NAME

mergecap − Merges two or more capture files into one

SYNOPSIS

mergecap [**−a**] [**−F** <*file format*>] [**−h**] [**−I** <*IDB merge mode*>] [**−s** <*snaplen*>] [**−v**] [**−V**] **−w** <*outfile*>|− <*infile*> [<*infile*> ...]

DESCRIPTION

Mergecap is a program that combines multiple saved capture files into a single output file specified by the **−w** argument. **Mergecap** knows how to read **pcap** and **pcapng** capture files, including those of **tcpdump**, **Wireshark** and other tools that write captures in those formats.

By default, **Mergecap** writes the capture file in **pcapng** format, and writes all of the packets from the input capture files to the output file.

Mergecap is able to detect, read and write the same capture files that are supported by **Wireshark**. The input files don'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 **Mergecap** handles this.

Mergecap 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, **mergecap −F** provides a list of the available output formats.

Packets from the input files are merged in chronological order based on each frame's timestamp, unless the **−a** flag is specified. **Mergecap** assumes that frames within a single capture file are already stored in chronological order. When the **−a** flag is specified, packets are copied directly from each input file to the output file, independent of each frame's timestamp.

The output file frame encapsulation type is set to the type of the input files if all input files have the same type. If not all of the input files have the same frame encapsulation type, the output file type is set to WTAP_ENCAP_PER_PACKET. Note that some capture file formats, most notably **pcap**, do not currently support WTAP_ENCAP_PER_PACKET. This combination will cause the output file creation to fail.

OPTIONS

−a

Causes the frame timestamps to be ignored, writing all packets from the first input file followed by all packets from the second input file. By default, when **−a** is not specified, the contents of the input files are merged in chronological order based on each frame's timestamp.

Note: when merging, **mergecap** assumes that packets within a capture file are already in chronological order.

−F <file format>

Sets the file format of the output capture file. **Mergecap** can write the file in several formats; **mergecap −F** provides a list of the available output formats. By default this is the **pcapng** format.

−h

Prints the version and options and exits.

−I <IDB merge mode>

Sets the Interface Description Block (IDB) merge mode to use during merging. **mergecap −I** provides

a list of the available IDB merge modes.

Every input file has one or more IDBs, which describe the interface(s) the capture was performed on originally. This includes encapsulation type, interface name, etc. When mergecap merges multiple input files, it has to merge these IDBs somehow for the new merged output file. This flag controls how that is accomplished. The currently available modes are:

none: No merging of IDBs is performed, and instead all IDBs are copied to the merged output file.

all: IDBs are merged only if all input files have the same number of IDBs, and each IDB matches their respective entry in the other files. This is the default mode.

any: Any and all duplicate IDBs are merged into one IDB, regardless of what file they are in.

Note that an IDB is only considered a matching duplicate if it has the same encapsulation type, name, speed, time precision, comments, description, etc.

−s <snaplen>

Sets the snapshot length to use when writing the data. If the **−s** flag is used to specify a snapshot length, frames 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 frames larger than the standard Ethernet MTU, making them incapable of handling gigabit Ethernet captures if jumbo frames were used).

−v

Causes **mergecap** to print a number of messages while it's working.

−V

Print the version and exit.

−w <outfile>|−

Sets the output filename. If the name is '−', stdout will be used. This setting is mandatory.

EXAMPLES

To merge two capture files together into a third capture file, in which the last packet of one file arrives 100 seconds before the first packet of another file, use the following sequence of commands.

First, use:

```
capinfos −aeS a.pcap b.pcap
```

to determine the start and end times of the two capture files, as seconds since January 1, 1970, 00:00:00 UTC.

If a.pcap starts at 1009932757 and b.pcap ends at 873660281, then the time adjustment to b.pcap that would make it end 100 seconds before a.pcap begins would be 1009932757 − 873660281 − 100 = 136272376 seconds.

Thus, the next step would be to use:

```
editcap −t 136272376 b.pcap b−shifted.pcap
```

to generate a version of b.pcap with its time stamps shifted 136272376 ahead.

Then the final step would be to use :

```
mergecap −w compare.pcap a.pcap b−shifted.pcap
```

to merge a.pcap and the shifted b.pcap into compare.pcap.

SEE ALSO

pcap(3), wireshark(1), tshark(1), dumpcap(1), editcap(1), text2pcap(1), pcap−filter(7) or tcpdump(8)

NOTES

Mergecap is based heavily upon **editcap** by Richard Sharpe <sharpe[AT]ns.aus.com> and Guy Harris <guy[AT]alum.mit.edu>.

This is the manual page for **Mergecap** 3.6.2. **Mergecap** 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

Scott Renfro <scott[AT]renfro.org>

Contributors

Bill Guyton <guyton[AT]bguyton.com>