



# POD Translation by *pod2pdf*

---

[ajf@afco.demon.co.uk](mailto:ajf@afco.demon.co.uk)

*tailfilter*



# Table of Contents

## tailfilter

NAME	1
Tailfilter	1
SYNOPSIS	1
DESCRIPTION	1
Zen and the Art of Reading Firewall Logs...	1
Note on Logfiles:	1
1.	1
2.	1
Caching /etc/services and /etc/protocols for fast port/protocol lookups:	2
SETUP	2
Tailfilter For Dummies	2
OPTIONS AND USAGE	2
-t, —iptables	2
-l (n), —lines=(n)	2
-c, —compressed	2
-n, —numeric	3
-q, —quiet	3
-h, —help, —usage	3
—docs	3
-v, —version	3
Notes:	3
1.	3
2.	3
3.	3
BUGS	3
CHANGES	3
Mon Mar 3 2003	4
Sat Feb 15 2003	4
Sun Dec 1 2002	4
Mon Nov 4 2002	4
Sun Oct 20 2002	4
Thu Aug 29 2002	4
Sat Aug 24 2002	4
Wed Aug 21 2002	4
Tue Aug 20 2002	5
Thu Aug 15 2002	5
Mon Aug 12 2002	5
Mon Aug 5 2002	5
Sat Aug 3 2002	5
Fri Aug 2 2002	5
Thu Aug 1 2002	5
Wed Jul 31 2002	5
Tue Jul 30 2002	6
Sun Jul 28 2002	6
AUTHOR	6
Author	6
Email	6
Last Update	6
COPYRIGHT	6



## NAME

### Tailfilter

## SYNOPSIS

Tailfilter is something that started off as a perl one-liner, took on a life of its own, and swiftly grew out of control because I couldn't stop thinking of ways to enhance the original idea.

Essentially, it's a logfile filter that reformats the output from the log and 'pretty-prints' a more legible arrangement that lends itself better to rapid and/or cursory analysis, as well as offering immediate notification of new events as they occur. It additionally caches the results of DNS lookups, as well as the TCP-based services in /etc/services to speed up the info lookups considerably.

To use it, simply do one of the following (or similar):

```
tail -f /var/log/messages |tailfilter

sudo tail -n 50 -f /var/log/messages |./tailfilter -l 40 -c
```

(depending on where you keep tailfilter :-) *Also see Setup below.*

## DESCRIPTION

### Zen and the Art of Reading Firewall Logs...

As a relative newcomer to Linux, I've found the logging information from a straight `tail -f` to be a little mysterious looking, and not at all conducive to casual inspection, and wanted to find a way to filter out the **information** I was interested in. It really did start as a simple Perl one-liner though, if you can believe it. :-)

Basically, tailfilter filters your logfiles for packet log information from the firewall and reformats them to look a little nicer, does a cached hostname lookup on the IP in question, and adds a columnised report layout, (thanks to Perl's `format STDOUT` command), which just goes one step further to making the information in the logfiles more understandable. Nothing in your logs is actually changed, mind you, just 'filtered' through a little Perl magic. ;-)

Tailfilter was designed to work with the output of iptables/ipchains, and your logfile entries should look like this:

ipchains:

```
Jul 28 06:36:37 pcp01487622pcs kernel: Packet log: input DENY eth0 PROTO=6
68.82.244.101:4434 68.82.41.167:80 L=48 S=0x00 I=23517 F=0x4000 T=119 SYN (#17)
```

iptables:

```
Aug 18 04:05:17 anakin kernel: Blocked incoming port: IN=eth0 OUT=
MAC=00:10:dc:21:ad:36:00:00:c5:7d:5d:2c:08:00 SRC=4.2.2.1 DST=216.163.77.13
LEN=72 TOS=0x00 PREC=0x00 TTL=242 ID=2208 DF PROTO=UDP SPT=53 DPT=32919 LEN=52
```

### Note on Logfiles:

1. If you have a different format in /var/log/messages for your firewall entries, this script will require adjustment to account for those differences. Please let me know if you do, and I'll see if there's anything I can do about it.
2. **Important:** Your firewall **MUST** be set to be logging some packets for whatever reasons you decide, or this won't find anything in your logs to parse. *i.e. it won't do nuttin'.* :-)

You can do a quick check with this:

```
grep "kernel:" /var/log/messages |less
```

Look for entries similar to the above two samples.

**Caching /etc/services and /etc/protocols for fast port/protocol lookups:**

During the reading in of /etc/services and /etc/protocols, it will warn you if it finds any service name that the current regex does not catch (and it will skip that line.) You can study the warning line and tweak the regex accordingly for your system.

**SETUP****Tailfilter For Dummies**

You will need to either be set up to use sudo, or you must 'su -' to root, in order to have access to the /var/log/messages file through tail.

You will need to ( `chmod 755 tailfilter` ) in order to make it executable.

You will need to either place it in your \$PATH (such as the ~/bin, or /usr/bin/ directories) or use the local directory convention of ./tailfilter, in order to pipe tail's output to it. If it is not in your path, the command would look more like :

```
tail -f /var/log/messages |/path/to/tailfilter
```

```
tail -f /var/log/messages |./tailfilter
```

If you are using the iptables firewall, ( `/sbin/lsmmod |egrep '^ip'` ), you will need to add the `-iptables,` or `-t` switch to the end.

*If you can't figure this out, you really shouldn't have root access ;-)*

**OPTIONS AND USAGE**

Usually it is sufficient merely to fire up an xterm and

```
sudo tail -f /var/log/messages |tailfilter
```

however, the following command-line options are also available:

**-t, --iptables**

Parse for the iptables logformat instead of ipchains.

By default, tailfilter assumes you are using ipchains. Using this switch, tailfilter switches gears to parse the logfile format generated by iptables, which is significantly different.

**-l (n), --lines=(n)**

Sets how many desired lines per 'page' of output (default 25).

However, if you have more screen real-estate and typically grow your xterm windows to more than 25 lines, you can pass the **-l (n)** switch to tailfilter, to specify how many lines to print before re-printing the top-of-form headers. In other words, you're telling tailfilter how many lines fit on a page.

```
sudo tail -f /var/log/messages |tailfilter -l 45
```

Or, if you're going to chuck a whole pile of logfile at it with something like:

```
sudo cat /var/log/messages* |tailfilter -ql 100 > /tmp/firewall_logreport
```

Adjust to suit your individual aesthetics. :-)

**-c, --compressed**

Output is compressed down to 80 columns.

With this option, output is compressed down to 80 columns, with IP addresses removed, and only showing up when they are unable to be resolved. Since this is short enough to, I included a test for some

common failure reasons in the output if this happens.

```
sudo tail -f /var/log/messages |tailfilter -c
```

### **-n, --numeric**

Numeric-only output.

With this option, output is compressed even further, and supplies only numeric information. Date, Time, IP address, Port, Protocol. Protocol was left in as alpha for now, as it's somewhat difficult to account for when some firewalls are configured to spit out PROTO=TCP, so it was left the same as standard output.

```
sudo tail -f /var/log/messages |tailfilter -n
```

### **-q, --quiet**

Quiet output; console beeps on receipt of packet are suppressed.

Here, the ASCII 'BEL' (i.e. the 'beep') character is not printed at the end of each line of filtered output, which can be useful both when streaming output into a file for later reading, or if you just don't want to hear the beeps when something hits the logs. By default, tailfilter beeps as each packet hits the filter and is output, notifying you of the intrusion.

```
sudo cat /var/log/messages* |tailfilter -ql 100 >/tmp/filtered.txt
```

### **-h, --help, --usage**

Print a short usage message to the console detailing the command line options in brief and exit.

### **--docs**

Invoke Perldoc on the user's behalf and present the full Tailfilter documentation and exit.

### **-v, --version**

Print the current version and the date of the last update, and exit.

### **Notes:**

1. While the `-l -q -t` switches can be combined with either `-c` or `-n`, `-c` and `-n` are mutually exclusive, and `-n` will override.
2. It is possible to bundle switches together thusly, just like a normal shell:  
`tailfilter -qcl50 -iptables`
3. *If you're getting the impression that I advocate the use of **sudo**, then you are probably correct.*

## **BUGS**

- This script was originally written around Red Hat Linux 7.2 and ipchains firewall. Compatibility with other distributions, firewalls, and configurations is in no way guaranteed, although iptables parsing has been successfully added.
- Not a bug really, just a warning: By default, a 'page' cannot be less than 10 lines, so if you do `-l 3`, it will simply default back to 25 lines per page instead of 3 (which would be silly anyway).
- Due to the fact that Perl's `format STDOUT` is globalized and **not** for some mysterious reason, lexically scoped, you normally get "format STDOUT redefined at line ###" errors, what with the way that this works, requiring different columnised reports for the differing switches. Effort needed to be made to suppress these warnings. This is something I hope the Perl folks get cleared up at some time in the near future. Formats aren't used all that often these days, but for something like this, they're perfect.

In this implementation I had to suppress the warnings, so it's dealt with, but still annoying code-wise. :-)

## **CHANGES**

**Mon Mar 3 2003**

- Added From-port to output to catch certain instances where it would be desirable to know the port number that the logged event is coming from.

**Sat Feb 15 2003**

- Finally figured out what I was doing wrong with long-domain-names not wrapping to the next line and truncating the previous variable in the format.  
For some really odd reason it wasn't working with the array variable (`$v[7] || $v[6]`), so I defined a local lexical (`$mydomain`) to the `swrite` sub, and had the format use `IT` instead.  
Now it works properly. \*shrug\* weird.
- I'm pondering using `Socket.pm` and `gethostbyaddr( inet_aton($ip), AF_INET )` instead of the `'host $ip 2&1'` + massive cleanup that I'm using now. The drawback is that you don't get the (SERVFAIL) (NXDOMAIN) type response messages, when you do that. It WOULD greatly simplify the code, but at the expense of some functionality I currently feel that I like.  
I'll mull it over, and test things to see what I can come up with.
- Perl 5.8.0 now warns about 'tainted' \$0 in EXEC, so I'll have to find a good way to launder the filepath to the tailfilter executable before running `exec ("perldoc", $0)` for the `—docs` switch. This will eventually be a fatal error, rather than a warning, so this needs to be done soonish.

**Sun Dec 1 2002**

- Slight documentation and usage info tweaking.

**Mon Nov 4 2002**

- Minor code cleanup tweaking for readability.
- More sensible die message with bad command-line options. :-)
- Added `-h` and `-v` switches (obvious in retrospect), and accompanying documentation and usage edits.

**Sun Oct 20 2002**

- Minor documentation tweaks and a few minor code and/or internal comment tweaks. No changes in functionality.
- Removed extraneous e-mail addresses to use solely the primary and hopefully permanent contact address.

**Thu Aug 29 2002**

- Reorganized the docs a little to make the bit about the `—iptables` switch slightly more obvious to the new user.

**Sat Aug 24 2002**

- Reformatted all command-line switches to have `—long` counterparts.  
As a consequence, some major/minor code funkiness in the earlier versions has now been avoided completely.
- Added `—usage/—help`, `—version`, and `—docs` (which evokes `perldoc /path/to/tailfilter` on the user's behalf).
- Documentation changes to reflect the above adjustments.

**Wed Aug 21 2002**

- Documentation twiddling to overcome some of the earlier rushed phrasing and repetition, and make things clearer.

**Tue Aug 20 2002**

- Added -t switch to optionally parse iptables logformat instead of ipchains
- Adjusted documentation accordingly with above

**Thu Aug 15 2002**

- Added the -q switch to suppress the beeps whenever a packet hits the log.
- Minor documentation tweaks.
- Added the fancy tailfilter logo to the html page.

**Mon Aug 12 2002**

- Fixed minor bug in cleanup regexes, where multiply-aliased IP's weren't being split properly, and displayed.  
Slight change to parsing order to both make more sense otherwise, and also to work with the above more cleanly.

**Mon Aug 5 2002**

- Added comment-folding to the code to make editing in vim easier. minor doc cleanup. no other changes.
- Added a catch for older versions of 'host' that dumped errors to stderr instead of stdout. (icky) (thanks again to honey for catching this one.)

**Sat Aug 3 2002**

- Added a \$SIG{\_\_WARN\_\_} handler to account for the mysterious fact that while Perl 5.6.1 handles the `no warnings 'redefine'; pragma` perfectly well, Perl 5.6.0 doesn't, resulting in a simple but very ugly looking workaround for the time being until I can determine the cause. Thanks go to honey yet again, for catching this one.

**Fri Aug 2 2002**

- Added -c switch for 'compressed' output (reduced to 80 cols) as requested. (by honey from #redhat)

Added -n switch for 'numeric only' output also as requested by honey.

Major hack-a-rama but it works solidly. Please report any unusual output so I can account for it in the output stream cleanup.

- *\*cough\** Minor documentation changes, Typo corrections, some re-formatting to account for differences in how the various *pod2\** utilities produce results for different output formats.

OK, OK, Fine, I practically rewrote a lot of it while twiddling it to be informative and look better.

**Thu Aug 1 2002**

- Minor docfile changes. Testing of the pod doc format with Pod::Pdf to generate pretty docs in .pdf format, via  
`perl -MPod::Pdf -e "pod2pdf('--paper=us-letter', '~/bin/tailfilter')"`

If you haven't tried Alan Fry's module yet for this stuff, do. The output is VERY nice, and VERY printable.

<http://www.cpan.org/authors/id/A/AJ/AJFRY/Pod-Pdf-1.2.tar.gz>

**Wed Jul 31 2002**

- Oh bugger all, I forgot to differentiate between `PROTO=tcp|icmp|udp`. *\*teeth gnashing\** surprised no one noticed this earlier. fixed it to do tcp and udp.

Later, as part of the PROTO update, I added cached lookups for /etc/protocols since doing this individually or adding them to the code manually would be silly considering the file exists already. As a consequence, icmp should now be reported

- Minor changes to column formatting to save some horizontal space. Date is now presented in the column headers, and updated whenever they get reprinted. Times may overlap over a day, but it'll be obvious that the clock rolled over. Also changed some of how proto/port/service is displayed. I'm limited somewhat in this by how Perl format's work. It could be prettier. I'll work on it.
- Minor bug. not filtering in\_addr.arpa case insensitively. fixed. (thanks to honey from #redhat for catching this one for me!)

### Tue Jul 30 2002

- Added taint-checking (suggested by alphablue. thanks guy!) since the program is running under sudo, this makes sense.

### Sun Jul 28 2002

- added host caching to prevent dns-lookup storms when someone nmaps your system.
- added lookups to /etc/services (also cached in a hash lookup table) to report the name of the port being probed in the firewall log
- reformatted the output to pretty-print via format STDOUT
- added the -l (n) switch so you could specify how many 'lines per page' before it reprints the format STDOUT\_TOP header
- changed all the #comment docs to use the =pod format so you can read them all pretty via perldoc /path/to/tailfilter. :)

## AUTHOR

### Author

Scott R. Godin

### Email

mactech@webdragon.net

### Last Update

Mon Mar 3 04:14:29 EST 2003

## COPYRIGHT

Copyright (c) 2001 Scott R. Godin. All rights reserved. This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.