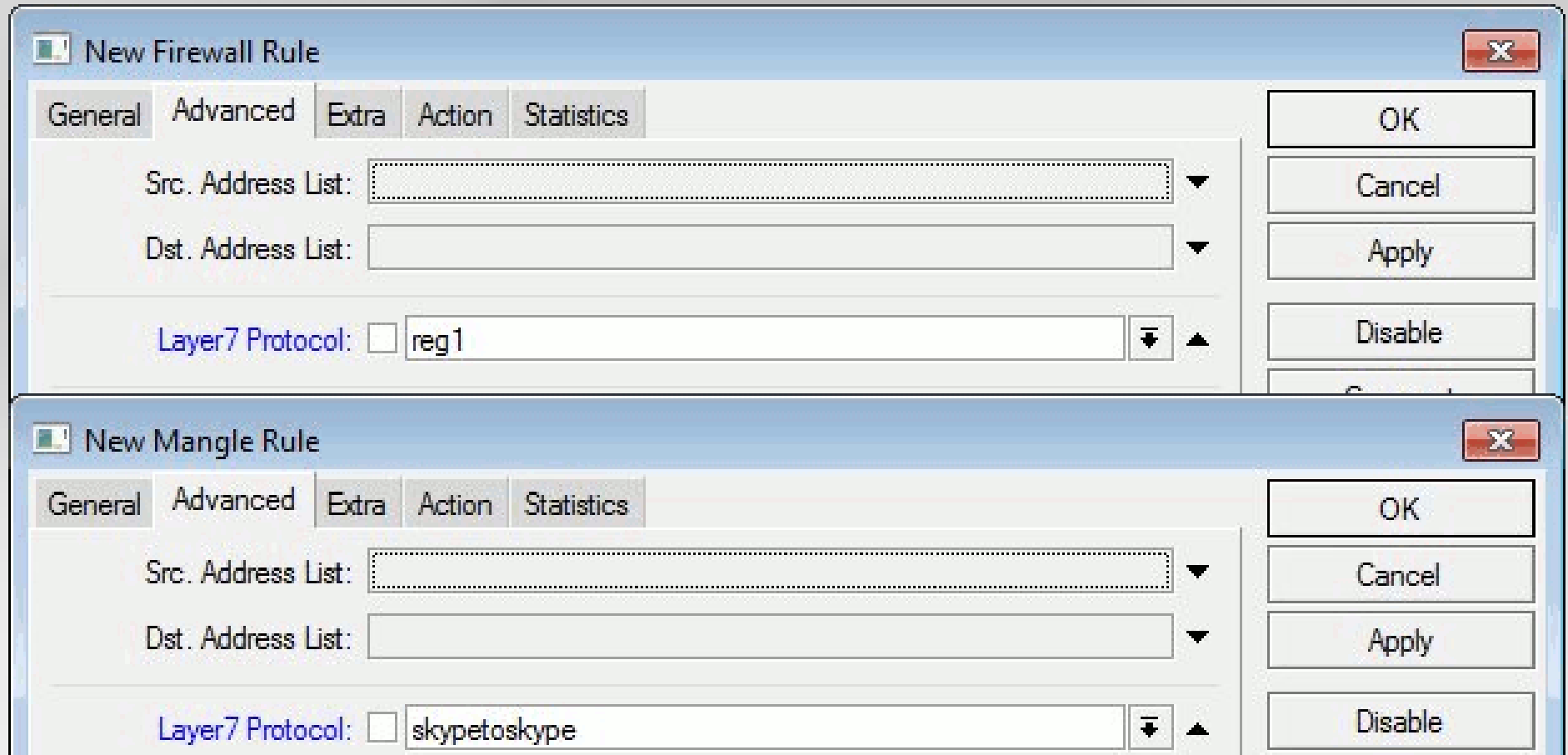
For traffic marking:

```
/ip firewall mangle add layer7-protocol=
```


Winbox configuration



Firewall and Mangle



Encrypted traffic

L7 do not work on SSL tunnel, this is because the only clear text packet following the TCP/IP handshake is the SSL server certificate.

One thing you might try is to look for criteria in the certificate, that is, you might decide not to trust individual certification authorities.

Summary

Advantages

Any traffic

High accuracy

Distinguish packets
working on the
same port

Disadvantages

CPU consumption

Can not recognize
encrypted traffic

References

L7-filter project website:
<http://l7-filter.sourceforge.net/>

Blocking protocols at Layer 7 with
the L7 patch by Jörg Harmuth