NAME

editcap − Edit and/or translate the format of capture files

SYNOPSIS

editcap [**−a** <frame:comment>] [**−A** <start time>] [**−B** <stop time>] [**−c** <packets per file>]
[**−C** [offset:]<choplen>] [**−E** <error probability>] [**−F** <file format>] [**−h**] [**−i** <seconds per file>]
[**−o** <change offset>] [**−L**] [**−r**] [**−s** <snaplen>] [**−S** <strict time adjustment>]
[**−t** <time adjustment>] [**−T** <encapsulation type>] [**−v**] [**−−inject−secrets** <secrets type>,<file>]
[**−−discard−all−secrets**] [**−−capture−comment** <comment>] [**−−discard−capture−comment**] *infile*
outfile [*packet#*[*−packet#*] ...]

editcap −d −D <dup window> **−w** <dup time window> [**−v**] [**−I** <bytes to ignore>]
[**−−skip−radiotap−header**] *infile outfile*

editcap [**−V**]

DESCRIPTION

Editcap is a program that reads some or all of the captured packets from the *infile*, optionally converts them in various ways and writes the resulting packets to the capture *outfile* (or outfiles).

By default, it reads all packets from the *infile* and writes them to the *outfile* in pcapng file format.

The **−A** and **−B** option allow you to limit the time range from which packets are read from the *infile*.

An optional list of packet numbers can be specified on the command tail; individual packet numbers separated by whitespace and/or ranges of packet numbers can be specified as *start−end*, referring to all packets from *start* to *end*. By default the selected packets with those numbers will *not* be written to the capture file. If the **−r** flag is specified, the whole packet selection is reversed; in that case *only* the selected packets will be written to the capture file.

Editcap can also be used to remove duplicate packets. Several different options (**−d**, **−D** and **−w**) are used to control the packet window or relative time window to be used for duplicate comparison.

Editcap can be used to assign comment strings to frame numbers.

Editcap is able to detect, read and write the same capture files that are supported by **Wireshark**. The input file doesn't need a specific filename extension; the file format and an optional gzip, zstd or lz4 compression will be automatically detected. Near the beginning of the DESCRIPTION section of wireshark(1) or https://www.wireshark.org/docs/man−pages/wireshark.html is a detailed description of the way **Wireshark** handles this, which is the same way **Editcap** handles this.

Editcap can write the file in several output formats. The **−F** flag can be used to specify the format in which to write the capture file; **editcap −F** provides a list of the available output formats.

OPTIONS

−a <framenum:comment>

For the specified frame number, assign the given comment string. Can be repeated for multiple frames. Quotes should be used with comment strings that include spaces.

−A <start time>

Reads only the packets whose timestamp is on or after start time. The time is given in ISO 8601 format, either YYYY−MM−DD HH:MM:SS[.nnnnnnnnn][Z|±hh:mm] or YYYY−MM−DDTHH:MM:SS[.nnnnnnnnn][Z|±hh:mm] . The fractional seconds are optional, as is the time zone offset from UTC (in which case local time is assumed). Unix epoch timestamps (floating

point format) are also accepted.

−B <stop time>

Reads only the packets whose timestamp is before stop time. The time is given in ISO 8601 format, either YYYY−MM−DD HH:MM:SS[.nnnnnnnnn][Z|±hh:mm] or YYYY−MM−DDTHH:MM:SS[.nnnnnnnnn][Z|±hh:mm] . The fractional seconds are optional, as is the time zone offset from UTC (in which case local time is assumed). Unix epoch timestamps (floating point format) are also accepted.

−c <packets per file>

Splits the packet output to different files based on uniform packet counts with a maximum of <packets per file> each.

Each output file will be created with an infix _nnnnn[_YYYYmmddHHMMSS] inserted before the file extension (which may be null) of *outfile*. The infix consists of the ordinal number of the output file, starting with 00000, followed by the timestamp of its first packet. The timestamp is omitted if the input file does not contain timestamp information.

After the specified number of packets is written to the output file, the next output file is opened. The default is to use a single output file. This option conflicts with **−i**.

−C [offset:]<choplen>

Sets the chop length to use when writing the packet data. Each packet is chopped by <choplen> bytes of data. Positive values chop at the packet beginning while negative values chop at the packet end.

If an optional offset precedes the <choplen>, then the bytes chopped will be offset from that value. Positive offsets are from the packet beginning, while negative offsets are from the packet end.

This is useful for chopping headers for decapsulation of an entire capture, removing tunneling headers, or in the rare case that the conversion between two file formats leaves some random bytes at the end of each packet. Another use is for removing vlan tags.

Note

This option can be used more than once, effectively allowing you to chop bytes from up to two different areas of a packet in a single pass provided that you specify at least one chop length as a positive value and at least one as a negative value. All positive chop lengths are added together as are all negative chop lengths.

−d

Attempts to remove duplicate packets. The length and MD5 hash of the current packet are compared to the previous four (4) packets. If a match is found, the current packet is skipped. This option is equivalent to using the option **−D 5**.

−D <dup window>

Attempts to remove duplicate packets. The length and MD5 hash of the current packet are compared to the previous <dup window> − 1 packets. If a match is found, the current packet is skipped.

The use of the option **−D 0** combined with the **−v** option is useful in that each packet's Packet number, Len and MD5 Hash will be printed to standard out. This verbose output (specifically the MD5 hash strings) can be useful in scripts to identify duplicate packets across trace files.

The <dup window> is specified as an integer value between 0 and 1000000 (inclusive).

Note

Specifying large <dup window> values with large tracefiles can result in very long processing times for **editcap**.

−E <error probability>

Sets the probability that bytes in the output file are randomly changed. **Editcap** uses that probability (between 0.0 and 1.0 inclusive) to apply errors to each data byte in the file. For instance, a probability of 0.02 means that each byte has a 2% chance of having an error.

This option is meant to be used for fuzz−testing protocol dissectors.

−F <file format>

Sets the file format of the output capture file. **Editcap** can write the file in several formats, **editcap −F** provides a list of the available output formats. The default is the **pcapng** format.

−h

Prints the version and options and exits.

−i <seconds per file>

Splits the packet output to different files based on uniform time intervals using a maximum interval of <seconds per file> each. Floating point values (e.g. 0.5) are allowed.

Each output file will be created with an infix _nnnnn[_YYYYmmddHHMMSS] inserted before the file extension (which may be null) of *outfile*. The infix consists of the ordinal number of the output file, starting with 00000, followed by the timestamp of its first packet. The timestamp is omitted if the input file does not contain timestamp information.

After packets for the specified time interval are written to the output file, the next output file is opened. The default is to use a single output file. This option conflicts with **−c**.

−I <bytes to ignore>

Ignore the specified number of bytes at the beginning of the frame during MD5 hash calculation, unless the frame is too short, then the full frame is used. Useful to remove duplicated packets taken on several routers (different mac addresses for example) e.g. −I 26 in case of Ether/IP will ignore ether(14) and IP header(20 − 4(src ip) − 4(dst ip)). The default value is 0.

−L

Adjust the original frame length accordingly when chopping and/or snapping (in addition to the captured length, which is always adjusted regardless of whether **−L** is specified or not). See also **−C <choplen>** and **−s <snaplen>**.

−o <change offset>

When used in conjunction with −E, skip some bytes from the beginning of the packet from being changed. In this way some headers don’t get changed, and the fuzzer is more focused on a smaller part of the packet. Keeping a part of the packet fixed the same dissector is triggered, that make the fuzzing more precise.

−r

Reverse the packet selection. Causes the packets whose packet numbers are specified on the command line to be written to the output capture file, instead of discarding them.

−s <snaplen>

Sets the snapshot length to use when writing the data. If the **−s** flag is used to specify a snapshot length, packets in the input file with more captured data than the specified snapshot length will have only the amount of data specified by the snapshot length written to the output file.

This may be useful if the program that is to read the output file cannot handle packets larger than a certain size (for example, the versions of snoop in Solaris 2.5.1 and Solaris 2.6 appear to reject Ethernet packets larger than the standard Ethernet MTU, making them incapable of handling gigabit Ethernet captures if jumbo packets were used).

−−seed <seed>

When used in conjunction with −E, set the seed for the pseudo−random number generator. This is useful for recreating a particular sequence of errors.

−−skip−radiotap−header

Skip the radiotap header of each frame when checking for packet duplicates. This is useful when processing a capture created by combining outputs of multiple capture devices on the same channel in the vicinity of each other.

−S <strict time adjustment>

Time adjust selected packets to ensure strict chronological order.

The <strict time adjustment> value represents relative seconds specified as *seconds*[.*fractional seconds*].

As the capture file is processed each packet's absolute time is *possibly* adjusted to be equal to or greater than the previous packet's absolute timestamp depending on the <strict time adjustment> value.

If <strict time adjustment> value is 0 or greater (e.g. 0.000001) then **only** packets with a timestamp less than the previous packet will adjusted. The adjusted timestamp value will be set to be equal to the timestamp value of the previous packet plus the value of the <strict time adjustment> value. A <strict time adjustment> value of 0 will adjust the minimum number of timestamp values necessary to ensure that the resulting capture file is in strict chronological order.

If <strict time adjustment> value is specified as a negative value, then the timestamp values of **all** packets will be adjusted to be equal to the timestamp value of the previous packet plus the absolute value of the <strict time adjustment> value. A <strict time adjustment> value of −0 will result in all packets having the timestamp value of the first packet.

This feature is useful when the trace file has an occasional packet with a negative delta time relative to the previous packet.

−t <time adjustment>

Sets the time adjustment to use on selected packets. If the **−t** flag is used to specify a time adjustment, the specified adjustment will be applied to all selected packets in the capture file. The adjustment is specified as *seconds*[.*fractional seconds*]. For example, **−t** 3600 advances the timestamp on selected packets by one hour while **−t** −0.5 reduces the timestamp on selected packets by one−half second.

This feature is useful when synchronizing dumps collected on different machines where the time difference between the two machines is known or can be estimated.

−T <encapsulation type>

Sets the packet encapsulation type of the output capture file. If the **−T** flag is used to specify an encapsulation type, the encapsulation type of the output capture file will be forced to the specified type. **editcap −T** provides a list of the available types. The default type is the one appropriate to the encapsulation type of the input capture file.

Note: this merely forces the encapsulation type of the output file to be the specified type; the packet headers of the packets will not be translated from the encapsulation type of the input capture file to the specified encapsulation type (for example, it will not translate an Ethernet capture to an FDDI capture if an Ethernet capture is read and '**−T fddi**' is specified). If you need to remove/add headers from/to a packet, you will need od(1)/text2pcap(1).

−v

Causes **editcap** to print verbose messages while it's working.

Use of **−v** with the de−duplication switches of **−d**, **−D** or **−w** will cause all MD5 hashes to be printed whether the packet is skipped or not.

−V

Print the version and exit.

−w <dup time window>

Attempts to remove duplicate packets. The current packet's arrival time is compared with up to 1000000 previous packets. If the packet's relative arrival time is *less than or equal to* the <dup time window> of a previous packet and the packet length and MD5 hash of the current packet are the same then the packet to skipped. The duplicate comparison test stops when the current packet's relative arrival time is greater than <dup time window>.

The <dup time window> is specified as *seconds*[.*fractional seconds*].

The [.fractional seconds] component can be specified to nine (9) decimal places (billionths of a second) but most typical trace files have resolution to six (6) decimal places (millionths of a second).

Note
Specifying large <dup time window> values with large tracefiles can result in very long processing times for **editcap**.

Note
The **−w** option assumes that the packets are in chronological order. If the packets are NOT in chronological order then the **−w** duplication removal option may not identify some duplicates.

−−inject−secrets <secrets type>,<file>

Inserts the contents of <file> into a Decryption Secrets Block (DSB) within the pcapng output file. This enables decryption without requiring additional configuration in protocol preferences.

The file format is described by <secrets type> which can be one of:

tls TLS Key Log as described at https://developer.mozilla.org/NSS_Key_Log_Format *wg* WireGuard Key Log, see https://gitlab.com/wireshark/wireshark/−/wikis/WireGuard#key−log−format

This option may be specified multiple times. The available options for <secrets type> can be listed with **−−inject−secrets help**.

−−discard−all−secrets

Discard all decryption secrets from the input file when writing the output file. Does not discard secrets added by **−−inject−secrets** in the same command line.

−−capture−comment <comment>

Adds the given comment to the output file, if supported by the output file format. New comments will be added *after* any comments present in the input file unless **−−discard−capture−comment** is also specified.

This option may be specified multiple times. Note that Wireshark currently only displays the first comment of a capture file.

−−discard−capture−comment

Discard all capture file comments from the input file when writing the output file. Does not discard comments added by **−−capture−comment** in the same command line.

EXAMPLES

To see more detailed description of the options use:

```
editcap −h
```

To shrink the capture file by truncating the packets at 64 bytes and writing it as Sun snoop file use:

```
editcap −s 64 −F snoop capture.pcapng shortcapture.snoop
```

To delete packet 1000 from the capture file use:

```
editcap capture.pcapng sans1000.pcapng 1000
```

To limit a capture file to packets from number 200 to 750 (inclusive) use:

```
editcap −r capture.pcapng small.pcapng 200−750
```

To get all packets from number 1−500 (inclusive) use:

```
editcap −r capture.pcapng first500.pcapng 1−500
```

or

```
editcap capture.pcapng first500.pcapng 501−9999999
```

To exclude packets 1, 5, 10 to 20 and 30 to 40 from the new file use:

```
editcap capture.pcapng exclude.pcapng 1 5 10−20 30−40
```

To select just packets 1, 5, 10 to 20 and 30 to 40 for the new file use:

```
editcap −r capture.pcapng select.pcapng 1 5 10−20 30−40
```

To remove duplicate packets seen within the prior four frames use:

```
editcap −d capture.pcapng dedup.pcapng
```

To remove duplicate packets seen within the prior four frames while skipping radiotap headers use:

```
editcap −d −−skip−radiotap−header capture.pcapng dedup.pcapng
```

To remove duplicate packets seen within the prior 100 frames use:

```
editcap −D 101 capture.pcapng dedup.pcapng
```

To remove duplicate packets seen *equal to or less than* 1/10th of a second:

```
editcap −w 0.1 capture.pcapng dedup.pcapng
```

To display the MD5 hash for all of the packets (and NOT generate any real output file):

```
editcap −v −D 0 capture.pcapng /dev/null
```

or on Windows systems

```
editcap −v −D 0 capture.pcapng NUL
```

To advance the timestamps of each packet forward by 3.0827 seconds:

```
editcap −t 3.0827 capture.pcapng adjusted.pcapng
```

To ensure all timestamps are in strict chronological order:

```
editcap −S 0 capture.pcapng adjusted.pcapng
```

To introduce 5% random errors in a capture file use:

```
editcap −E 0.05 capture.pcapng capture_error.pcapng
```

To remove vlan tags from all packets within an Ethernet−encapsulated capture file, use:

```
editcap −L −C 12:4 capture_vlan.pcapng capture_no_vlan.pcapng
```

To chop both the 10 byte and 20 byte regions from the following 75 byte packet in a single pass, use any of the 8 possible methods provided below:

```
<−−−−−−−−−−−−−−−−−−−−−−−− 75 −−−−−−−−−−−−−−−−−−−−−−−−−−>

+−−−+−−−−−−−+−−−−−−−−−−+−−−−−−−−−−−−−−+−−−−−−−−−−−−−−−−−−+
| 5 |   10  |     15    |       20       |         25         |
+−−−+−−−−−−−+−−−−−−−−−−+−−−−−−−−−−−−−−+−−−−−−−−−−−−−−−−−−+

1) editcap −C 5:10 −C −25:−20 capture.pcapng chopped.pcapng
2) editcap −C 5:10 −C 50:−20 capture.pcapng chopped.pcapng
3) editcap −C −70:10 −C −25:−20 capture.pcapng chopped.pcapng
4) editcap −C −70:10 −C 50:−20 capture.pcapng chopped.pcapng
5) editcap −C 30:20 −C −60:−10 capture.pcapng chopped.pcapng
6) editcap −C 30:20 −C 15:−10 capture.pcapng chopped.pcapng
7) editcap −C −45:20 −C −60:−10 capture.pcapng chopped.pcapng
```

```
8) editcap −C −45:20 −C 15:−10 capture.pcapng chopped.pcapng
```

To add comment strings to the first 2 input frames, use:

```
editcap −a "1:1st frame" −a 2:Second capture.pcapng capture−comments.pcapng
```

SEE ALSO

pcap(3), wireshark(1), tshark(1), mergecap(1), dumpcap(1), capinfos(1), text2pcap(1), reordercap(1), od(1), pcap−filter(7) or tcpdump(8)

NOTES

This is the manual page for **Editcap** 3.6.2. **Editcap** is part of the **Wireshark** distribution. The latest version of **Wireshark** can be found at https://www.wireshark.org.

HTML versions of the Wireshark project man pages are available at https://www.wireshark.org/docs/man−pages.

AUTHORS

Original Author
Richard Sharpe <sharpe[AT]ns.aus.com>

Contributors
Guy Harris <guy[AT]alum.mit.edu>
Ulf Lamping <ulf.lamping[AT]web.de>