Open Source Used In ThousandEyes Endpoint Agent 1.189.1 ENTERPRISE PROD

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1911965258

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this form.

In your requests please include the following reference number 78EE117C99-1911965258

Contents

1.1 json-cpp 1.9.5

1.1.1 Available under license :

The JsonCpp library's source code, including accompanying documentation,
tests and demonstration applications, are licensed under the following
conditions...

Baptiste Lepilleur and The JsonCpp Authors explicitly disclaim copyright in all
jurisdictions which recognize such a disclaimer. In such jurisdictions,
this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of
2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur and
The JsonCpp Authors, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this
software may choose to accept it either as 1) Public Domain, 2) under the
conditions of the MIT License (see below), or 3) under the terms of dual
Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is
described in clear, concise
terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

==========
Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
==========
(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial
software, affording one all of the rights of Public Domain with the
minor nuisance of being required to keep the above copyright notice
and license text in the source code. Note also that by accepting the
Public Domain "license" you can re-license your copy using whatever
license you like.

1.2 boost 1.81.0

1.2.1 Available under license :

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including

the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3 sqlite 3.40.1

1.3.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
** CAPI3REF: Configuration Options
** KEYWORDS: {configuration option}
**
** These constants are the available integer configuration options that
** can be passed as the first argument to the [sqlite3_config()] interface.
**
** New configuration options may be added in future releases of SQLite.
** Existing configuration options might be discontinued.  Applications
** should check the return code from [sqlite3_config()] to make sure that
** the call worked.  The [sqlite3_config()] interface will return a
** non-zero [error code] if a discontinued or unsupported configuration option
** is invoked.
**
** <dl>
** [[SQLITE_CONFIG_SINGLETHREAD]] <dt>SQLITE_CONFIG_SINGLETHREAD</dt>
** <dd>There are no arguments to this option.  ^This option sets the
** [threading mode] to Single-thread.  In other words, it disables
** all mutexing and puts SQLite into a mode where it can only be used
** by a single thread.   ^If SQLite
 is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** it is not possible to change the [threading mode] from its default
** value of Single-thread and so [sqlite3_config()] will return
** [SQLITE_ERROR] if called with the SQLITE_CONFIG_SINGLETHREAD
** configuration option.</dd>
**
```

```
** [[SQLITE_CONFIG_MULTITHREAD]] <dt>SQLITE_CONFIG_MULTITHREAD</dt>
** <dd>There are no arguments to this option.  ^This option sets the
** [threading mode] to Multi-thread.  In other words, it disables
** mutexing on [database connection] and [prepared statement] objects.
** The application is responsible for serializing access to
** [database connections] and [prepared statements].  But other mutexes
** are enabled so that SQLite will be safe to use in a multi-threaded
** environment as long as no two threads attempt to use the same
** [database connection] at the same time.  ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option
 then
** it is not possible to set the Multi-thread [threading mode] and
** [sqlite3_config()] will return [SQLITE_ERROR] if called with the
** SQLITE_CONFIG_MULTITHREAD configuration option.</dd>
**
** [[SQLITE_CONFIG_SERIALIZED]] <dt>SQLITE_CONFIG_SERIALIZED</dt>
** <dd>There are no arguments to this option.  ^This option sets the
** [threading mode] to Serialized. In other words, this option enables
** all mutexes including the recursive
** mutexes on [database connection] and [prepared statement] objects.
** In this mode (which is the default when SQLite is compiled with
** [SQLITE_THREADSAFE=1]) the SQLite library will itself serialize access
** to [database connections] and [prepared statements] so that the
** application is free to use the same [database connection] or the
** same [prepared statement] in different threads at the same time.
** ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** it is not possible to set the
 Serialized [threading mode] and
** [sqlite3_config()] will return [SQLITE_ERROR] if called with the
** SQLITE_CONFIG_SERIALIZED configuration option.</dd>
**
** [[SQLITE_CONFIG_MALLOC]] <dt>SQLITE_CONFIG_MALLOC</dt>
** <dd> ^(The SQLITE_CONFIG_MALLOC option takes a single argument which is
** a pointer to an instance of the [sqlite3_mem_methods] structure.
** The argument specifies
** alternative low-level memory allocation routines to be used in place of
** the memory allocation routines built into SQLite.)^ ^SQLite makes
** its own private copy of the content of the [sqlite3_mem_methods] structure
** before the [sqlite3_config()] call returns.</dd>
**
** [[SQLITE_CONFIG_GETMALLOC]] <dt>SQLITE_CONFIG_GETMALLOC</dt>
** <dd> ^(The SQLITE_CONFIG_GETMALLOC option takes a single argument which
** is a pointer to an instance of the [sqlite3_mem_methods] structure.
** The [sqlite3_mem_methods]
** structure is filled with the currently defined memory allocation routines.)^
** This option can
```

be used to overload the default memory allocation
** routines with a wrapper that simulations memory allocation failure or
** tracks memory usage, for example. </dd>
**
** [[SQLITE_CONFIG_SMALL_MALLOC]] <dt>SQLITE_CONFIG_SMALL_MALLOC</dt>
** <dd> ^The SQLITE_CONFIG_SMALL_MALLOC option takes single argument of
** type int, interpreted as a boolean, which if true provides a hint to
** SQLite that it should avoid large memory allocations if possible.
** SQLite will run faster if it is free to make large memory allocations,
** but some application might prefer to run slower in exchange for
** guarantees about memory fragmentation that are possible if large
** allocations are avoided. This hint is normally off.
** </dd>
**
** [[SQLITE_CONFIG_MEMSTATUS]] <dt>SQLITE_CONFIG_MEMSTATUS</dt>
** <dd> ^The SQLITE_CONFIG_MEMSTATUS option takes single argument of type int,
** interpreted as a boolean, which enables or disables the collection of
** memory allocation statistics. ^(When memory allocation
statistics are
** disabled, the following SQLite interfaces become non-operational:
**
**   [sqlite3_hard_heap_limit64()]
**   [sqlite3_memory_used()]
**   [sqlite3_memory_highwater()]
**   [sqlite3_soft_heap_limit64()]
**   [sqlite3_status64()]
**   )^
** ^Memory allocation statistics are enabled by default unless SQLite is
** compiled with [SQLITE_DEFAULT_MEMSTATUS]=0 in which case memory
** allocation statistics are disabled by default.
** </dd>
**
** [[SQLITE_CONFIG_SCRATCH]] <dt>SQLITE_CONFIG_SCRATCH</dt>
** <dd> The SQLITE_CONFIG_SCRATCH option is no longer used.
** </dd>
**
** [[SQLITE_CONFIG_PAGECACHE]] <dt>SQLITE_CONFIG_PAGECACHE</dt>
** <dd> ^The SQLITE_CONFIG_PAGECACHE option specifies a memory pool
** that SQLite can use for the database page cache with the default page
** cache implementation.
** This configuration option is a no-op if an application-defined page
** cache implementation is loaded using the [SQLITE_CONFIG_PCACHE2].
**
^There are three arguments to SQLITE_CONFIG_PAGECACHE: A pointer to
** 8-byte aligned memory (pMem), the size of each page cache line (sz),
** and the number of cache lines (N).
** The sz argument should be the size of the largest database page
** (a power of two between 512 and 65536) plus some extra bytes for each

** page header. ^The number of extra bytes needed by the page header
** can be determined using [SQLITE_CONFIG_PCACHE_HDRSZ].
** ^It is harmless, apart from the wasted memory,
** for the sz parameter to be larger than necessary. The pMem
** argument must be either a NULL pointer or a pointer to an 8-byte
** aligned block of memory of at least sz*N bytes, otherwise
** subsequent behavior is undefined.
** ^When pMem is not NULL, SQLite will strive to use the memory provided
** to satisfy page cache needs, falling back to [sqlite3_malloc()] if
** a page cache line is larger than sz bytes or if all of the pMem buffer
** is exhausted.
** ^If pMem is NULL and N is non-zero, then
 each database connection
** does an initial bulk allocation for page cache memory
** from [sqlite3_malloc()] sufficient for N cache lines if N is positive or
** of -1024*N bytes if N is negative, . ^If additional
** page cache memory is needed beyond what is provided by the initial
** allocation, then SQLite goes to [sqlite3_malloc()] separately for each
** additional cache line. </dd>
**
** [[SQLITE_CONFIG_HEAP]] <dt>SQLITE_CONFIG_HEAP</dt>
** <dd> ^The SQLITE_CONFIG_HEAP option specifies a static memory buffer
** that SQLite will use for all of its dynamic memory allocation needs
** beyond those provided for by [SQLITE_CONFIG_PAGECACHE].
** ^The SQLITE_CONFIG_HEAP option is only available if SQLite is compiled
** with either [SQLITE_ENABLE_MEMSYS3] or [SQLITE_ENABLE_MEMSYS5] and returns
** [SQLITE_ERROR] if invoked otherwise.
** ^There are three arguments to SQLITE_CONFIG_HEAP:
** An 8-byte aligned pointer to the memory,
** the number of bytes in the memory buffer, and the minimum
 allocation size.
** ^If the first pointer (the memory pointer) is NULL, then SQLite reverts
** to using its default memory allocator (the system malloc() implementation),
** undoing any prior invocation of [SQLITE_CONFIG_MALLOC]. ^If the
** memory pointer is not NULL then the alternative memory
** allocator is engaged to handle all of SQLites memory allocation needs.
** The first pointer (the memory pointer) must be aligned to an 8-byte
** boundary or subsequent behavior of SQLite will be undefined.
** The minimum allocation size is capped at 2**12. Reasonable values
** for the minimum allocation size are 2**5 through 2**8.</dd>
**
** [[SQLITE_CONFIG_MUTEX]] <dt>SQLITE_CONFIG_MUTEX</dt>
** <dd> ^(The SQLITE_CONFIG_MUTEX option takes a single argument which is a
** pointer to an instance of the [sqlite3_mutex_methods] structure.
** The argument specifies alternative low-level mutex routines to be used
** in place the mutex routines built into SQLite.)^ ^SQLite makes a copy of
** the

content of the [sqlite3_mutex_methods] structure before the call to
** [sqlite3_config()] returns. ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** the entire mutexing subsystem is omitted from the build and hence calls to
** [sqlite3_config()] with the SQLITE_CONFIG_MUTEX configuration option will
** return [SQLITE_ERROR].</dd>
**
** [[SQLITE_CONFIG_GETMUTEX]] <dt>SQLITE_CONFIG_GETMUTEX</dt>
** <dd> ^(The SQLITE_CONFIG_GETMUTEX option takes a single argument which
** is a pointer to an instance of the [sqlite3_mutex_methods] structure. The
** [sqlite3_mutex_methods]
** structure is filled with the currently defined mutex routines.)^
** This option can be used to overload the default mutex allocation
** routines with a wrapper used to track mutex usage for performance
** profiling or testing, for example. ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** the entire mutexing
subsystem is omitted from the build and hence calls to
** [sqlite3_config()] with the SQLITE_CONFIG_GETMUTEX configuration option will
** return [SQLITE_ERROR].</dd>
**
** [[SQLITE_CONFIG_LOOKASIDE]] <dt>SQLITE_CONFIG_LOOKASIDE</dt>
** <dd> ^(The SQLITE_CONFIG_LOOKASIDE option takes two arguments that determine
** the default size of lookaside memory on each [database connection].
** The first argument is the
** size of each lookaside buffer slot and the second is the number of
** slots allocated to each database connection.)^ ^(SQLITE_CONFIG_LOOKASIDE
** sets the default lookaside size. The [SQLITE_DBCONFIG_LOOKASIDE]
** option to [sqlite3_db_config()] can be used to change the lookaside
** configuration on individual connections.)^ </dd>
**
** [[SQLITE_CONFIG_PCACHE2]] <dt>SQLITE_CONFIG_PCACHE2</dt>
** <dd> ^(The SQLITE_CONFIG_PCACHE2 option takes a single argument which is
** a pointer to an [sqlite3_pcache_methods2] object. This object specifies
** the interface to a custom
page cache implementation.)^
** ^SQLite makes a copy of the [sqlite3_pcache_methods2] object.</dd>
**
** [[SQLITE_CONFIG_GETPCACHE2]] <dt>SQLITE_CONFIG_GETPCACHE2</dt>
** <dd> ^(The SQLITE_CONFIG_GETPCACHE2 option takes a single argument which
** is a pointer to an [sqlite3_pcache_methods2] object. SQLite copies of
** the current page cache implementation into that object.)^ </dd>
**
** [[SQLITE_CONFIG_LOG]] <dt>SQLITE_CONFIG_LOG</dt>
** <dd> The SQLITE_CONFIG_LOG option is used to configure the SQLite
** global [error log].
** (^The SQLITE_CONFIG_LOG option takes two arguments: a pointer to a
** function with a call signature of void(*)(void*,int,const char*),

** and a pointer to void. ^If the function pointer is not NULL, it is
** invoked by [sqlite3_log()] to process each logging event. ^If the
** function pointer is NULL, the [sqlite3_log()] interface becomes a no-op.
** ^The void pointer that is the second argument to SQLITE_CONFIG_LOG is
** passed through as the first parameter
 to the application-defined logger
** function whenever that function is invoked. ^The second parameter to
** the logger function is a copy of the first parameter to the corresponding
** [sqlite3_log()] call and is intended to be a [result code] or an
** [extended result code]. ^The third parameter passed to the logger is
** log message after formatting via [sqlite3_snprintf()].
** The SQLite logging interface is not reentrant; the logger function
** supplied by the application must not invoke any SQLite interface.
** In a multi-threaded application, the application-defined logger
** function must be threadsafe. </dd>
**
** [[SQLITE_CONFIG_URI]] <dt>SQLITE_CONFIG_URI
** <dd>^(The SQLITE_CONFIG_URI option takes a single argument of type int.
** If non-zero, then URI handling is globally enabled. If the parameter is zero,
** then URI handling is globally disabled.)^ ^If URI handling is globally
** enabled, all filenames passed to [sqlite3_open()], [sqlite3_open_v2()],
** [sqlite3_open16()]
 or
** specified as part of [ATTACH] commands are interpreted as URIs, regardless
** of whether or not the [SQLITE_OPEN_URI] flag is set when the database
** connection is opened. ^If it is globally disabled, filenames are
** only interpreted as URIs if the SQLITE_OPEN_URI flag is set when the
** database connection is opened. ^(By default, URI handling is globally
** disabled. The default value may be changed by compiling with the
** [SQLITE_USE_URI] symbol defined.)^
**
** [[SQLITE_CONFIG_COVERING_INDEX_SCAN]] <dt>SQLITE_CONFIG_COVERING_INDEX_SCAN
** <dd>^The SQLITE_CONFIG_COVERING_INDEX_SCAN option takes a single integer
** argument which is interpreted as a boolean in order to enable or disable
** the use of covering indices for full table scans in the query optimizer.
** ^The default setting is determined
** by the [SQLITE_ALLOW_COVERING_INDEX_SCAN] compile-time option, or is "on"
** if that compile-time option is omitted.
** The ability to disable the use of covering indices
 for full table scans
** is because some incorrectly coded legacy applications might malfunction
** when the optimization is enabled. Providing the ability to
** disable the optimization allows the older, buggy application code to work
** without change even with newer versions of SQLite.
**
** [[SQLITE_CONFIG_PCACHE]] [[SQLITE_CONFIG_GETPCACHE]]
** <dt>SQLITE_CONFIG_PCACHE and SQLITE_CONFIG_GETPCACHE
** <dd> These options are obsolete and should not be used by new code.

```
** They are retained for backwards compatibility but are now no-ops.
** </dd>
**
** [[SQLITE_CONFIG_SQLLOG]]
** <dt>SQLITE_CONFIG_SQLLOG
** <dd>This option is only available if sqlite is compiled with the
** [SQLITE_ENABLE_SQLLOG] pre-processor macro defined. The first argument should
** be a pointer to a function of type void(*)(void*,sqlite3*,const char*, int).
** The second should be of type (void*). The callback is invoked by the library
** in three separate circumstances, identified by the value passed as the
**
 fourth parameter. If the fourth parameter is 0, then the database connection
** passed as the second argument has just been opened. The third argument
** points to a buffer containing the name of the main database file. If the
** fourth parameter is 1, then the SQL statement that the third parameter
** points to has just been executed. Or, if the fourth parameter is 2, then
** the connection being passed as the second parameter is being closed. The
** third parameter is passed NULL In this case.  An example of using this
** configuration option can be seen in the "test_sqllog.c" source file in
** the canonical SQLite source tree.</dd>
**
** [[SQLITE_CONFIG_MMAP_SIZE]]
** <dt>SQLITE_CONFIG_MMAP_SIZE
** <dd>^SQLITE_CONFIG_MMAP_SIZE takes two 64-bit integer (sqlite3_int64) values
** that are the default mmap size limit (the default setting for
** [PRAGMA mmap_size]) and the maximum allowed mmap size limit.
** ^The default setting can be overridden by each database connection using
** either
 the [PRAGMA mmap_size] command, or by using the
** [SQLITE_FCNTL_MMAP_SIZE] file control.  ^(The maximum allowed mmap size
** will be silently truncated if necessary so that it does not exceed the
** compile-time maximum mmap size set by the
** [SQLITE_MAX_MMAP_SIZE] compile-time option.)^
** ^If either argument to this option is negative, then that argument is
** changed to its compile-time default.
**
** [[SQLITE_CONFIG_WIN32_HEAPSIZE]]
** <dt>SQLITE_CONFIG_WIN32_HEAPSIZE
** <dd>^The SQLITE_CONFIG_WIN32_HEAPSIZE option is only available if SQLite is
** compiled for Windows with the [SQLITE_WIN32_MALLOC] pre-processor macro
** defined. ^SQLITE_CONFIG_WIN32_HEAPSIZE takes a 32-bit unsigned integer value
** that specifies the maximum size of the created heap.
**
** [[SQLITE_CONFIG_PCACHE_HDRSZ]]
** <dt>SQLITE_CONFIG_PCACHE_HDRSZ
** <dd>^The SQLITE_CONFIG_PCACHE_HDRSZ option takes a single parameter which
** is a pointer to an integer and writes into that integer the number of extra
**
```

bytes per page required for each page in [SQLITE_CONFIG_PAGECACHE].
** The amount of extra space required can change depending on the compiler,
** target platform, and SQLite version.
**
** [[SQLITE_CONFIG_PMASZ]]
** <dt>SQLITE_CONFIG_PMASZ
** <dd>^The SQLITE_CONFIG_PMASZ option takes a single parameter which
** is an unsigned integer and sets the "Minimum PMA Size" for the multithreaded
** sorter to that integer. The default minimum PMA Size is set by the
** [SQLITE_SORTER_PMASZ] compile-time option. New threads are launched
** to help with sort operations when multithreaded sorting
** is enabled (using the [PRAGMA threads] command) and the amount of content
** to be sorted exceeds the page size times the minimum of the
** [PRAGMA cache_size] setting and this value.
**
** [[SQLITE_CONFIG_STMTJRNL_SPILL]]
** <dt>SQLITE_CONFIG_STMTJRNL_SPILL
** <dd>^The SQLITE_CONFIG_STMTJRNL_SPILL option takes a single parameter which
** becomes the [statement journal] spill-to-disk threshold.
**
[Statement journals] are held in memory until their size (in bytes)
** exceeds this threshold, at which point they are written to disk.
** Or if the threshold is -1, statement journals are always held
** exclusively in memory.
** Since many statement journals never become large, setting the spill
** threshold to a value such as 64KiB can greatly reduce the amount of
** I/O required to support statement rollback.
** The default value for this setting is controlled by the
** [SQLITE_STMTJRNL_SPILL] compile-time option.
**
** [[SQLITE_CONFIG_SORTERREF_SIZE]]
** <dt>SQLITE_CONFIG_SORTERREF_SIZE
** <dd>The SQLITE_CONFIG_SORTERREF_SIZE option accepts a single parameter
** of type (int) - the new value of the sorter-reference size threshold.
** Usually, when SQLite uses an external sort to order records according
** to an ORDER BY clause, all fields required by the caller are present in the
** sorted records. However, if SQLite determines based on the declared type
** of a table column that
its values are likely to be very large - larger
** than the configured sorter-reference size threshold - then a reference
** is stored in each sorted record and the required column values loaded
** from the database as records are returned in sorted order. The default
** value for this option is to never use this optimization. Specifying a
** negative value for this option restores the default behaviour.
** This option is only available if SQLite is compiled with the
** [SQLITE_ENABLE_SORTER_REFERENCES] compile-time option.
**
** [[SQLITE_CONFIG_MEMDB_MAXSIZE]]

```
** <dt>SQLITE_CONFIG_MEMDB_MAXSIZE
** <dd>The SQLITE_CONFIG_MEMDB_MAXSIZE option accepts a single parameter
** [sqlite3_int64] parameter which is the default maximum size for an in-memory
** database created using [sqlite3_deserialize()].  This default maximum
** size can be adjusted up or down for individual databases using the
** [SQLITE_FCNTL_SIZE_LIMIT] [sqlite3_file_control|file-control].  If this
** configuration setting is
 never used, then the default maximum is determined
** by the [SQLITE_MEMDB_DEFAULT_MAXSIZE] compile-time option.  If that
** compile-time option is not set, then the default maximum is 1073741824.
** </dl>
*/
```

Found in path(s):
* /opt/cola/permits/1541729670_1675313227.6905026/0/sqlite-amalgamation-3400100-zip/sqlite-amalgamation-3400100/sqlite3.h
No license file was found, but licenses were detected in source scan.

```
/* This will be more informative in a later version. */
```

Found in path(s):
* /opt/cola/permits/1541729670_1675313227.6905026/0/sqlite-amalgamation-3400100-zip/sqlite-amalgamation-3400100/shell.c
No license file was found, but licenses were detected in source scan.

```
/*
** CAPI3REF: Configuration Options
** KEYWORDS: {configuration option}
**
** These constants are the available integer configuration options that
** can be passed as the first argument to the [sqlite3_config()] interface.
**
** New configuration options may be added in future releases of SQLite.
** Existing configuration options might be discontinued.  Applications
** should check the return code from [sqlite3_config()] to make sure that
** the call worked.  The [sqlite3_config()] interface will return a
** non-zero [error code] if a discontinued or unsupported configuration option
** is invoked.
**
** <dl>
** [[SQLITE_CONFIG_SINGLETHREAD]] <dt>SQLITE_CONFIG_SINGLETHREAD</dt>
** <dd>There are no arguments to this option.  ^This option sets the
** [threading mode] to Single-thread.  In other words, it disables
** all mutexing and puts SQLite into a mode where it can only be used
** by a single thread.   ^If SQLite
 is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** it is not possible to change the [threading mode] from its default
```

** value of Single-thread and so [sqlite3_config()] will return
** [SQLITE_ERROR] if called with the SQLITE_CONFIG_SINGLETHREAD
** configuration option.</dd>
**
** [[SQLITE_CONFIG_MULTITHREAD]] <dt>SQLITE_CONFIG_MULTITHREAD</dt>
** <dd>There are no arguments to this option. ^This option sets the
** [threading mode] to Multi-thread. In other words, it disables
** mutexing on [database connection] and [prepared statement] objects.
** The application is responsible for serializing access to
** [database connections] and [prepared statements]. But other mutexes
** are enabled so that SQLite will be safe to use in a multi-threaded
** environment as long as no two threads attempt to use the same
** [database connection] at the same time. ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option
 then
** it is not possible to set the Multi-thread [threading mode] and
** [sqlite3_config()] will return [SQLITE_ERROR] if called with the
** SQLITE_CONFIG_MULTITHREAD configuration option.</dd>
**
** [[SQLITE_CONFIG_SERIALIZED]] <dt>SQLITE_CONFIG_SERIALIZED</dt>
** <dd>There are no arguments to this option. ^This option sets the
** [threading mode] to Serialized. In other words, this option enables
** all mutexes including the recursive
** mutexes on [database connection] and [prepared statement] objects.
** In this mode (which is the default when SQLite is compiled with
** [SQLITE_THREADSAFE=1]) the SQLite library will itself serialize access
** to [database connections] and [prepared statements] so that the
** application is free to use the same [database connection] or the
** same [prepared statement] in different threads at the same time.
** ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** it is not possible to set the
 Serialized [threading mode] and
** [sqlite3_config()] will return [SQLITE_ERROR] if called with the
** SQLITE_CONFIG_SERIALIZED configuration option.</dd>
**
** [[SQLITE_CONFIG_MALLOC]] <dt>SQLITE_CONFIG_MALLOC</dt>
** <dd> ^(The SQLITE_CONFIG_MALLOC option takes a single argument which is
** a pointer to an instance of the [sqlite3_mem_methods] structure.
** The argument specifies
** alternative low-level memory allocation routines to be used in place of
** the memory allocation routines built into SQLite.)^ ^SQLite makes
** its own private copy of the content of the [sqlite3_mem_methods] structure
** before the [sqlite3_config()] call returns.</dd>
**
** [[SQLITE_CONFIG_GETMALLOC]] <dt>SQLITE_CONFIG_GETMALLOC</dt>
** <dd> ^(The SQLITE_CONFIG_GETMALLOC option takes a single argument which
** is a pointer to an instance of the [sqlite3_mem_methods] structure.

```
** The [sqlite3_mem_methods]
** structure is filled with the currently defined memory allocation routines.)^
** This option can
 be used to overload the default memory allocation
** routines with a wrapper that simulations memory allocation failure or
** tracks memory usage, for example. </dd>
**
** [[SQLITE_CONFIG_SMALL_MALLOC]] <dt>SQLITE_CONFIG_SMALL_MALLOC</dt>
** <dd> ^The SQLITE_CONFIG_SMALL_MALLOC option takes single argument of
** type int, interpreted as a boolean, which if true provides a hint to
** SQLite that it should avoid large memory allocations if possible.
** SQLite will run faster if it is free to make large memory allocations,
** but some application might prefer to run slower in exchange for
** guarantees about memory fragmentation that are possible if large
** allocations are avoided.  This hint is normally off.
** </dd>
**
** [[SQLITE_CONFIG_MEMSTATUS]] <dt>SQLITE_CONFIG_MEMSTATUS</dt>
** <dd> ^The SQLITE_CONFIG_MEMSTATUS option takes single argument of type int,
** interpreted as a boolean, which enables or disables the collection of
** memory allocation statistics. ^(When memory allocation
 statistics are
** disabled, the following SQLite interfaces become non-operational:
**   <ul>
**   <li> [sqlite3_hard_heap_limit64()]
**   <li> [sqlite3_memory_used()]
**   <li> [sqlite3_memory_highwater()]
**   <li> [sqlite3_soft_heap_limit64()]
**   <li> [sqlite3_status64()]
**   </ul>)^
** ^Memory allocation statistics are enabled by default unless SQLite is
** compiled with [SQLITE_DEFAULT_MEMSTATUS]=0 in which case memory
** allocation statistics are disabled by default.
** </dd>
**
** [[SQLITE_CONFIG_SCRATCH]] <dt>SQLITE_CONFIG_SCRATCH</dt>
** <dd> The SQLITE_CONFIG_SCRATCH option is no longer used.
** </dd>
**
** [[SQLITE_CONFIG_PAGECACHE]] <dt>SQLITE_CONFIG_PAGECACHE</dt>
** <dd> ^The SQLITE_CONFIG_PAGECACHE option specifies a memory pool
** that SQLite can use for the database page cache with the default page
** cache implementation.
** This configuration option is a no-op if an application-defined page
** cache implementation is loaded using the [SQLITE_CONFIG_PCACHE2].
**
 ^There are three arguments to SQLITE_CONFIG_PAGECACHE: A pointer to
** 8-byte aligned memory (pMem), the size of each page cache line (sz),
```

** and the number of cache lines (N).
** The sz argument should be the size of the largest database page
** (a power of two between 512 and 65536) plus some extra bytes for each
** page header. ^The number of extra bytes needed by the page header
** can be determined using [SQLITE_CONFIG_PCACHE_HDRSZ].
** ^It is harmless, apart from the wasted memory,
** for the sz parameter to be larger than necessary. The pMem
** argument must be either a NULL pointer or a pointer to an 8-byte
** aligned block of memory of at least sz*N bytes, otherwise
** subsequent behavior is undefined.
** ^When pMem is not NULL, SQLite will strive to use the memory provided
** to satisfy page cache needs, falling back to [sqlite3_malloc()] if
** a page cache line is larger than sz bytes or if all of the pMem buffer
** is exhausted.
** ^If pMem is NULL and N is non-zero, then
 each database connection
** does an initial bulk allocation for page cache memory
** from [sqlite3_malloc()] sufficient for N cache lines if N is positive or
** of -1024*N bytes if N is negative, . ^If additional
** page cache memory is needed beyond what is provided by the initial
** allocation, then SQLite goes to [sqlite3_malloc()] separately for each
** additional cache line. </dd>
**
** [[SQLITE_CONFIG_HEAP]] <dt>SQLITE_CONFIG_HEAP</dt>
** <dd> ^The SQLITE_CONFIG_HEAP option specifies a static memory buffer
** that SQLite will use for all of its dynamic memory allocation needs
** beyond those provided for by [SQLITE_CONFIG_PAGECACHE].
** ^The SQLITE_CONFIG_HEAP option is only available if SQLite is compiled
** with either [SQLITE_ENABLE_MEMSYS3] or [SQLITE_ENABLE_MEMSYS5] and returns
** [SQLITE_ERROR] if invoked otherwise.
** ^There are three arguments to SQLITE_CONFIG_HEAP:
** An 8-byte aligned pointer to the memory,
** the number of bytes in the memory buffer, and the minimum
 allocation size.
** ^If the first pointer (the memory pointer) is NULL, then SQLite reverts
** to using its default memory allocator (the system malloc() implementation),
** undoing any prior invocation of [SQLITE_CONFIG_MALLOC]. ^If the
** memory pointer is not NULL then the alternative memory
** allocator is engaged to handle all of SQLites memory allocation needs.
** The first pointer (the memory pointer) must be aligned to an 8-byte
** boundary or subsequent behavior of SQLite will be undefined.
** The minimum allocation size is capped at 2**12. Reasonable values
** for the minimum allocation size are 2**5 through 2**8.</dd>
**
** [[SQLITE_CONFIG_MUTEX]] <dt>SQLITE_CONFIG_MUTEX</dt>
** <dd> ^(The SQLITE_CONFIG_MUTEX option takes a single argument which is a
** pointer to an instance of the [sqlite3_mutex_methods] structure.
** The argument specifies alternative low-level mutex routines to be used

```
** in place the mutex routines built into SQLite.)^  ^SQLite makes a copy of
** the
 content of the [sqlite3_mutex_methods] structure before the call to
** [sqlite3_config()] returns. ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** the entire mutexing subsystem is omitted from the build and hence calls to
** [sqlite3_config()] with the SQLITE_CONFIG_MUTEX configuration option will
** return [SQLITE_ERROR].</dd>
**
** [[SQLITE_CONFIG_GETMUTEX]] <dt>SQLITE_CONFIG_GETMUTEX</dt>
** <dd> ^(The SQLITE_CONFIG_GETMUTEX option takes a single argument which
** is a pointer to an instance of the [sqlite3_mutex_methods] structure.  The
** [sqlite3_mutex_methods]
** structure is filled with the currently defined mutex routines.)^
** This option can be used to overload the default mutex allocation
** routines with a wrapper used to track mutex usage for performance
** profiling or testing, for example.   ^If SQLite is compiled with
** the [SQLITE_THREADSAFE | SQLITE_THREADSAFE=0] compile-time option then
** the entire mutexing
 subsystem is omitted from the build and hence calls to
** [sqlite3_config()] with the SQLITE_CONFIG_GETMUTEX configuration option will
** return [SQLITE_ERROR].</dd>
**
** [[SQLITE_CONFIG_LOOKASIDE]] <dt>SQLITE_CONFIG_LOOKASIDE</dt>
** <dd> ^(The SQLITE_CONFIG_LOOKASIDE option takes two arguments that determine
** the default size of lookaside memory on each [database connection].
** The first argument is the
** size of each lookaside buffer slot and the second is the number of
** slots allocated to each database connection.)^  ^(SQLITE_CONFIG_LOOKASIDE
** sets the <i>default</i> lookaside size. The [SQLITE_DBCONFIG_LOOKASIDE]
** option to [sqlite3_db_config()] can be used to change the lookaside
** configuration on individual connections.)^ </dd>
**
** [[SQLITE_CONFIG_PCACHE2]] <dt>SQLITE_CONFIG_PCACHE2</dt>
** <dd> ^(The SQLITE_CONFIG_PCACHE2 option takes a single argument which is
** a pointer to an [sqlite3_pcache_methods2] object.  This object specifies
** the interface to a custom
 page cache implementation.)^
** ^SQLite makes a copy of the [sqlite3_pcache_methods2] object.</dd>
**
** [[SQLITE_CONFIG_GETPCACHE2]] <dt>SQLITE_CONFIG_GETPCACHE2</dt>
** <dd> ^(The SQLITE_CONFIG_GETPCACHE2 option takes a single argument which
** is a pointer to an [sqlite3_pcache_methods2] object.  SQLite copies of
** the current page cache implementation into that object.)^ </dd>
**
** [[SQLITE_CONFIG_LOG]] <dt>SQLITE_CONFIG_LOG</dt>
** <dd> The SQLITE_CONFIG_LOG option is used to configure the SQLite
** global [error log].
```

```
** (^The SQLITE_CONFIG_LOG option takes two arguments: a pointer to a
** function with a call signature of void(*)(void*,int,const char*),
** and a pointer to void. ^If the function pointer is not NULL, it is
** invoked by [sqlite3_log()] to process each logging event.  ^If the
** function pointer is NULL, the [sqlite3_log()] interface becomes a no-op.
** ^The void pointer that is the second argument to SQLITE_CONFIG_LOG is
** passed through as the first parameter
 to the application-defined logger
** function whenever that function is invoked.  ^The second parameter to
** the logger function is a copy of the first parameter to the corresponding
** [sqlite3_log()] call and is intended to be a [result code] or an
** [extended result code].  ^The third parameter passed to the logger is
** log message after formatting via [sqlite3_snprintf()].
** The SQLite logging interface is not reentrant; the logger function
** supplied by the application must not invoke any SQLite interface.
** In a multi-threaded application, the application-defined logger
** function must be threadsafe. </dd>
**
** [[SQLITE_CONFIG_URI]] <dt>SQLITE_CONFIG_URI
** <dd>^(The SQLITE_CONFIG_URI option takes a single argument of type int.
** If non-zero, then URI handling is globally enabled. If the parameter is zero,
** then URI handling is globally disabled.)^ ^If URI handling is globally
** enabled, all filenames passed to [sqlite3_open()], [sqlite3_open_v2()],
** [sqlite3_open16()]
 or
** specified as part of [ATTACH] commands are interpreted as URIs, regardless
** of whether or not the [SQLITE_OPEN_URI] flag is set when the database
** connection is opened. ^If it is globally disabled, filenames are
** only interpreted as URIs if the SQLITE_OPEN_URI flag is set when the
** database connection is opened. ^(By default, URI handling is globally
** disabled. The default value may be changed by compiling with the
** [SQLITE_USE_URI] symbol defined.)^
**
** [[SQLITE_CONFIG_COVERING_INDEX_SCAN]] <dt>SQLITE_CONFIG_COVERING_INDEX_SCAN
** <dd>^The SQLITE_CONFIG_COVERING_INDEX_SCAN option takes a single integer
** argument which is interpreted as a boolean in order to enable or disable
** the use of covering indices for full table scans in the query optimizer.
** ^The default setting is determined
** by the [SQLITE_ALLOW_COVERING_INDEX_SCAN] compile-time option, or is "on"
** if that compile-time option is omitted.
** The ability to disable the use of covering indices
 for full table scans
** is because some incorrectly coded legacy applications might malfunction
** when the optimization is enabled.  Providing the ability to
** disable the optimization allows the older, buggy application code to work
** without change even with newer versions of SQLite.
**
** [[SQLITE_CONFIG_PCACHE]] [[SQLITE_CONFIG_GETPCACHE]]
```

** <dt>SQLITE_CONFIG_PCACHE and SQLITE_CONFIG_GETPCACHE
** <dd> These options are obsolete and should not be used by new code.
** They are retained for backwards compatibility but are now no-ops.
** </dd>
**
** [[SQLITE_CONFIG_SQLLOG]]
** <dt>SQLITE_CONFIG_SQLLOG
** <dd>This option is only available if sqlite is compiled with the
** [SQLITE_ENABLE_SQLLOG] pre-processor macro defined. The first argument should
** be a pointer to a function of type void(*)(void*,sqlite3*,const char*, int).
** The second should be of type (void*). The callback is invoked by the library
** in three separate circumstances, identified by the value passed as the
**
 fourth parameter. If the fourth parameter is 0, then the database connection
** passed as the second argument has just been opened. The third argument
** points to a buffer containing the name of the main database file. If the
** fourth parameter is 1, then the SQL statement that the third parameter
** points to has just been executed. Or, if the fourth parameter is 2, then
** the connection being passed as the second parameter is being closed. The
** third parameter is passed NULL In this case. An example of using this
** configuration option can be seen in the "test_sqllog.c" source file in
** the canonical SQLite source tree.</dd>
**
** [[SQLITE_CONFIG_MMAP_SIZE]]
** <dt>SQLITE_CONFIG_MMAP_SIZE
** <dd>^SQLITE_CONFIG_MMAP_SIZE takes two 64-bit integer (sqlite3_int64) values
** that are the default mmap size limit (the default setting for
** [PRAGMA mmap_size]) and the maximum allowed mmap size limit.
** ^The default setting can be overridden by each database connection using
** either
 the [PRAGMA mmap_size] command, or by using the
** [SQLITE_FCNTL_MMAP_SIZE] file control. ^(The maximum allowed mmap size
** will be silently truncated if necessary so that it does not exceed the
** compile-time maximum mmap size set by the
** [SQLITE_MAX_MMAP_SIZE] compile-time option.)^
** ^If either argument to this option is negative, then that argument is
** changed to its compile-time default.
**
** [[SQLITE_CONFIG_WIN32_HEAPSIZE]]
** <dt>SQLITE_CONFIG_WIN32_HEAPSIZE
** <dd>^The SQLITE_CONFIG_WIN32_HEAPSIZE option is only available if SQLite is
** compiled for Windows with the [SQLITE_WIN32_MALLOC] pre-processor macro
** defined. ^SQLITE_CONFIG_WIN32_HEAPSIZE takes a 32-bit unsigned integer value
** that specifies the maximum size of the created heap.
**
** [[SQLITE_CONFIG_PCACHE_HDRSZ]]
** <dt>SQLITE_CONFIG_PCACHE_HDRSZ
** <dd>^The SQLITE_CONFIG_PCACHE_HDRSZ option takes a single parameter which

```
** is a pointer to an integer and writes into that integer the number of extra
**
 bytes per page required for each page in [SQLITE_CONFIG_PAGECACHE].
** The amount of extra space required can change depending on the compiler,
** target platform, and SQLite version.
**
** [[SQLITE_CONFIG_PMASZ]]
** <dt>SQLITE_CONFIG_PMASZ
** <dd>^The SQLITE_CONFIG_PMASZ option takes a single parameter which
** is an unsigned integer and sets the "Minimum PMA Size" for the multithreaded
** sorter to that integer.  The default minimum PMA Size is set by the
** [SQLITE_SORTER_PMASZ] compile-time option.  New threads are launched
** to help with sort operations when multithreaded sorting
** is enabled (using the [PRAGMA threads] command) and the amount of content
** to be sorted exceeds the page size times the minimum of the
** [PRAGMA cache_size] setting and this value.
**
** [[SQLITE_CONFIG_STMTJRNL_SPILL]]
** <dt>SQLITE_CONFIG_STMTJRNL_SPILL
** <dd>^The SQLITE_CONFIG_STMTJRNL_SPILL option takes a single parameter which
** becomes the [statement journal] spill-to-disk threshold.
**
 [Statement journals] are held in memory until their size (in bytes)
** exceeds this threshold, at which point they are written to disk.
** Or if the threshold is -1, statement journals are always held
** exclusively in memory.
** Since many statement journals never become large, setting the spill
** threshold to a value such as 64KiB can greatly reduce the amount of
** I/O required to support statement rollback.
** The default value for this setting is controlled by the
** [SQLITE_STMTJRNL_SPILL] compile-time option.
**
** [[SQLITE_CONFIG_SORTERREF_SIZE]]
** <dt>SQLITE_CONFIG_SORTERREF_SIZE
** <dd>The SQLITE_CONFIG_SORTERREF_SIZE option accepts a single parameter
** of type (int) - the new value of the sorter-reference size threshold.
** Usually, when SQLite uses an external sort to order records according
** to an ORDER BY clause, all fields required by the caller are present in the
** sorted records. However, if SQLite determines based on the declared type
** of a table column that
 its values are likely to be very large - larger
** than the configured sorter-reference size threshold - then a reference
** is stored in each sorted record and the required column values loaded
** from the database as records are returned in sorted order. The default
** value for this option is to never use this optimization. Specifying a
** negative value for this option restores the default behaviour.
** This option is only available if SQLite is compiled with the
** [SQLITE_ENABLE_SORTER_REFERENCES] compile-time option.
```

```
**
** [[SQLITE_CONFIG_MEMDB_MAXSIZE]]
** <dt>SQLITE_CONFIG_MEMDB_MAXSIZE
** <dd>The SQLITE_CONFIG_MEMDB_MAXSIZE option accepts a single parameter
** [sqlite3_int64] parameter which is the default maximum size for an in-memory
** database created using [sqlite3_deserialize()].  This default maximum
** size can be adjusted up or down for individual databases using the
** [SQLITE_FCNTL_SIZE_LIMIT] [sqlite3_file_control|file-control].  If this
** configuration setting is
 never used, then the default maximum is determined
** by the [SQLITE_MEMDB_DEFAULT_MAXSIZE] compile-time option.  If that
** compile-time option is not set, then the default maximum is 1073741824.
** </dl>
*/
/*
** The "printf" code that follows dates from the 1980's.  It is in
** the public domain.
**
**************************************************************************
**
** This file contains code for a set of "printf"-like routines.  These
** routines format strings much like the printf() from the standard C
** library, though the implementation here has enhancements to support
** SQLite.
*/
/*
** 2004 May 22
**
** The author disclaims copyright to this source code.  In place of
** a legal notice, here is a blessing:
**
**    May you do good and not evil.
**    May you find forgiveness for yourself and forgive others.
**    May you share freely, never taking more than you give.
**
**************************************************************************
**
** This file
 contains the VFS implementation for unix-like operating systems
** include Linux, MacOSX, *BSD, QNX, VxWorks, AIX, HPUX, and others.
**
** There are actually several different VFS implementations in this file.
** The differences are in the way that file locking is done.  The default
** implementation uses Posix Advisory Locks.  Alternative implementations
** use flock(), dot-files, various proprietary locking schemas, or simply
** skip locking all together.
**
** This source file is organized into divisions where the logic for various
```

** subfunctions is contained within the appropriate division. PLEASE
** KEEP THE STRUCTURE OF THIS FILE INTACT. New code should be placed
** in the correct division and should be clearly labeled.
**
** The layout of divisions is as follows:
**
**   * General-purpose declarations and utility functions.
**   * Unique file ID logic used by VxWorks.
**   * Various locking primitive implementations (all except proxy locking):
**   + for Posix Advisory
 Locks
**   + for no-op locks
**   + for dot-file locks
**   + for flock() locking
**   + for named semaphore locks (VxWorks only)
**   + for AFP filesystem locks (MacOSX only)
**   * sqlite3_file methods not associated with locking.
**   * Definitions of sqlite3_io_methods objects for all locking
**   methods plus "finder" functions for each locking method.
**   * sqlite3_vfs method implementations.
**   * Locking primitives for the proxy uber-locking-method. (MacOSX only)
**   * Definitions of sqlite3_vfs objects for all locking methods
**   plus implementations of sqlite3_os_init() and sqlite3_os_end().
*/
/*
** Return a pointer to the "temporary page" buffer held internally
** by the pager. This is a buffer that is big enough to hold the
** entire content of a database page. This buffer is used internally
** during rollback and will be overwritten whenever a rollback
** occurs. But other modules are free to use it too, as long as
** no rollbacks are happening.
*/

Found
 in path(s):
* /opt/cola/permits/1541729670_1675313227.6905026/0/sqlite-amalgamation-3400100-zip/sqlite-amalgamation-3400100/sqlite3.c

1.4 log4cplus 2.0.7

1.4.1 Available under license :

Copyright (c) 2012-2015 Jakob Progsch

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

Modified for log4cplus, copyright (c) 2014-2015 Vclav Zeman.
See the LICENSE file.
Log4cplus license
==========

Each file of log4cplus source is licensed using either two clause BSD
license or Apache license 2.0. Log4cplus is derived work from log4j.

Threadpool code is licensed under separate license.

Two clause BSD license

Copyright 1999--2009 Contributors to log4cplus project.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modifica tion, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
with the distribution.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache 2.0 license

Apache License; Version 2.0, January 2004; http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship,
including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on
behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily
infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion
in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE,
NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has
been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Log4j license
==========

Apache License; Version 2.0, January 2004; http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall
mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software
source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind
by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the
copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer
the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation,
if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission
of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work
==========

To
 apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced
with your own identifying information. (Don't include the brackets!)
The text should be enclosed in the appropriate comment syntax for the
file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the
copyright notice for easier identification within third-party
archives.

  Copyright 1999-2009 [Contributors to log4cplus project]

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied.
  See the License for the specific language governing permissions and
  limitations under the License.

Threadpool license
==========

Copyright (c) 2012-2015 Jakob Progsch

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

  1. The origin of this software must not be misrepresented; you must not
  claim that you wrote the original software. If you use this software
  in a product, an acknowledgment in the product documentation would be
  appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Modified for log4cplus, copyright (c) 2014-2015 Vclav Zeman.
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
// Module: Log4CPLUS
// File: consoleappender.cxx
// Created: 6/2001
// Author: Tad E. Smith
//
//
// Copyright 2001-2017 Tad E. Smith
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,

// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

1.5 crashpad 20210210-adf40b

1.5.1 Available under license :

// Copyright 2017 The Crashpad Authors. All rights reserved.
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//   http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.6 re2 20230602-7c5e396

1.6.1 Available under license :

// Copyright (c) 2009 The RE2 Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are

// met:
//
//   * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//   * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//   * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE
 ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.7 curl 7.88.1

1.7.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2020, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

1.8 minizip 3.0.8

1.8.1 Available under license :

Condition of use and distribution are the same as zlib:

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgement in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

1.9 npcap 1.55

1.9.1 Available under license :

NPCAP COPYRIGHT / END USER LICENSE AGREEMENT

Npcap is a Windows packet sniffing driver and library and is copyright (c) 2013-2021 by Insecure.Com LLC ("The Nmap Project"). All rights reserved.

Even though Npcap source code is publicly available for review, it is not open source software and may not be redistributed without special permission from the Nmap Project. The standard version is also limited to installation on five systems. We fund the Npcap project by selling two types of commercial licenses to a special Npcap OEM edition:

1) Npcap OEM Redistribution License allows companies to redistribute Npcap with their products.

2) Npcap OEM Internal Use License allows companies to use Npcap OEM internally in excess of the free/demo version's normal 5-system limitation.

Both of these licenses include updates and support as well as a warranty. Npcap OEM also includes a silent installer for unattended installation. Further details about Npcap OEM are available from https://nmap.org/npcap/oem/,
and you are also welcome to contact us at sales@nmap.com to ask any questions or set up a license for your organization.

Free and open source software producers are also welcome to contact us for redistribution requests. However, we normally recommend that such authors instead ask your users to download and install Npcap themselves.

If the Nmap Project (directly or through one of our commercial licensing customers) has granted you additional rights to Npcap or Npcap OEM, those additional rights take precedence where they conflict with the terms of this license agreement.

Since the Npcap source code is available for download and review, users sometimes contribute code patches to fix bugs or add new features. By sending these changes to the Nmap Project (including through direct email or our mailing lists or submitting pull requests through our source code repository), it is understood unless you specify otherwise that you are offering the Nmap Project the unlimited, non-exclusive
right to reuse, modify, and relicence your code contribution so that we may (but are not obligated to) incorporate it into Npcap. If you wish to specify special license conditions or restrictions on your contributions, just say so when you send them.

This copy of Npcap (the "Software") and accompanying documentation is licensed and not sold. This Software is protected by copyright laws and treaties, as well as laws and treaties related to other forms of intellectual property. The Nmap Project owns intellectual property rights in the Software. The Licensee's ("you" or "your") license to download, use, copy, or change the Software is subject to these rights and to all the terms and conditions of this End User License Agreement ("Agreement").

ACCEPTANCE

By accepting this agreement or by downloading, installing, using, or copying the Software, or by clicking "I Agree", you agree to be bound by the terms of this EULA. If you do not agree to the terms of this

EULA, do not
install, use, or copy the Software.

LICENSE GRANT

This Agreement entitles you to install and use five (5) copies of the Software. In addition, you may make archival copies of the Software which may only be used for the reinstallation of the Software. This Agreement does not permit the installation or use of more than 5 copies of the Software, or the installation of the Software on more than five computer at any given time, on a system that allows shared used of applications by more than five users, or on any configuration or system of computers that allows more than five users. A user may only have one instance of this Agreement active at once. For example, downloading the software multiple times, downloading multiple versions of the software, and/or executing the software installer multiple times do not grant any additional rights such as using the software on more machines.

The terms "computer" and "machine" in this license include any computing device, including software computing
instances such as
virtual machines and Docker containers.

Copies of Npcap do not count toward the five copy, five computer, or five user limitations imposed by this section if they are installed and used solely in conjunction with any of the following software:

o The Nmap Security Scanner, as distributed from https://nmap.org

o The Wireshark network protocol analyzer, as distributed from
https://www.wireshark.org/

o Microsoft Defender for Identity, as distributed from
https://www.microsoft.com/en-us/microsoft-365/security/identity-defender

Users wishing to redistribute Npcap or exceed the usage limits imposed by this free license or benefit from commercial support and features such as a silent installer should contact sales@nmap.com to obtain an appropriate commercial license agreement. More details on our OEM edition is also available from https://nmap.org/npcap/oem/.

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

This program is distributed in the hope that it
will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

RESTRICTIONS ON TRANSFER

Without first obtaining the express written consent of the Nmap Project, you may not assign your rights and obligations under this Agreement, or redistribute, encumber, sell, rent, lease, sublicense, or otherwise transfer your rights to the Software Product.

RESTRICTIONS ON USE

You may not use, copy, or install the Software Product on more than five computers, or permit the use, copying, or installation of the Software Product by more than five users or on more than five computers.

RESTRICTIONS ON COPYING

You may not copy any part of the Software except to the extent that licensed use inherently demands the creation of a temporary copy stored in computer memory and not permanently affixed on storage medium. You may make archival copies as well.

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

UNLESS OTHERWISE EXPLICITLY
AGREED TO IN WRITING BY THE NMAP PROJECT,
THE NMAP PROJECT MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, IN FACT OR IN LAW, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OTHER THAN AS SET FORTH IN THIS AGREEMENT OR IN THE LIMITED WARRANTY DOCUMENTS PROVIDED WITH THE SOFTWARE.

The Nmap Project makes no warranty that the Software will meet your requirements or operate under your specific conditions of use. The Nmap Project makes no warranty that operation of the Software Product will be secure, error free, or free from interruption. YOU MUST DETERMINE WHETHER THE SOFTWARE SUFFICIENTLY MEETS YOUR REQUIREMENTS FOR SECURITY AND UNINTERRUPTABILITY. YOU BEAR SOLE RESPONSIBILITY AND ALL LIABILITY FOR ANY LOSS INCURRED DUE TO FAILURE OF THE SOFTWARE TO MEET YOUR REQUIREMENTS. THE NMAP PROJECT WILL NOT, UNDER ANY CIRCUMSTANCES, BE RESPONSIBLE OR LIABLE FOR THE LOSS OF DATA ON ANY COMPUTER OR INFORMATION STORAGE DEVICE.

UNDER NO CIRCUMSTANCES
SHALL THE NMAP PROJECT, ITS DIRECTORS,
OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU OR ANY OTHER PARTY FOR INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOST REVENUES OR PROFITS OR LOSS OF

BUSINESS) RESULTING FROM THIS AGREEMENT, OR FROM THE FURNISHING, PERFORMANCE, INSTALLATION, OR USE OF THE SOFTWARE, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, OR THE NEGLIGENCE OF THE NMAP PROJECT OR ANY OTHER PARTY, EVEN IF THE NMAP PROJECT IS ADVISED BEFOREHAND OF THE POSSIBILITY OF SUCH DAMAGES. TO THE EXTENT THAT THE APPLICABLE JURISDICTION LIMITS THE NMAP PROJECT'S ABILITY TO DISCLAIM ANY IMPLIED WARRANTIES, THIS DISCLAIMER SHALL BE EFFECTIVE TO THE MAXIMUM EXTENT PERMITTED.

LIMITATIONS OF REMEDIES AND DAMAGES

Your remedy for a breach of this Agreement or of any warranty included in this Agreement is the correction or replacement of the Software or a refund of the purchase price of the Software, exclusive of any costs for shipping and handling. Selection of whether to correct or replace or refund shall be solely at the discretion of the Nmap Project. The Nmap Project reserves the right to substitute a functionally equivalent copy of the Software Product as a replacement.

Any claim must be made within the applicable warranty period. All warranties cover only defects arising under normal use and do not include malfunctions or failure resulting from misuse, abuse, neglect, alteration, problems with electrical power, acts of nature, unusual temperatures or humidity, improper installation, or damage determined by the Nmap Project to have been caused by you. All limited warranties on the Software Product are granted only to you and are non-transferable.

You agree to indemnify and hold the Nmap Project harmless from all claims, judgments, liabilities, expenses, or costs arising from your breach of this Agreement and/or acts or omissions.

GOVERNING LAW, JURISDICTION AND COSTS

This Agreement is governed by the laws the United States of America and Washington State, without regard to Washington's conflict or choice of law provisions.

SEVERABILITY

If any provision of this Agreement shall be held to be invalid or unenforceable, the remainder of this Agreement shall remain in full force and effect. To the extent any express or implied restrictions are not permitted by applicable laws, these express or implied restrictions shall remain in force and effect to the maximum extent permitted by such applicable laws.

THIRD PARTY SOFTWARE ATTRIBUTION

Npcap uses several 3rd party open source software libraries:

* The libpcap portable packet capturing library from https://tcpdump.org
* The Winpcap packet capturing library. It has been abandoned, but is
currently still available from https://www.winpcap.org/.
* The ieee80211_radiotap.h header file from David Young

All of these are open source with BSD-style licenses that allow for
unlimited use and royalty-free redistribution within other
software
(including commercial/proprietary software). Some include a warranty
disclaimer (relating to the original authors) and require a small
amount of acknowledgment text be added somewhere in the documentation
of any software which includes them (including indirect inclusion
through Npcap).

The required acknowledgement text as well as full license text and
source details for these libraries is available from:
https://npcap.org/src/docs/Npcap-Third-Party-Open-Source.pdf .

Since Insecure.Com LLC is not the author of this 3rd party code, we
can not waive or modify its software copyright or license. Npcap
users and redistributors must comply with the relevant Npcap license
(either the free/demo license or a commercial Npcap OEM license they
may have purchased) as well as the minimal requirements of this 3rd
party open source software.

1.10 zlib 1.2.13

1.10.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software

in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

1.11 openssl 3.0.12.8.0.83

1.11.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HTTP SERVER SUBCOMPONENTS:

The Apache HTTP Server includes a number of subcomponents with
separate copyright notices and license terms. Your use of the source
code for the these subcomponents is subject to the terms and
conditions of the following licenses.

For the mod_mime_magic component:

/*
* mod_mime_magic: MIME type lookup via file magic numbers
* Copyright (c) 1996-1997 Cisco Systems, Inc.
*
* This software was submitted by Cisco Systems to the Apache Group in July

* 1997. Future revisions and derivatives of this source code must
* acknowledge Cisco Systems as the original contributor of this module.
* All other
 licensing and usage conditions are those of the Apache Group.
*
* Some of this code is derived from the free version of the file command
* originally posted to comp.sources.unix. Copyright info for that program
* is included below as required.
* ---
* - Copyright (c) Ian F. Darwin, 1987. Written by Ian F. Darwin.
*
* This software is not subject to any license of the American Telephone and
* Telegraph Company or of the Regents of the University of California.
*
* Permission is granted to anyone to use this software for any purpose on any
* computer system, and to alter it and redistribute it freely, subject to
* the following restrictions:
*
* 1. The author is not responsible for the consequences of use of this
* software, no matter how awful, even if they arise from flaws in it.
*
* 2. The origin of this software must not be misrepresented, either by
* explicit claim or by omission. Since
 few users ever read sources, credits
* must appear in the documentation.
*
* 3. Altered versions must be plainly marked as such, and must not be
* misrepresented as being the original software. Since few users ever read
* sources, credits must appear in the documentation.
*
* 4. This notice may not be removed or altered.
* ---
*
*/

For the modules\mappers\mod_imagemap.c component:

 "macmartinized" polygon code copyright 1992 by Eric Haines, erich@eye.com

For the server\util_md5.c component:

/***
* NCSA HTTPd Server
* Software Development Group
* National Center for Supercomputing Applications
* University of Illinois at Urbana-Champaign
* 605 E. Springfield, Champaign, IL 61820

```
* httpd@ncsa.uiuc.edu
*
* Copyright  (C)  1995, Board of Trustees of the University of Illinois
*
************************************************************************
 *
* md5.c: NCSA HTTPd code which uses the md5c.c RSA Code
*
*  Original Code Copyright (C) 1994, Jeff Hostetler, Spyglass, Inc.
*  Portions of Content-MD5 code Copyright (C) 1993, 1994 by Carnegie Mellon
*     University (see Copyright below).
*  Portions of Content-MD5 code Copyright (C) 1991 Bell Communications
*     Research, Inc. (Bellcore) (see Copyright below).
*  Portions extracted from mpack, John G. Myers - jgm+@cmu.edu
*  Content-MD5 Code contributed by Martin Hamilton (martin@net.lut.ac.uk)
*
*/


/* these portions extracted from mpack, John G. Myers - jgm+@cmu.edu */
/* (C) Copyright 1993,1994 by Carnegie Mellon University
* All Rights Reserved.
*
* Permission to use, copy, modify, distribute, and sell this software
* and its documentation for any purpose is hereby granted without
* fee, provided that the above copyright notice appear in all copies
* and that both that copyright notice and this permission notice
* appear in supporting
 documentation, and that the name of Carnegie
* Mellon University not be used in advertising or publicity
* pertaining to distribution of the software without specific,
* written prior permission.  Carnegie Mellon University makes no
* representations about the suitability of this software for any
* purpose.  It is provided "as is" without express or implied
* warranty.
*
* CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO
* THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
* AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE
* FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
* AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
* OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
* SOFTWARE.
*/

/*
* Copyright (c) 1991 Bell Communications Research, Inc. (Bellcore)
```

*
* Permission to use, copy, modify,
 and distribute this material
* for any purpose and without fee is hereby granted, provided
* that the above copyright notice and this permission notice
* appear in all copies, and that the name of Bellcore not be
* used in advertising or publicity pertaining to this
* material without the specific, prior written permission
* of an authorized representative of Bellcore. BELLCORE
* MAKES NO REPRESENTATIONS ABOUT THE ACCURACY OR SUITABILITY
* OF THIS MATERIAL FOR ANY PURPOSE. IT IS PROVIDED "AS IS",
* WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES.
*/

For the srclib\apr\include\apr_md5.h component:
/*
* This is work is derived from material Copyright RSA Data Security, Inc.
*
* The RSA copyright statement and Licence for that original material is
* included below. This is followed by the Apache copyright statement and
* licence for the modifications made to that material.
*/

/* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
  rights reserved.

  License
 to copy and use this software is granted provided that it
  is identified as the "RSA Data Security, Inc. MD5 Message-Digest
  Algorithm" in all material mentioning or referencing this software
  or this function.

  License is also granted to make and use derivative works provided
  that such works are identified as "derived from the RSA Data
  Security, Inc. MD5 Message-Digest Algorithm" in all material
  mentioning or referencing the derived work.

  RSA Data Security, Inc. makes no representations concerning either
  the merchantability of this software or the suitability of this
  software for any particular purpose. It is provided "as is"
  without express or implied warranty of any kind.

  These notices must be retained in any copies of any part of this
  documentation and/or software.
*/

For the srclib\apr\passwd\apr_md5.c component:

/*
* This is work is derived from material Copyright RSA Data Security, Inc.
*
* The RSA copyright statement and Licence
 for that original material is
* included below. This is followed by the Apache copyright statement and
* licence for the modifications made to that material.
*/

/* MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm
*/

/* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
  rights reserved.

  License to copy and use this software is granted provided that it
  is identified as the "RSA Data Security, Inc. MD5 Message-Digest
  Algorithm" in all material mentioning or referencing this software
  or this function.

  License is also granted to make and use derivative works provided
  that such works are identified as "derived from the RSA Data
  Security, Inc. MD5 Message-Digest Algorithm" in all material
  mentioning or referencing the derived work.

  RSA Data Security, Inc. makes no representations concerning either
  the merchantability of this software or the suitability of this
  software for any particular purpose. It is provided "as is"

  without express or implied warranty of any kind.

  These notices must be retained in any copies of any part of this
  documentation and/or software.
*/
/*
* The apr_md5_encode() routine uses much code obtained from the FreeBSD 3.0
* MD5 crypt() function, which is licenced as follows:
* --
* "THE BEER-WARE LICENSE" (Revision 42):
* <phk@login.dknet.dk> wrote this file. As long as you retain this notice you
* can do whatever you want with this stuff. If we meet some day, and you think
* this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp
* --
*/

For the srclib\apr-util\crypto\apr_md4.c component:

* This is derived from material copyright RSA Data Security, Inc.

* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
* rights reserved.
 *
* License to copy and use this software is granted provided that it
* is identified as the "RSA Data Security, Inc. MD4 Message-Digest
* Algorithm" in all material mentioning or referencing this software
* or this function.
*
* License is also granted to make and use derivative works provided
* that such works are identified as "derived from the RSA Data
* Security, Inc. MD4 Message-Digest Algorithm" in all material
* mentioning or referencing the derived work.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/

For the srclib\apr-util\include\apr_md4.h component:

*
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below
 in its entirety.
*
* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
* rights reserved.
*
* License to copy and use this software is granted provided that it
* is identified as the "RSA Data Security, Inc. MD4 Message-Digest
* Algorithm" in all material mentioning or referencing this software
* or this function.
*
* License is also granted to make and use derivative works provided
* that such works are identified as "derived from the RSA Data
* Security, Inc. MD4 Message-Digest Algorithm" in all material
* mentioning or referencing the derived work.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.

```
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/
```

For the srclib\apr-util\test\testmd4.c component:

```
 *
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1990-2, RSA Data Security, Inc. Created 1990. All
* rights reserved.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/
```

For the srclib\apr-util\xml\expat\conftools\install-sh component:

```
#
# install - install a program, script, or datafile
# This comes from X11R5 (mit/util/scripts/install.sh).
#
# Copyright 1991 by the Massachusetts Institute of Technology
#
# Permission to use, copy, modify, distribute, and sell this software and its
# documentation for any purpose is hereby granted without fee, provided that
# the above copyright
 notice appear in all copies and that both that
# copyright notice and this permission notice appear in supporting
# documentation, and that the name of M.I.T. not be used in advertising or
# publicity pertaining to distribution of the software without specific,
# written prior permission.  M.I.T. makes no representations about the
# suitability of this software for any purpose.  It is provided "as is"
# without express or implied warranty.
#
```

For the test\zb.c component:

```
/*                    ZeusBench V1.01
     ===============
```

This program is Copyright (C) Zeus Technology Limited 1996.

This program may be used and copied freely providing this copyright notice
is not removed.

This software is provided "as is" and any express or implied waranties,
including but not limited to, the implied warranties of merchantability and
fitness for a particular purpose are disclaimed. In no event shall
Zeus Technology Ltd. be liable for any direct, indirect, incidental, special,
exemplary,
or consequential damaged (including, but not limited to,
procurement of substitute good or services; loss of use, data, or profits;
or business interruption) however caused and on theory of liability. Whether
in contract, strict liability or tort (including negligence or otherwise)
arising in any way out of the use of this software, even if advised of the
possibility of such damage.

Written by Adam Twiss (adam@zeus.co.uk). March 1996

Thanks to the following people for their input:
Mike Belshe (mbelshe@netscape.com)
Michael Campanella (campanella@stevms.enet.dec.com)

*/

For the expat xml parser component:

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd
and Clark Cooper

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy,
modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==========

1.12 libpcap 1.13

1.12.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 1994, 1996
* The Regents of the University of California.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the Laboratory may be
 used
*    to endorse or promote products derived from this software without
*    specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
```

Found in path(s):

* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/namedb.h

*
/opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap-namedb.h
No license file was found, but licenses were detected in source scan.

```
/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
*      This product includes software developed by the University of
*      California, Berkeley and its contributors.
* 4. Neither the name of the University
 nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/vlan.h
No license file was found, but licenses were detected in source scan.

<html><head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><title>Npcap Reference Guide</title><meta name="generator" content="DocBook XSL Stylesheets V1.79.2"><meta name="description" content="A manual and guide to Npcap, a packet capture and network analysis framework for Windows, for users and software developers. Npcap is a modern, safe, and compatible update to WinPcap."><link rel="home" href="index.html" title="Npcap Reference Guide"><link rel="next" href="npcap-users-guide.html" title="Npcap Users' Guide"></head><body bgcolor="white" text="black" link="#0000FF" vlink="#840084"

alink="#0000FF"> <table width="100%" summary="Navigation header"><tr><th colspan="3" align="center">Npcap Reference Guide</th></tr><tr><td width="20%" align="left"></td><th width="60%" align="center"></th><td width="20%" align="right">Next</td></tr></table><hr> <h2 class="title">Npcap Reference Guide</h2> Abstract
A manual and guide to Npcap, a packet capture and network analysis framework for Windows, for users and software developers. Npcap is a modern, safe, and compatible update to WinPcap.
<hr> Table of Contents <dl class="toc"><dt>Introduction</dt><dd><dl><dt>What is Npcap?</dt><dt>Npcap Features</dt><dt>Purpose of this manual</dt><dt>Terminology</dt><dt>Npcap License</dt><dt>Obtaining Npcap</dt><dt>Acknowledgements and copyright</dt></dl></dd><dt>Npcap Users' Guide</dt><dd><dl><dt>Installation</dt><dt>Windows platforms supported</dt><dt>How to use Wireshark to capture raw 802.11 traffic in “Monitor Mode”</dt><dt>Q & A</dt><dt>Reporting Bugs</dt></dl></dd><dt>Developing software with Npcap</dt><dd><dl><dt>Using the Npcap SDK</dt><dt>Examples</dt><dt>Updating WinPcap software to Npcap</dt><dt>How to detect what version Npcap/WinPcap you are using?</dt><dt>For software that want to use Npcap first when Npcap and WinPcap coexist</dt><dt>For software that uses Npcap loopback feature</dt><dt>For software that uses Npcap raw 802.11 feature</dt></dl></dd><dt>The Npcap API</dt><dd><dl><dt>Extensions to libpcap for Windows</dt></dl></dd><dt>Npcap Development Tutorial</dt><dd><dl><dt>Obtaining the device list</dt><dt>Obtaining advanced information about installed devices</dt><dt>Opening an adapter and capturing the packets</dt><dt>Capturing the packets without the callback</dt><dt>Filtering the

traffic</dt><dt>Interpreting the packets</dt><dt>Handling offline dump files</dt><dt>Sending Packets</dt><dt>Gathering Statistics on the network traffic</dt></dl></dd><dt>Npcap internals</dt><dd><dl><dt>Npcap structure</dt><dt>Npcap driver internals</dt><dt>Further reading</dt></dl></dd></dl>

<h2
class="title" style="clear: both">Introduction</h2>

This Manual describes the programming interface and the source code of
Npcap. It provides detailed descriptions of the functions and structures
exported to programmers, along with complete documentation of the Npcap
internals. Several tutorials and examples are provided as well.

<h3 class="title">What is Npcap?</h3>

Npcap is an architecture for packet capture and network analysis for
Windows operating systems, consisting of a software library and a network
driver.

Most networking applications access the network through widely-used
operating system primitives such as sockets. It is easy to access data on
the network with this approach since the operating system copes with the
low level
details (protocol handling, packet reassembly, etc.) and
provides a familiar interface that is similar to the one used to read and
write files.

Sometimes, however, the “easy way” is not up to the task,
since some applications require direct access to packets on the network.
That is, they need access to the “raw” data on the network
without the interposition of protocol processing by the operating
system.

The purpose of Npcap is to give this kind of access to Windows
applications. It provides facilities to:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">capture raw packets, both the ones destined to the machine where
it's running and the ones exchanged by other hosts (on shared media)
<li class="listitem">filter the packets according to user-specified rules before
dispatching them to the application
<li class="listitem">transmit raw packets to the network
<li class="listitem">gather statistical information on the network traffic

This set of capabilities is obtained by means of a device driver,
which is installed inside the networking portion of the Windows kernel,
plus a couple of DLLs.

All of these features are exported through a powerful programming
interface, easily usable by applications. The main goal of this manual is
to document this interface, with the help of several examples.

<h4 class="title">What kind of programs use Npcap?</h4>

The Npcap programming interface can be used by many types of
network tools for analysis, troubleshooting, security and monitoring.
In particular,
classical tools that rely on Npcap are:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">network and protocol analyzers
<li class="listitem">network monitors
<li class="listitem">traffic loggers
<li class="listitem">traffic generators
<li class="listitem">user-level bridges and routers
<li class="listitem">network intrusion detection systems (NIDS)
<li class="listitem">network scanners
<li class="listitem">security tools

<h4 class="title">What Npcap can't do</h4>

Npcap receives and sends the packets independently from the host
protocols, like TCP/IP. This means that it isn't able to block, filter or
manipulate the traffic generated by other programs on the same machine: it
simply “sniffs”
the packets that transit on the wire. Therefore, it does not
provide the appropriate support for applications like traffic shapers, QoS
schedulers and personal firewalls.

<h3 class="title">Npcap Features</h3>

Npcap has many exciting features that set it above other packet capture solutions:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem"> Built for modern Windows: Npcap is written for Windows 10, Windows 8.1, Windows 8, and Windows 7. Using up-to-date NDIS versions, it allows you to capture traffic without slowing down the network stack. Npcap is implemented as a NDIS 6 Lightweight Filter driver, faster and with less overhead
than the legacy <a class="ulink" href="https://docs.microsoft.com/en-us/previous-versions/windows/hardware/network/ff557149(v=vs.85)"
target="_top">NDIS 5 Protocol Driver
used by WinPcap.
<li class="listitem"> WinPcap compatibility: Npcap is a
drop-in replacement for WinPcap
in most applications.
<li class="listitem"> Updated cross-platform libpcap API:
The libpcap API allows cross-platform packet capture applications
to target Linux, Windows, macOS, BSD, Solaris and others. Npcap includes
the latest version of libpcap,
providing the best solution for compatibility, performance, functionality, and security.
<li class="listitem"> Loopback packet capture and injection: Npcap is able to
see
Windows loopback packets using the
<a class="ulink" href="https://msdn.microsoft.com/en-us/library/windows/desktop/aa366510(v=vs.85).aspx"
target="_top">
Windows Filtering Platform (WFP). Npcap supplies an
interface named “NPF_Loopback”, with the description “Adapter for loopback capture.”
Wireshark users can choose this adapter to capture all loopback traffic the same way as other
non-loopback adapters.
Packet injection works as well with <code class="function">pcap_inject()</code>.
<li class="listitem"> Raw 802.11 Packet Capture Support: Npcap is able to see
802.11 frames instead of emulated Ethernet frames on ordinary wireless

adapters. You need to select the <code class="option">Support raw 802.11 traffic (and monitor
mode) for wireless adapters</code> option in the installation wizard to enable
this feature. When your adapter is in “Monitor Mode”, Npcap will supply all
802.11 data + control + management packets with Radiotap headers. When
your adapter is in “Managed Mode”, Npcap will only supply Ethernet

packets. Npcap directly supports using Wireshark to capture in “Monitor Mode”.
Npcap also provides the <code class="filename">WlanHelper.exe</code>
tool to manually configure WiFi PHY parameters.
See more details
about this feature in the section called “For software that uses Npcap raw 802.11 feature”.
<li class="listitem"> “Admin-only Mode” Support: Npcap supports restricting its
use to Administrators for safety purpose. If Npcap is installed with
the option “Restrict Npcap driver's access to Administrators only” checked,
only Built-in Administrators may access its features via user software (Nmap, Wireshark, etc).
This provides a level of restriction similar to requiring root access for packet capture on Linux/UNIX.

<h3
class="title">Purpose of this manual</h3>

The purpose of this manual is to provide a comprehensive and easy way
to browse the documentation of the Npcap architecture. You will find
three main sections:

the section called “Npcap Users' Guide” is for end users of Npcap, and
primarily concerns installation options, hardware compatibility, and bug
reporting procedures.

the section called “Developing software with Npcap” is for programmers who need to use
Npcap from an application: it contains information about functions and
data structures exported by the Npcap API, a manual for writing packet
filters, and information on how to include it in an application. A
tutorial with
several code samples is provided as well; it can be used to
learn the basics of the Npcap API using a step-by-step approach, but it
also offers code snippets that demonstrate advanced features.

the section called “Npcap internals” is intended for Npcap developers
and maintainers, or for people who are curious about how this system
works: it provides a general description of the Npcap architecture and
explains how it works. Additionally, it documents the complete device
driver structure, the source code, the Packet.dll interface and the
low-level Npcap API. If you want to understand what happens inside Npcap

or if you need to extend it, this is the section you will want to
read.

<h3 class="title"><a
name="id569495">Terminology</h3>

We call
Npcap an architecture rather than
library because packet capture is a low level
mechanism that requires a strict interaction with the network adapter and
with the operating system, in particular with its networking
implementation, so a simple library is not sufficient.

For consistency with the literature, we will use the term
packet even though
frame is more accurate since the capture process
is done at the data-link layer and the data-link header is included in
the captured data.

<h3 class="title">Npcap
License</h3>

Even though Npcap source code is publicly available for review, it is
not open source software and may not be redistributed without special
permission from the Nmap
Project. The
Npcap
Free License
allows end users to download, install, and use up to 5 copies of
Npcap from our site for
free. Copies which are only used with Nmap, <a
class="ulink" href="https://www.wireshark.org" target="_top">Wireshark, and/or
<a class="ulink" href="https://www.microsoft.com/en-us/microsoft-365/security/identity-defender"
target="_top">Microsoft
Defender for Identity don't count toward this 5-install
limit.

We fund the Npcap project by selling the Npcap OEM
Edition. This special version of Npcap includes enterprise
features such as the silent installer and commercial support as
well as special license rights allowing customers to redistribute

Npcap with their products or to install it on more systems within
their organization
with easy enterprise deployment. We offer two
commercial license types:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">
The Npcap OEM
Redistribution License is for companies that wish to
distribute Npcap OEM within their products (the free Npcap
edition does not allow this). Licensees generally use the
Npcap OEM silent installer, ensuring a seamless experience for
end users. Licensees may choose between a perpetual unlimited
license or an annual term license, along with options for
commercial support and updates.
<li class="listitem">
The Npcap OEM
Internal-Use License is for organizations that wish to
use Npcap OEM internally without redistribution outside their
organization. This allows them to bypass the 5-system usage
cap of
the Npcap free edition. It includes commercial support
and update options, and provides the extra Npcap OEM features
such as the silent installer for enterprise-wide
deployment.

<h3 class="title"><a name="npcap-
download">Obtaining Npcap</h3>

The latest Npcap release can always be found
on the Npcap
website as an executable installer and as a source code
archive.

<h3 class="title"><a name="npcap-guide-
copyright">Acknowledgements and copyright</h3>

Npcap is an update of WinPcap.
It is developed
by the Nmap Project
as a continuation
of the project started by Yang Luo
under <a class="ulink" href="https://www.google-
melange.com/gsoc/project/details/google/gsoc2013/hsluoyz/5727390428823552" target="_top">Google Summer of

Code 2013 and
2015.
It also received many helpful tests from Wireshark
and NetScanTools.

Portions of this guide were adapted from the WinPcap documentation.
Copyright 2002-2005 Politecnico di Torino. Copyright
2005-2010 CACE Technologies. Copyright 2010-2013 Riverbed
Technology. Copyright 2022 Insecure.Com, LLC. All rights
reserved.

<hr><table width="100%" summary="Navigation footer"><tr><td
width="40%" align="left"></td><td width="20%" align="center"></td><td width="40%" align="right"><a
accesskey="n" href="npcap-users-guide.html">Next</td></tr><tr><td width="40%" align="left"
valign="top"></td><td width="20%" align="center"></td><td width="40%" align="right" valign="top">Npcap
Users' Guide</td></tr></table> </body></html>

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/docs/index.html
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2006 Paolo Abeni (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

```
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
 FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* Basic USB data struct
* By Paolo Abeni <paolo.abeni@email.it>
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/usb.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2002 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2009 CACE Technologies, Inc. Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
```

AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/pcap-inttypes.h
No license file was found, but licenses were detected in source scan.

/**********************IMPORTANT NPCAP LICENSE TERMS**********************
*
* Npcap (https://npcap.com) is a Windows packet sniffing driver and library
* and is copyright (c) 2013-2022 by Nmap Software LLC ("The Nmap Project").
* All rights reserved.
*
* Even though Npcap source code is publicly available for review, it
* is not open source software and may not be redistributed or used in
* other software without special permission from the Nmap
* Project. The standard (free) version is usually limited to
* installation on five systems. For more details, see the LICENSE
* file included with Npcap and also avaialble at
* https://github.com/nmap/npcap/blob/master/LICENSE. This header file
* summarizes a few important aspects of the Npcap license, but is not
* a substitute for that full Npcap license agreement.
*
* We fund the Npcap project by selling two types of commercial licenses to a
* special Npcap
OEM edition:
*
* 1) The Npcap OEM Redistribution License allows companies distribute Npcap
* OEM within their products. Licensees generally use the Npcap OEM silent
* installer, ensuring a seamless experience for end users. Licensees may
* choose between a perpetual unlimited license or a quarterly term license,
* along with options for commercial support and updates. Prices and details:
* https://npcap.com/oem/redist.html
*
* 2) The Npcap OEM Internal-Use License is for organizations that wish to
* use Npcap OEM internally, without redistribution outside their
* organization. This allows them to bypass the 5-system usage cap of the
* Npcap free edition. It includes commercial support and update options, and

```
* provides the extra Npcap OEM features such as the silent installer for
* automated deployment. Prices and details:
* https://npcap.com/oem/internal.html
*
* Both of these licenses include updates and support as well as a
* warranty. Npcap OEM also includes
 a silent installer for unattended
* installation. Further details about Npcap OEM are available from
* https://npcap.com/oem/, and you are also welcome to contact us at
* sales@nmap.com to ask any questions or set up a license for your
* organization.
*
* Free and open source software producers are also welcome to contact us for
* redistribution requests. However, we normally recommend that such authors
* instead ask your users to download and install Npcap themselves. It will
* be free for them if they need 5 or fewer copies.
*
* If the Nmap Project (directly or through one of our commercial
* licensing customers) has granted you additional rights to Npcap or
* Npcap OEM, those additional rights take precedence where they
* conflict with the terms of the license agreement.
*
* Since the Npcap source code is available for download and review, users
* sometimes contribute code patches to fix bugs or add new features.  By
* sending these changes to the Nmap Project (including
 through direct email
* or our mailing lists or submitting pull requests through our source code
* repository), it is understood unless you specify otherwise that you are
* offering the Nmap Project the unlimited, non-exclusive right to reuse,
* modify, and relicense your code contribution so that we may (but are not
* obligated to) incorporate it into Npcap.  If you wish to specify special
* license conditions or restrictions on your contributions, just say so when
* you send them.
*
* This software is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
* or FITNESS FOR A PARTICULAR PURPOSE. Warranty rights and commercial
* support are available for the OEM Edition described above.
*
* Other copyright notices and attribution may appear below this license
* header. We have kept those for attribution purposes, but any license terms
* granted by those notices apply only to their original work,
 and not to any
* changes made by the Nmap Project or to this entire file.
*
*****************************************************************************/
/*
* Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
```

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/Packet32.h
No license file was found, but licenses were detected in source scan.

<html><head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><title>Npcap Development Tutorial</title><meta name="generator" content="DocBook XSL Stylesheets V1.79.2"><meta name="description" content="A step-by-step guide to writing software that uses Npcap to list network adapters, capture packets, and send network traffic."><link rel="home" href="index.html" title="Npcap Reference Guide"><link rel="up" href="index.html" title="Npcap Reference Guide"><link rel="prev" href="npcap-api.html" title="The Npcap API"><link rel="next" href="npcap-internals.html" title="Npcap internals"></head><body bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF"> <table width="100%" summary="Navigation header"><tr><th colspan="3" align="center">Npcap Development Tutorial</th></tr><tr><td width="20%" align="left">Prev</td><th width="60%" align="center"></th><td width="20%" align="right">Next</td></tr></table><hr> <h2 class="title" style="clear: both">Npcap Development

Tutorial</h2> Abstract
A step-by-step guide to writing software that uses Npcap to list
network adapters, capture packets, and send network traffic.

This section shows how to use the features of the Npcap API. It is
organized as a tutorial, subdivided into a set of lessons that will
introduce the reader, in a step-by-step fashion, to program development
using Npcap, from the basic functions (obtaining the adapter list,
starting a capture, etc.) to the most advanced ones (handling send queues
and gathering statistics about network traffic).

The samples are
written in plain C, so a basic knowledge of C
programming is required. Also, since this is a tutorial about a library
dealing with "raw" networking packets, good knowledge of networks and
network protocols is assumed.

The code in this section is copied from the <a class="xref" href="npcap-devguide.html#npcap-examples"
title="Examples">the section called “Examples” in the source
distribution and the SDK. The code is released under a BSD-3-clause license and
copyright: NetGroup, Politecnico di Torino (Italy); CACE Technologies,
Davis (California); and Insecure.com, LLC. Full text of the code license
can be found in each source file.

<h3 class="title"><a name="npcap-tutorial-
devlist">Obtaining the device list</h3>

Typically, the first thing that a Npcap-based application does is
get a list of attached network adapters. Both libpcap and Npcap
provide
the pcap_findalldevs_ex() function for
this purpose:
this function returns a linked list of <code class="literal">pcap_if</code> structures, each of which contains
comprehensive information about an attached adapter. In particular, the
fields <code class="literal">name</code> and <code class="literal">description</code> contain the name and a
human readable description, respectively, of the corresponding
device.

The following code retrieves the adapter list and shows it on the
screen, printing an error if no adapters are found.

<pre class="programlisting">
#include "pcap.h"

```
main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int i=0;
 char errbuf[PCAP_ERRBUF_SIZE];

 /* Retrieve the device list from the local machine */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
   NULL /* auth is not needed */,
   &amp;alldevs, errbuf) == -1)
 {
   fprintf(stderr,
      "Error in pcap_findalldevs_ex: %s\n",
     errbuf);
   exit(1);
 }

 /* Print the list */
 for(d= alldevs; d != NULL; d= d-&gt;next)
 {
   printf("%d. %s", ++i, d-&gt;name);
   if (d-&gt;description)
     printf(" (%s)\n", d-&gt;description);
   else
     printf(" (No description available)\n");
 }

 if (i == 0)
 {
   printf("\nNo interfaces found! Make sure Npcap is installed.\n");
   return;
 }

 /* We don't need any more the device list. Free it */
 pcap_freealldevs(alldevs);
}
</pre>

   <p>Some comments about this code.</p>

   <p>First of all, <a class="ulink" href="./wpcap/pcap_findalldevs.html" target="_top">pcap_findalldevs_ex()</a>,
like
     other libpcap functions, has an <code class="literal">errbuf</code> parameter. This
     parameter points to a string filled by libpcap with a description of the
     error if something goes wrong.</p>
```

Second, remember that not all the OSes supported by libpcap provide a
description
of the network interfaces, therefore if we want to write a
portable application, we must consider the case in which
<code class="literal">description</code> is null: we print the string "No
description available" in that situation.

Note finally that we free the list with <a class="ulink" href="wpcap/pcap_findalldevs.html"
target="_top">pcap_freealldevs() once when
we have finished with it.

Assuming we have compiled the program, let's try to run it. On a
particular Windows workstation, the result we optained is

<pre class="screen">
1. \Device\NPF_{4E273621-5161-46C8-895A-48D0E52A0B83} (Realtek RTL8029(AS) Ethernet Adapter)
2. \Device\NPF_{5D24AE04-C486-4A96-83FB-8B5EC6C7F430} (3Com EtherLink PCI)</pre>

As you can see, the name of the network adapters (that will be passed
to libpcap when opening the devices) under Windows are quite unreadable,
so the parenthetical descriptions can be very helpful.

<h3 class="title"><a name="npcap-tutorial-
devdetails">Obtaining advanced information about installed devices</h3>

Lesson 1 (<a class="xref" href="npcap-tutorial.html#npcap-tutorial-devlist" title="Obtaining the device
list">the section called “Obtaining the device list”) demonstrated how
to get basic information (i.e. device name and description) about
available adapters. Actually, Npcap provides also other advanced
information. In particular, every <code class="literal">pcap_if</code> structure
returned by pcap_findalldevs_ex()
contains also a list of <code class="literal">pcap_addr</code> structures,
with:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">a list of
addresses for that interface.
<li
class="listitem">a list of netmasks (each of which corresponds to an entry in
the addresses list).
<li class="listitem">a list of broadcast addresses (each of which corresponds to an
entry in the addresses list).
<li class="listitem">a list of destination addresses (each of which corresponds to
an entry in the addresses list).

Additionally, <code class="literal">pcap_findalldevs_ex()</code> can also
return remote adapters and a list of pcap files that are located in a
given local folder.

The following sample provides an ifprint() function that prints the

complete contents of a <code class="literal">pcap_if</code> structure. It is
invoked by the program for every entry returned by
<code class="literal">pcap_findalldevs_ex()</code>.</p>

<pre class="programlisting">
/* Print all the available information on the given interface */
void ifprint(pcap_if_t *d)
{
 pcap_addr_t *a;
  char ip6str[128];

 /* Name */
 printf("%s\n",d->name);

 /* Description */
 if (d->description)
  printf("\tDescription: %s\n",d->description);

 /* Loopback Address*/
 printf("\tLoopback: %s\n",(d->flags & PCAP_IF_LOOPBACK)?"yes":"no");

 /* IP addresses */
 for(a=d->addresses;a;a=a->next) {
  printf("\tAddress Family: #%d\n",a->addr->sa_family);

  switch(a->addr->sa_family)
  {
    case AF_INET:
      printf("\tAddress Family Name: AF_INET\n");
      if (a->addr)
        printf("\tAddress: %s\n",iptos(((struct sockaddr_in *)a->addr)->sin_addr.s_addr));
      if (a->netmask)
        printf("\tNetmask: %s\n",iptos(((struct sockaddr_in *)a->netmask)->sin_addr.s_addr));
      if (a->broadaddr)
        printf("\tBroadcast Address: %s\n",iptos(((struct sockaddr_in *)a->broadaddr)->sin_addr.s_addr));
      if (a->dstaddr)
        printf("\tDestination Address: %s\n",iptos(((struct sockaddr_in
 *)a->dstaddr)->sin_addr.s_addr));
      break;

    case AF_INET6:
      printf("\tAddress Family Name: AF_INET6\n");
      if (a->addr)
        printf("\tAddress: %s\n", ip6tos(a->addr, ip6str, sizeof(ip6str)));
     break;

    default:
      printf("\tAddress Family Name: Unknown\n");

```
      break;
    }
  }
 printf("\n");
}
</pre>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
openadapter"></a>Opening an adapter and capturing the packets</h3></div></div></div>


   <p>Now that we've seen how to obtain an adapter to play with, let's
     start the real job, opening an adapter and capturing some traffic. In
     this lesson we'll write a program that prints some information about each
     packet flowing through the adapter.</p>

   <p>The function that opens a capture device is <a class="ulink" href="wpcap/pcap_open.html"
target="_top">pcap_open()</a>. The parameters,
     <code
 class="literal">snaplen</code>, <code class="literal">flags</code> and
     <code class="literal">to_ms</code> deserve some explanation.</p>

   <p><code class="literal">snaplen</code> specifies the portion of the packet to capture. On
     some OSes (like xBSD and Win32), the packet driver can be configured to
     capture only the initial part of any packet: this decreases the amount of
     data to copy to the application and therefore improves the efficiency of
     the capture. In this case we use the value 65536 which is higher than the
     greatest MTU that we could encounter. In this manner we ensure that the
     application will always receive the whole packet.</p>

   <p><code class="literal">flags:</code> the most important flag is the one that
     indicates if the adapter will be put in promiscuous mode. In normal
     operation, an adapter only captures packets from the network that are
     destined to it; the packets exchanged by other hosts are therefore
      ignored. Instead, when the adapter is in promiscuous mode it captures all
     packets whether they are destined to it or not. This means that on shared
     media (like non-switched Ethernet), Npcap will be able to capture the
     packets of other hosts. Promiscuous mode is the default for most capture
     applications, so we enable it in the following example.</p>

   <p><code class="literal">to_ms</code> specifies the read timeout, in milliseconds.
     A read on the adapter (for example, with <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_dispatch()</a> or <a class="ulink" href="wpcap/pcap_next_ex.html"
target="_top">pcap_next_ex()</a>) will always
     return after <code class="literal">to_ms</code> milliseconds, even if no packets
     are available from the network. <code class="literal">to_ms</code> also defines the
```

```
    interval between statistical reports if the adapter is in statistical
    mode (see the lesson "\ref wpcap_tut9" for information
 about statistical
    mode). Setting <code class="literal">to_ms</code> to 0 means no timeout, a read on
    the adapter never returns if no packets arrive. A -1 timeout on the other
    side causes a read on the adapter to always return immediately.</p>

  <pre class="programlisting">
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

/* prototype of the packet handler */
void packet_handler(
 u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);

int main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Retrieve the device list on the local machine */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL, &amp;alldevs, errbuf) == -1)
 {
  fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
  exit(1);
 }

 /* Print
 the list */
 for(d=alldevs; d; d=d-&gt;next)
 {
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
```

```
  else
    printf(" (No description available)\n");
}

if(i==0)
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);

/* Open the device */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
        65536, // portion of the packet to capture
               // 65536 guarantees that the whole packet will
               // be captured on all the link layers
        PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode

     1000, // read timeout
        NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

printf("\nlistening on %s...\n", d-&gt;description);

/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);
```

```
 /* start the capture */
 pcap_loop(adhandle, 0, packet_handler, NULL);

 return 0;
}


/* Callback function invoked by libpcap for every incoming packet */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct tm ltime;
 char timestr[16];
 time_t local_tv_sec;

 /*
  * unused variables
  */
 (VOID)(param);
 (VOID)(pkt_data);

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime,
 &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 printf("%s,%.6d len:%d\n",
   timestr, header-&gt;ts.tv_usec, header-&gt;len);
}
</pre>

   <p>Once the adapter is opened, the capture can be started with <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_dispatch()</a> or <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_loop()</a>. These two functions are
     very similar, the difference is that <code class="literal">pcap_dispatch()</code>
     returns (although not guaranteed) when the timeout expires while
     <code class="literal">pcap_loop()</code> doesn't return until
     <code class="literal">cnt</code> packets have been captured, so it can block for an
     arbitrary period on an under-utilized network.
     <code class="literal">pcap_loop()</code> is enough for the purpose of this sample,
     while <code class="literal">pcap_dispatch()</code> is normally used in a more
     complex program.</p>

    <p>Both of these functions have a <code class="literal">callback</code> parameter,
     <code class="literal">packet_handler</code>, pointing to a function that will
```

receive the packets. This function is invoked by libpcap for every new
packet coming from the network and receives a generic status
(corresponding to the <code class="literal">user</code> parameter of pcap_loop() and pcap_dispatch()), a header with some
information on the packet like the timestamp and the length and the
actual data of the packet including all the protocol headers. Note that
the frame CRC is normally not present, because it is removed by the
network adapter after frame validation. Note also that most adapters
discard packets with wrong CRCs, therefore Npcap is normally not able
to capture them.

The above example
extracts the timestamp and the length of every
packet from the <code class="literal">pcap_pkthdr</code> header and prints them on
the screen.

Please note that there may be a drawback using pcap_loop() mainly related to the
fact that the handler is called by the packet capture driver; therefore
the user application does not have direct control over it. Another
approach (and to have more readable programs) is to use the pcap_next_ex() function, which is
presented in the next example (the section called “Capturing the packets without the callback”).

<h3 class="title">Capturing
the packets without the callback</h3>

The example program in this lesson behaves exactly like the previous
program (the section called “Opening an adapter and capturing the packets”), but it uses
pcap_next_ex() instead of
pcap_loop().

The callback-based capture mechanism of pcap_loop() is elegant and it could
be a good choice in some situations. However, handling a callback is
sometimes not practical—it often makes the program more complex
especially in situations with multithreaded applications or C++
classes.

In these cases, <a class="ulink" href="wpcap/pcap_next_ex.html"

```
 target="_top">pcap_next_ex()</a> retrievs a packet
    with a direct call&#8212;using <code class="literal">pcap_next_ex()</code>,
    packets are received only when the programmer wants them.</p>

  <p>The parameters of this function are the same as a capture callback.
    It takes an adapter descriptor and a couple of pointers that will be
    initialized and returned to the user (one to a
    <code class="literal">pcap_pkthdr</code> structure and another to a buffer with the
    packet data).</p>

  <p>In the following program, we recycle the callback code of the
    previous lesson's example and move it inside main() right after the call
    to <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a>.</p>

  <pre class="programlisting">
/* Open the device */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
    65536, // portion of the packet to capture.
          // 65536 guarantees that the whole
 packet will
          // be captured on all the link layers
    PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
    1000, // read timeout
    NULL, // authentication on the remote machine
    errbuf // error buffer
 ) ) == NULL)
{
 fprintf(stderr,
  "\nUnable to open the adapter. %s is not supported by Npcap\n",
  d-&gt;name);
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}

printf("\nlistening on %s...\n", d-&gt;description);

/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);

/* Retrieve the packets */
while((res = pcap_next_ex( adhandle, &amp;header, &amp;pkt_data)) &gt;= 0){

 if(res == 0)
  /* Timeout elapsed */
  continue;

 /* convert the timestamp to readable format */
```

```
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 printf("%s,%.6d len:%d\n", timestr, header-&gt;ts.tv_usec, header-&gt;len);
}

if(res
 == -1){
 printf("Error reading the packets: %s\n", pcap_geterr(adhandle));
 return -1;
}
</pre>

  <p>Why do we use <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> instead
of the old
    <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_next()</a>? Because
    <code class="literal">pcap_next()</code> has some drawbacks. First of all, it is
    inefficient because it hides the callback method but still relies on
    <code class="literal">pcap_dispatch()</code>. Second, it is not able to detect EOF,
    so it's not very useful when gathering packets from a file.</p>

  <p>Notice also that <code class="literal">pcap_next_ex()</code> returns different
    values for success, timeout elapsed, error and EOF conditions.</p>
 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
filtering"></a>Filtering the traffic</h3></div></div></div>


  <p>One of the most
 powerful features offered by Npcap (and by libpcap as
    well) is the filtering engine. It provides a very efficient way to
    receive subsets of the network traffic, and is (usually) integrated with
    the capture mechanism provided by Npcap. The functions used to filter
    packets are <a class="ulink" href="wpcap/pcap_compile.html" target="_top">pcap_compile()</a>
    and <a class="ulink" href="wpcap/pcap_setfilter.html" target="_top">pcap_setfilter()</a>.</p>

  <p><a class="ulink" href="wpcap/pcap_compile.html" target="_top">pcap_compile()</a> takes a
    string containing a high-level Boolean (filter) expression and produces a
    low-level byte code that can be interpreted by the fileter engine in the
    packet driver. The syntax of the boolean expression can be found in the
    <a class="ulink" href="wpcap/pcap-filter.html" target="_top">Filtering expression syntax</a>
    section of this documentation.</p>

  <p><a class="ulink" href="wpcap/pcap_setfilter.html"
 target="_top">pcap_setfilter()</a>
    associates a filter with a capture session in the kernel driver. Once
```

<code class="literal">pcap_setfilter()</code> is called, the associated filter will
be applied to all the packets coming from the network, and all the
conformant packets (i.e., packets for which the Boolean expression
evaluates to true) will be actually copied to the application.

The following code shows how to compile and set a filter. Note that
we must retrieve the netmask from the <code class="literal">pcap_if</code>
structure that describes the adapter, because some filters created by
<code class="literal">pcap_compile()</code> require it.

The filter passed to <code class="literal">pcap_compile()</code> in this code
snippet is "ip and tcp", which means to "keep only the packets that are
both IPv4 and TCP and deliver them to the application".

<pre class="programlisting">
if
 (d->addresses != NULL)
 /* Retrieve the mask of the first address of the interface */
 netmask=((struct sockaddr_in *)(d->addresses->netmask))->sin_addr.S_un.S_addr;
else
 /* If the interface is without an address
  * we suppose to be in a C class network */
 netmask=0xffffff;

//compile the filter
if (pcap_compile(adhandle, &fcode, "ip and tcp", 1, netmask) < 0)
{
 fprintf(stderr,
  "\nUnable to compile the packet filter. Check the syntax.\n");
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}

//set the filter
if (pcap_setfilter(adhandle, &fcode) < 0)
{
 fprintf(stderr,"\nError setting the filter.\n");
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}
</pre>

If you want to see some code that uses the filtering functions shown
in this lesson, look at the example presented in the next Lesson, <a class="xref" href="npcap-

tutorial.html#npcap-tutorial-interpreting" title="Interpreting the
 packets">the section called “Interpreting the packets”.

 <h3 class="title">Interpreting the packets</h3>

   Now that we are able to capture and filter network traffic, we want
    to put our knowledge to work with a simple "real world"
    application.

   In this lesson we will take code from the previous lessons and use
    these pieces to build a more useful program. the main purpose of the
    current program is to show how the protocol headers of a captured packet
    can be parsed and interpreted. The resulting application, called UDPdump,
    prints a summary of the UDP traffic on our network.

   We have chosen to parse and display the UDP protocol because it is
    more accessible than other protocols such as TCP and consequently is an
    excellent initial example. Let's look at the code:

    <pre class="programlisting">
#include <pcap.h>
#include <Winsock2.h>
#include <tchar.h>
BOOL LoadNpcapDlls()
{
 _TCHAR npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory: %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, _T("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
  return FALSE;
 }
 return TRUE;
}

/* 4 bytes IP address */
typedef struct ip_address{
 u_char byte1;

```
 u_char byte2;
 u_char byte3;
 u_char byte4;
}ip_address;

/* IPv4 header */
typedef struct ip_header{
 u_char  ver_ihl; // Version (4 bits) + IP header length (4 bits)
 u_char  tos;     // Type of service
 u_short tlen;    // Total length
 u_short identification; // Identification
 u_short flags_fo; // Flags (3 bits) + Fragment offset (13 bits)
 u_char  ttl;      // Time to live
 u_char  proto;
   // Protocol
 u_short crc;      // Header checksum
 ip_address  saddr; // Source address
 ip_address  daddr; // Destination address
 u_int  op_pad;     // Option + Padding
}ip_header;

/* UDP header*/
typedef struct udp_header{
 u_short sport; // Source port
 u_short dport; // Destination port
 u_short len;   // Datagram length
 u_short crc;   // Checksum
}udp_header;

/* prototype of the packet handler */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);


int main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];
 u_int netmask;
 char packet_filter[] = "ip and udp";
 struct bpf_program fcode;

 /* Load Npcap and its functions. */
```

```
if (!LoadNpcapDlls())
{
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
}

/* Retrieve the device list */
if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
   NULL, &amp;alldevs, errbuf) == -1)
{
  fprintf(stderr,"Error
in pcap_findalldevs: %s\n", errbuf);
  exit(1);
}

/* Print the list */
for(d=alldevs; d; d=d-&gt;next)
{
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
}

if(i==0)
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);

/* Open the adapter */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
            65536, // portion of the packet to capture.
```

```
                // 65536 grants that the whole packet
                // will be captured
on all the MACs.
        PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
        1000, // read timeout
        NULL, // remote authentication
        errbuf // error buffer
        ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Check the link layer. We support only Ethernet for simplicity. */
if(pcap_datalink(adhandle) != DLT_EN10MB)
{
  fprintf(stderr,"\nThis program works only on Ethernet networks.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

if(d-&gt;addresses != NULL)
  /* Retrieve the mask of the first address of the interface */
  netmask=((struct sockaddr_in *)(d-&gt;addresses-&gt;netmask))-&gt;sin_addr.S_un.S_addr;
else
  /* If the interface is without addresses
   * we suppose to be in a C class network */
  netmask=0xffffff;



//compile the filter
if (pcap_compile(adhandle, &amp;fcode, packet_filter, 1, netmask) &lt;0 )
{
  fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

//set the filter
if (pcap_setfilter(adhandle, &amp;fcode)&lt;0)
{
```

```
  fprintf(stderr,"\nError setting the filter.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
 }

 printf("\nlistening on %s...\n", d-&gt;description);

 /* At this point, we don't need any more the device list. Free it */
 pcap_freealldevs(alldevs);

 /* start the capture */
 pcap_loop(adhandle, 0, packet_handler, NULL);

 return 0;
}

/* Callback function invoked by libpcap for every incoming packet */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct tm ltime;
 char timestr[16];
 ip_header *ih;
 udp_header *uh;
 u_int
 ip_len;
 u_short sport,dport;
 time_t local_tv_sec;

 /*
  * Unused variable
  */
 (VOID)(param);

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 /* print timestamp and length of the packet */
 printf("%s.%.6d len:%d ", timestr, header-&gt;ts.tv_usec, header-&gt;len);

 /* retireve the position of the ip header */
 ih = (ip_header *) (pkt_data +
   14); //length of ethernet header
```

```
 /* retireve the position of the udp header */
 ip_len = (ih-&gt;ver_ihl &amp; 0xf) * 4;
 uh = (udp_header *) ((u_char*)ih + ip_len);

 /* convert from network byte order to host byte order */
 sport = ntohs( uh-&gt;sport );
 dport = ntohs( uh-&gt;dport );

 /* print ip addresses and udp ports */
 printf("%d.%d.%d.%d.%d -&gt; %d.%d.%d.%d.%d\n",
   ih-&gt;saddr.byte1,
   ih-&gt;saddr.byte2,
   ih-&gt;saddr.byte3,
   ih-&gt;saddr.byte4,

   sport,
   ih-&gt;daddr.byte1,
   ih-&gt;daddr.byte2,
   ih-&gt;daddr.byte3,
   ih-&gt;daddr.byte4,
   dport);
}
</pre>

   <p>First of all, we set the filter to "ip and udp". In this way we are
     sure that packet_handler() will receive only UDP packets over IPv4: this
     simplifies the parsing and increases the efficiency of the
     program.</p>

   <p>We have also created a couple of structs that describe the IP and UDP
     headers. These structs are used by packet_handler() to properly locate
     the various header fields.</p>

   <p>packet_handler(), although limited to a single protocol dissector
     (UDP over IPv4), shows how complex "sniffers" like tcpdump/WinDump decode
     the network traffic. Since we aren't interested in the MAC header, we
     skip it. For simplicity and before starting the capture, we check the MAC
     layer with <a class="ulink" href="wpcap/pcap_datalink.html" target="_top">pcap_datalink()</a>
     to make sure that we are dealing
  with an Ethernet network. This way we
     can be sure that the MAC header is exactly 14 bytes.</p>

   <p>The IP header is located just after the MAC header. We will extract
     the IP source and destination addresses from the IP header.</p>

   <p>Reaching the UDP header is a bit more complicated, because the IP
     header doesn't have a fixed length. Therefore, we use the IP header's
     length field to know its size. Once we know the location of the UDP
```

header, we extract the source and destination ports.

The extracted values are printed on the screen, and the result is
something like:
<pre class="screen">
\Device\Packet_{A7FD048A-5D4B-478E-B3C1-34401AC3B72F} (Xircom t 10/100 Adapter)
Enter the interface number (1-2):1
listening on Xircom CardBus Ethernet 10/100 Adapter...
16:13:15.312784 len:87 130.192.31.67.2682 -> 130.192.3.21.53
16:13:15.314796 len:137 130.192.3.21.53 -> 130.192.31.67.2682
16:13:15.322101 len:78 130.192.31.67.2683
-> 130.192.3.21.53</pre>
Each of the final 3 lines represents a different packet.

<h3 class="title">Handling offline dump files</h3>

In this lession we are going to learn how to handle packet capture to
a file (dump to file). Npcap offers a wide range of functions to save
the network traffic to a file and to read the content of
dumps—this lesson will teach how to use all of these
functions.

The format for dump files is the libpcap one. This format contains
the data of the captured packets in binary form and is a standard used by
many network tools including WinDump, Wireshark and Snort.

<h4 class="title">Saving packets to a dump file</h4>

First
of all, let's see how to write packets in libpcap
format.

The following example captures the packets from the selected
interface and saves them on a file whose name is provided by the
user.
<pre class="programlisting">
#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

/* prototype of the packet handler */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);

```
int main(int argc, char **argv)
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];
 pcap_dumper_t *dumpfile;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Check command line */
 if(argc != 2)
 {
  printf("usage: %s filename", argv[0]);
  return -1;
 }

 /* Retrieve the device list on the local machine */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL, &amp;alldevs,
errbuf) == -1)
 {
  fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
  exit(1);
 }

 /* Print the list */
 for(d=alldevs; d; d=d-&gt;next)
 {
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
 }

 if(i==0)
 {
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
```

```
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);


/* Open the device */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
            65536, // portion of the packet to capture
                  // 65536 guarantees that
the whole packet
                  // will be captured on all the link layers
            PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
            1000, // read timeout
            NULL, // authentication on the remote machine
            errbuf // error buffer
            ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Open the dump file */
dumpfile = pcap_dump_open(adhandle, argv[1]);

if(dumpfile==NULL)
{
  fprintf(stderr,"\nError opening output file\n");
  return -1;
}

printf("\nlistening on %s... Press Ctrl+C to stop...\n", d-&gt;description);
```

```
 /* At this point, we no longer need the device list. Free it */
 pcap_freealldevs(alldevs);

 /* start the capture */
 pcap_loop(adhandle, 0, packet_handler, (unsigned char *)dumpfile);

 return 0;
}

/* Callback function invoked
 by libpcap for every incoming packet */
void packet_handler(u_char *dumpfile,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 /* save the packet on the dump file */
 pcap_dump(dumpfile, header, pkt_data);
}
</pre>

    <p>As you can see, the structure of the program is very similar to the
      ones we have seen in the previous lessons. The differences
      are:</p>

    <div class="itemizedlist"><ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">a call to <a
class="ulink" href="wpcap/pcap_dump_open.html" target="_top">pcap_dump_open()</a> is issued
        once the interface is opened. This call opens a dump file and
        associates it with the interface.</li><li class="listitem">the packets are written to this file with a <a
class="ulink" href="wpcap/pcap_dump.html" target="_top">pcap_dump()</a> from the
        packet_handler() callback. The parameters of
        <code class="literal">pcap_dump()</code> are in 1-1
 correspondence with the
        parameters of <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_handler()</a>.</li></ul></div>
  </div>

  <div class="sect3"><div class="titlepage"><div><div><h4 class="title"><a name="npcap-tutorial-offline-
reading"></a>Reading packets from a dump file</h4></div></div></div>


    <p>Now that we have a dump file available, we can try to read its
      content. The following code opens a Npcap/libpcap dump file and
      displays every packet contained in the file. The file is opened with
      <a class="ulink" href="wpcap/pcap_open_offline.html" target="_top">pcap_open_offline()</a>,
      then the usual <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_loop()</a> is
      used to sequence through the packets. As you can see, reading packets
      from an offline capture is nearly identical to receiving them from a
      physical interface.</p>
```

```
    <p>This example introduces another function:
      <code class="literal">pcap_createsrcstr()</code>. This function is required to
      create a source string that begins with a marker used to tell Npcap the
      type of the source, e.g. "rpcap://" if we are going to open an adapter,
      or "file://" if we are going to open a file. This step is not required
      when <code class="literal">pcap_findalldevs_ex()</code> is used (the returned
      values already contain these strings). However, it is required in this
      example because the name of the file is read from the user
      input.</p>

    <pre class="programlisting">
#include &lt;stdio.h&gt;
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

#define LINE_LEN 16

void dispatcher_handler(u_char *,
 const struct pcap_pkthdr *,
 const u_char *);

int main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 char source[PCAP_BUF_SIZE];

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr,
 "Couldn't load Npcap\n");
  exit(1);
 }

 if(argc != 2){

  printf("usage: %s filename", argv[0]);
  return -1;

 }

 /* Create the source string according to the new Npcap syntax */
 if ( pcap_createsrcstr(  source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
          NULL,      // remote host
          NULL,      // port on the remote host
```

```
            argv[1],    // name of the file we want to open
            errbuf      // error buffer
            ) != 0)
{
  fprintf(stderr,"\nError creating a source string\n");
  return -1;
}

/* Open the capture file */
if ( (fp= pcap_open(source, // name of the device
          65536, // portion of the packet to capture
                 // 65536 guarantees that the whole packet
                 // will be captured on all the link layers
           PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
           1000, // read timeout

  NULL, // authentication on the remote machine
           errbuf // error buffer
           ) ) == NULL)
{
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return -1;
}

// read and dispatch packets until EOF is reached
pcap_loop(fp, 0, dispatcher_handler, NULL);

return 0;
}


void dispatcher_handler(u_char *temp1,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 u_int i=0;

 /*
  * Unused variable
  */
 (VOID)temp1;

 /* print pkt timestamp and pkt len */
 printf("%ld:%ld (%ld)\n", header-&gt;ts.tv_sec, header-&gt;ts.tv_usec, header-&gt;len);

 /* Print the packet */
 for (i=1; (i &lt; header-&gt;caplen + 1 ) ; i++)
 {
```

```
  printf("%.2x ", pkt_data[i-1]);
  if ( (i % LINE_LEN) == 0) printf("\n");
 }

 printf("\n\n");

}
</pre>

    <p>The following example has the same purpose of the last one, but
      <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> is used
      instead
 of the <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_loop()</a>
      callback method.</p>

    <pre class="programlisting">
#include &lt;stdio.h&gt;
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

#define LINE_LEN 16

int main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 char source[PCAP_BUF_SIZE];
 struct pcap_pkthdr *header;
 const u_char *pkt_data;
 u_int i=0;
 int res;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 if(argc != 2)
 {
  printf("usage: %s filename", argv[0]);
  return -1;
 }

 /* Create the source string according to the new Npcap syntax */
 if ( pcap_createsrcstr(  source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
```

```
        NULL,      // remote host
        NULL,      // port on the remote host
        argv[1],    // name of
the file we want to open
        errbuf      // error buffer
        ) != 0)
{
 fprintf(stderr,"\nError creating a source string\n");
 return -1;
}

/* Open the capture file */
if ( (fp= pcap_open(source, // name of the device
       65536, // portion of the packet to capture
            // 65536 guarantees that the whole packet
            // will be captured on all the link layers
        PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
        1000, // read timeout
        NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
{
 fprintf(stderr,"\nUnable to open the file %s.\n", source);
 return -1;
}

/* Retrieve the packets from the file */
while((res = pcap_next_ex( fp, &amp;header, &amp;pkt_data)) &gt;= 0)
{
 /* print pkt timestamp and pkt len */
 printf("%ld:%ld (%ld)\n", header-&gt;ts.tv_sec, header-&gt;ts.tv_usec, header-&gt;len);


 /* Print the packet */
 for (i=1; (i &lt; header-&gt;caplen + 1 ) ; i++)
 {
   printf("%.2x ", pkt_data[i-1]);
   if ( (i % LINE_LEN) == 0) printf("\n");
 }

 printf("\n\n");
}


if (res == -1)
{
 printf("Error reading the packets: %s\n", pcap_geterr(fp));
}
```

```
 return 0;
}
</pre>

  </div>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
sending"></a>Sending Packets</h3></div></div></div>

  <p>Although the name <span class="emphasis"><em>Npcap</em></span> indicates clearly that the purpose
    of the library is packet capture, other useful features for raw
    networking are provided. Among them, the user can find a complete set of
    functions to send packets.</p>

  <div class="sect3"><div class="titlepage"><div><div><h4 class="title"><a name="id582117"></a>Sending a
single packet with <code class="literal">pcap_sendpacket()</code></h4></div></div></div>


     <p>The simplest way to send a packet is shown in the following code
      snippet. After opening an adapter, <a class="ulink" href="wpcap/pcap_inject.html"
target="_top">pcap_sendpacket()</a> is called to
      send a hand-crafted packet. <code class="literal">pcap_sendpacket()</code> takes
      as arguments a buffer containing the data to send, the length of the
      buffer and the adapter that will send it. Notice that the buffer is
      sent to the net as is, without any manipulation. This means that the
      application has to create the correct protocol headers in order to send
      something meaningful.</p>

    <pre class="programlisting">
#include &lt;stdlib.h&gt;
#include &lt;stdio.h&gt;

#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

void main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 u_char packet[100];
 int i;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
```

```
  fprintf(stderr, "Couldn't
load Npcap\n");
  exit(1);
}

/* Check the validity of the command line */
if (argc != 2)
{
  printf("usage: %s interface (e.g. 'rpcap://eth0')", argv[0]);
  return;
}

/* Open the output device */
if ( (fp= pcap_open(argv[1], // name of the device
          100, // portion of the packet to capture
          PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
          1000, // read timeout
          NULL, // authentication on the remote machine
          errbuf // error buffer
          ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    argv[1]);
  return;
}

/* Supposing to be on ethernet, set mac destination to 1:1:1:1:1:1 */
packet[0]=1;
packet[1]=1;
packet[2]=1;
packet[3]=1;
packet[4]=1;
packet[5]=1;

/* set mac source to 2:2:2:2:2:2 */
packet[6]=2;
packet[7]=2;
packet[8]=2;
packet[9]=2;
packet[10]=2;
packet[11]=2;

/* Fill the rest of the packet
*/
for(i=12;i&lt;100;i++)
{
  packet[i]=(u_char)i;
```

```
 }

 /* Send down the packet */
 if (pcap_sendpacket(fp, packet, 100 /* size */) != 0)
 {
  fprintf(stderr,"\nError sending the packet: %s\n", pcap_geterr(fp));
  return;
 }

 return;
}
</pre>

  </div>

  <div class="sect3"><div class="titlepage"><div><div><h4 class="title"><a name="id582137"></a>Send queues</h4></div></div></div>


    <p>While <a class="ulink" href="wpcap/pcap_inject.html" target="_top">pcap_sendpacket()</a>
      offers a simple and immediate way to send a single packet,
      <span class="emphasis"><em>send queues</em></span> provide an advanced, powerful and
      optimized mechanism to send a collection of packets. A send queue is a
      container for a variable number of packets that will be sent to the
      network. It has a size, that represents the maximum amount of bytes it
      can store.</p>

    <p>A send queue is created calling the
      <code class="literal">pcap_sendqueue_alloc()</code>
 function, specifying the size
      of the new send queue.</p>

    <p>Once the send queue is created,
      <code class="literal">pcap_sendqueue_queue()</code> can be used to add a packet
      to the send queue. This function takes a <code class="literal">pcap_pkthdr</code>
      with the timestamp and the length and a buffer with the data of the
      packet. These parameters are the same as those received by <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> and
      <code class="literal">pcap_handler()</code>, therefore queuing a packet that was
      just captured or read from a file is a matter of passing these
      parameters to <code class="literal">pcap_sendqueue_queue()</code>.</p>

    <p>To transmit a send queue, Npcap provides the
      <code class="literal">pcap_sendqueue_transmit()</code> function. Note the third
      parameter: if nonzero, the send will be
      <span class="emphasis"><em>synchronized</em></span>,
 i.e. the relative timestamps of the
      packets will be respected. This operation requires a remarkable amount
```

```
      of CPU, because the synchronization takes place in the kernel driver
      using "busy wait" loops. Although this operation is quite CPU
      intensive, it often results in very high precision packet transmissions
      (often around few microseconds or less).</p>

    <p>Note that transmitting a send queue with
      <code class="literal">pcap_sendqueue_transmit()</code> is much more efficient
      than performing a series of <a class="ulink" href="wpcap/pcap_inject.html"
target="_top">pcap_sendpacket()</a>, because the
      send queue is buffered at kernel level drastically decreasing the
      number of context switches.</p>

    <p>When a queue is no longer needed, it can be deleted with
      <code class="literal">pcap_sendqueue_destroy()</code> that frees all the buffers
      associated with the
 send queue.</p>

    <p>The next program shows how to use send queues. It opens a capture
      file with <a class="ulink" href="wpcap/pcap_open_offline.html" target="_top">pcap_open_offline()</a>, then
      it moves the packets from the file to a properly allocated send queue.
      At his point it transmits the queue, synchronizing it if requested by
      the user.</p>

    <p>Note that the link-layer of the dumpfile is compared with the one
      of the interface that will send the packets using <a class="ulink" href="wpcap/pcap_datalink.html"
target="_top">pcap_datalink()</a>, and a warning
      is printed if they are different&amp;#8212;it is important that the
      capture-file link-layer be the same as the adapter's link layer for
      otherwise the transmission is pointless.</p>

    <pre class="programlisting">
#include &amp;lt;stdlib.h&amp;gt;
#include &amp;lt;stdio.h&amp;gt;

#include &amp;lt;pcap.h&amp;gt;

#ifdef _WIN32
#include &amp;lt;tchar.h&amp;gt;
BOOL LoadNpcapDlls()
{
 TCHAR
 npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory: %x", GetLastError());
  return FALSE;
 }
```

```
 _tcscat_s(npcap_dir, 512, TEXT("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
   fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
   return FALSE;
 }
 return TRUE;
}
#endif

void usage();

void main(int argc, char **argv)
{
 pcap_t *indesc,*outdesc;
 char errbuf[PCAP_ERRBUF_SIZE];
 char source[PCAP_BUF_SIZE];
 FILE *capfile;
 int caplen, sync;
 u_int res;
 pcap_send_queue *squeue;
 struct pcap_pkthdr *pktheader;
 u_char *pktdata;
 float cpu_time;
 u_int npacks = 0;
 errno_t fopen_error;

#ifdef _WIN32
 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
   fprintf(stderr, "Couldn't load Npcap\n");
   exit(1);
 }
#endif

 /* Check the validity of the command line */
 if (argc &lt;= 2 || argc &gt;= 5)
 {
   usage();
   return;
 }

 /* Retrieve
 the length of the capture file */
 fopen_error = fopen_s(&amp;capfile, argv[1],"rb");
 if(fopen_error != 0){
   printf("Error opening the file, errno %d.\n", fopen_error);
   return;
```

```
}

fseek(capfile , 0, SEEK_END);
caplen= ftell(capfile)- sizeof(struct pcap_file_header);
fclose(capfile);

/* Chek if the timestamps must be respected */
if(argc == 4 &amp;amp;&amp;amp; argv[3][0] == 's')
  sync = TRUE;
else
  sync = FALSE;

/* Open the capture */
/* Create the source string according to the new WinPcap syntax */
if ( pcap_createsrcstr(
          source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
          NULL,      // remote host
          NULL,      // port on the remote host
          argv[1],    // name of the file we want to open
          errbuf      // error buffer
          ) != 0)
{
  fprintf(stderr,"\nError creating a source string\n");
  return;
}

/*
Open the capture file */
if ( (indesc= pcap_open(source, 65536, PCAP_OPENFLAG_PROMISCUOUS,
                    1000, NULL, errbuf) ) == NULL)
{
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return;
}

/* Open the output adapter */
if ( (outdesc= pcap_open(argv[2], 100, PCAP_OPENFLAG_PROMISCUOUS,
                     1000, NULL, errbuf) ) == NULL)
{
  fprintf(stderr,"\nUnable to open adapter %s.\n", source);
  return;
}

/* Check the MAC type */
if (pcap_datalink(indesc) != pcap_datalink(outdesc))
{
  printf("Warning: the datalink of the capture differs"
```

```
        " from the one of the selected interface.\n");
  printf("Press a key to continue, or CTRL+C to stop.\n");
  getchar();
}

/* Allocate a send queue */
squeue = pcap_sendqueue_alloc(caplen);

/* Fill the queue with the packets from the file */
while ((res = pcap_next_ex( indesc, &amp;pktheader, &amp;pktdata)) == 1)
{
  if (pcap_sendqueue_queue(squeue,
pktheader, pktdata) == -1)
  {
    printf("Warning: packet buffer too small, not all the packets will be sent.\n");
    break;
  }

  npacks++;
}

if (res == -1)
{
  printf("Corrupted input file.\n");
  pcap_sendqueue_destroy(squeue);
  return;
}

/* Transmit the queue */

cpu_time = (float)clock ();

if ((res = pcap_sendqueue_transmit(outdesc, squeue, sync)) &lt; squeue-&gt;len)
{
  printf("An error occurred sending the packets: %s."
        " Only %d bytes were sent\n", pcap_geterr(outdesc), res);
}

cpu_time = (clock() - cpu_time)/CLK_TCK;

printf ("\n\nElapsed time: %5.3f\n", cpu_time);
printf ("\nTotal packets generated = %d", npacks);
printf ("\nAverage packets per second = %d", (int)((double)npacks/cpu_time));
printf ("\n");

/* free the send queue */
pcap_sendqueue_destroy(squeue);
```

```
 /* Close the input file */
 pcap_close(indesc);

 /*
  * close the output adapter
  * IMPORTANT: remember to close the adapter, otherwise
 there will be no
  * guarantee that all the packets will be sent!
  */
 pcap_close(outdesc);

 return;
}


void usage()
{

 printf("\nSendcap, sends a libpcap/tcpdump capture file to the net."
       " Copyright (C) 2002 Loris Degioanni.\n");
 printf("\nUsage:\n");
 printf("\t sendcap file_name adapter [s]\n");
 printf("\nParameters:\n");
 printf("\nfile_name: the name of the dump file that will be sent to the network\n");
 printf("\nadapter: the device to use. Use \"WinDump -D\" for a list of valid devices\n");
 printf("\ns: if present, forces the packets to be sent synchronously,"
       " i.e. respecting the timestamps in the dump file.\n\n");

 exit(0);
}
</pre>
   </div>
 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
statistics"></a>Gathering Statistics on the network traffic</h3></div></div></div>


   <p>This lesson shows another advanced feature of Npcap: the ability to
     collect statistics
 about network traffic. The statistical engine makes
     use of the kernel-level packet filter to efficiently classify the
     incoming packet.</p>

   <p>In order to use this feature, the programmer must open an adapter and
     put it in <span class="emphasis"><em>statistical mode</em></span>. This can be done with
     <code class="literal">pcap_setmode()</code>. In particular,
     <code class="literal">MODE_STAT</code> must be used as the <code class="literal">mode</code>
```

argument of this function.

With statistical mode, making an application that monitors the TCP traffic load is a matter of few lines of code. The following sample shows how to do it.

<pre class="programlisting">
#include <stdlib.h>
#include <stdio.h>

#include <pcap.h>

#include <tchar.h>
BOOL LoadNpcapDlls()
{
 _TCHAR npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory:
 %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, _T("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
  return FALSE;
 }
 return TRUE;
}

void usage();

void dispatcher_handler(u_char *, const struct pcap_pkthdr *, const u_char *);

void main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 struct timeval st_ts;
 u_int netmask;
 struct bpf_program fcode;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");

```
 exit(1);
}

/* Check the validity of the command line */
if (argc != 2)
{
 usage();
 return;
}

/* Open the output adapter */
if ( (fp= pcap_open(argv[1], 100, PCAP_OPENFLAG_PROMISCUOUS,
                1000, NULL, errbuf) ) == NULL)
{
 fprintf(stderr,"\nUnable to open adapter %s.\n", errbuf);
 return;
}

/* Don't care about netmask, it
won't be used for this filter */
netmask=0xffffff;

//compile the filter
if (pcap_compile(fp, &amp;fcode, "tcp", 1, netmask) &lt;0 )
{
 fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
 /* Free the device list */
 return;
}

//set the filter
if (pcap_setfilter(fp, &amp;fcode)&lt;0)
{
 fprintf(stderr,"\nError setting the filter.\n");
 pcap_close(fp);
 /* Free the device list */
 return;
}

/* Put the interface in statstics mode */
if (pcap_setmode(fp, MODE_STAT)&lt;0)
{
 fprintf(stderr,"\nError setting the mode.\n");
 pcap_close(fp);
 /* Free the device list */
 return;
}
```

```
 printf("TCP traffic summary:\n");

 /* Start the main loop */
 pcap_loop(fp, 0, dispatcher_handler, (PUCHAR)&amp;amp;st_ts);

 pcap_close(fp);
 return;
}

void dispatcher_handler(u_char *state,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct timeval *old_ts = (struct timeval *)state;
 u_int delay;
  LARGE_INTEGER Bps,Pps;
 struct tm ltime;
 char timestr[16];
 time_t local_tv_sec;

 /* Calculate the delay in microseconds from the last sample. This value
  * is obtained from the timestamp that the associated with the sample. */
 delay = (header-&amp;gt;ts.tv_sec - old_ts-&amp;gt;tv_sec) * 1000000
   - old_ts-&amp;gt;tv_usec + header-&amp;gt;ts.tv_usec;
 /* Get the number of Bits per second */
 Bps.QuadPart=(((*(LONGLONG*)(pkt_data + 8)) * 8 * 1000000) / (delay));
 /*                                     ^     ^
                                        |     |
                                        |     |
                                        |     |
                    converts bytes in bits --       |
                                                    |
                delay is expressed in microseconds --
 */

 /* Get the number of Packets per second */
 Pps.QuadPart=(((*(LONGLONG*)(pkt_data)) * 1000000) / (delay));

 /* Convert
 the timestamp to readable format */
 local_tv_sec = header-&amp;gt;ts.tv_sec;
 localtime_s(&amp;amp;ltime, &amp;amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;amp;ltime);

 /* Print timestamp*/
 printf("%s ", timestr);
```

```
 /* Print the samples */
 printf("BPS=%I64u ", Bps.QuadPart);
 printf("PPS=%I64u\n", Pps.QuadPart);

 //store current timestamp
 old_ts-&gt;tv_sec=header-&gt;ts.tv_sec;
 old_ts-&gt;tv_usec=header-&gt;ts.tv_usec;
}


void usage()
{
 printf("\nShows the TCP traffic load, in bits per second and packets per second."
       "\nCopyright (C) 2002 Loris Degioanni.\n");
 printf("\nUsage:\n");
 printf("\t tcptop adapter\n");
 printf("\t You can use \"WinDump -D\" if you don't know the name of your adapters.\n");

 exit(0);
}
</pre>

  <p>Before enabling statistical mode, the user has the option to set a
    filter that defines the subset of network traffic that will be monitored.
    See the <a class="ulink" href="wpcap/pcap-filter.html" target="_top">Filtering
 expression
      syntax</a> documentation for details. If no filter has been set,
    all of the traffic will be monitored.</p>

  <p>Once
    </p><div class="itemizedlist"><ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">the
filter is set</li><li class="listitem"><code class="literal">pcap_setmode()</code> is called</li><li
class="listitem">callback invocation is enabled with <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_loop()</a></li></ul></div><p>
    the interface descriptor starts to work in statistical mode. Notice the
    fourth parameter (<code class="literal">to_ms</code>) of <a class="ulink" href="wpcap/pcap_open.html"
target="_top">pcap_open()</a>: it defines the interval
    among the statistical samples. The callback function receives the samples
    calculated by the driver every <code class="literal">to_ms</code> milliseconds.
    These samples are encapsulated in the second and third
 parameters of the
    callback function.

    Two 64-bit counters are provided: the number of packets and the amount of
    bytes received during the last interval.</p>

  <p>In the example, the adapter is opened with a timeout of 1000 ms. This
    means that dispatcher_handler() is called once per second. At this point
```

a filter that keeps only tcp packets is compiled and set. Then
<code class="literal">pcap_setmode()</code> and <code class="literal">pcap_loop()</code> are
called. Note that a struct timeval pointer is passed to
<code class="literal">pcap_loop()</code> as the <code class="literal">user</code> parameter.
This structure will be used to store a timestamp in order to calculate
the interval between two samples. dispatcher_handler()uses this interval
to obtain the bits per second and the packets per second and then prints
these values on the screen.

Note finally that this example is by far more efficient
than a
program that captures the packets in the traditional way and calculates
statistics at user-level. Statistical mode requires the minumum amount of
data copies and context switches and therefore the CPU is optimized.
Moreover, a very small amount of memory is required.

<hr><table width="100%" summary="Navigation footer"><tr><td width="40%" align="left">Prev</td><td width="20%" align="center"></td><td width="40%" align="right">Next</td></tr><tr><td width="40%" align="left" valign="top">The Npcap API</td><td width="20%" align="center">Home</td><td width="40%" align="right" valign="top">Npcap internals</td></tr></table> </body></html>

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/docs/npcap-tutorial.html
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names
 of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.

```
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
*      @(#)bpf.h       7.1 (Berkeley) 5/7/91
*/
/*
* 208 is reserved
 for an as-yet-unspecified proprietary link-layer
* type, as requested by Will Barker.
*/
/*
* Broadcom Ethernet switches (ROBO switch) 4 bytes proprietary tagging format.
*/
/*
* Marvell (Ethertype) Distributed Switch Architecture proprietary tagging format.
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/dlt.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of CACE Technologies nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
```

```
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This sample was contributed by
* Marcin Okraszewski (Marcin.OkraszewskiATpl.compuware.com)
*
*/
```

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/pcap_fopen/pcap_fopen.cpp
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2006 Paolo Abeni (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
 FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
```

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* bluetooth data struct
* By Paolo Abeni <paolo.abeni@email.it>
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/bluetooth.h
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features
 or use of this software
*   must display the following acknowledgement:
*   This product includes software developed by the University of
*   California, Berkeley and its contributors.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap-bpf.h
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/ipnet.h
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/can_socketcan.h
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/sll.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino, CACE Technologies
* nor the names of its contributors may be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED
BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-

remote/pktdump_ex/pktdump_ex.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/UDPdump/udpdump.c
*
 /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/sendcap/sendcap.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-pcap/iflist/iflist.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/iflist/iflist.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/TestPacketCapture/TestPacketCapture.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-pcap/pcap_filter/pcap_filter.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/pcap_filter/pcap_filter.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/smp_1/smp_1.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/TestPacketSend/TestPacketSend.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-pcap/pktdump_ex/pktdump_ex.c
*
 /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-pcap/UDPdump/udpdump.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/GetMacAddress/GetMacAddress.c
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/tcptop/tcptop.c
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names
 of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
*   @(#)bpf.h   7.1 (Berkeley) 5/7/91
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/bpf.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2005 - 2006
* CACE Technologies, Davis, CA
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of CACE Technologies nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
 INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Examples-remote/UserLevelBridge/UserBridge.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the
 Laboratory may be used
*   to endorse or promote products derived from this software without
*   specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/compiler-tests.h
*
 /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap.h
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/socket.h
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/funcattrs.h

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2013, Petar Alilovic,
* Faculty of Electrical Engineering and Computing, University of Zagreb
* All rights reserved
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* * Redistributions of source code must retain the above copyright notice,
*  this list of conditions and the following disclaimer.
* * Redistributions in binary form must reproduce the above copyright
*  notice, this list of conditions and the following disclaimer in the
*  documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS
 BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
* DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/nflog.h
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
```

* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the
Laboratory may be used
* to endorse or promote products derived from this software without
* specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
/*
* Remote packet capture mechanisms and extensions from WinPcap:
*
* Copyright
(c) 2002 - 2003
* NetGroup, Politecnico di Torino (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911979002_1708442417.4332352/0/npcap-sdk-1-13-zip/Include/pcap/pcap.h

1.13 winpcap 1.13

1.13.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names
 of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.
*
*	@(#)bpf.h	7.1 (Berkeley) 5/7/91
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/bpf.h
No license file was found, but licenses were detected in source scan.

<html><head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><title>Npcap Development Tutorial</title><meta name="generator" content="DocBook XSL Stylesheets V1.79.2"><meta name="description" content="A step-by-step guide to writing software that uses Npcap to list network adapters, capture packets, and send network traffic."><link rel="home" href="index.html" title="Npcap Reference Guide"><link rel="up" href="index.html" title="Npcap Reference Guide"><link rel="prev" href="npcap-api.html" title="The Npcap API"><link rel="next" href="npcap-internals.html" title="Npcap internals"></head><body bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF"> <table width="100%" summary="Navigation header"><tr><th colspan="3" align="center">Npcap Development Tutorial</th></tr><tr><td width="20%" align="left">Prev</td><th width="60%" align="center"></th><td width="20%" align="right">Next</td></tr></table><hr> <h2 class="title" style="clear: both">Npcap Development Tutorial</h2> Abstract
A step-by-step guide to writing software that uses Npcap to list network adapters, capture packets, and send network traffic.

This section shows how to use the features of the Npcap API. It is organized as a tutorial, subdivided into a set of lessons that will introduce the reader, in a step-by-step fashion, to program development using Npcap, from the basic functions (obtaining the adapter list, starting a capture, etc.) to the most advanced ones (handling send queues and gathering statistics about network traffic).

The samples are written in plain C, so a basic knowledge of C programming is required. Also, since this is a tutorial about a library dealing with "raw" networking packets, good knowledge of networks and network protocols is assumed.

The code in this section is copied from the the section called “Examples” in the source distribution and the SDK. The code is released under a BSD-3-clause license and copyright: NetGroup, Politecnico di Torino (Italy); CACE Technologies, Davis (California); and Insecure.com, LLC. Full text of the code license

can be found in each source file.

<h3 class="title">Obtaining the device list</h3>

Typically, the first thing that a Npcap-based application does is
get a list of attached network adapters. Both libpcap and Npcap
provide
the pcap_findalldevs_ex() function for this purpose:
this function returns a linked list of <code class="literal">pcap_if</code> structures, each of which contains
comprehensive information about an attached adapter. In particular, the
fields <code class="literal">name</code> and <code class="literal">description</code> contain the name and a
human readable description, respectively, of the corresponding
device.

The following code retrieves the adapter list and shows it on the
screen, printing an error if no adapters are found.

<pre class="programlisting">
#include "pcap.h"

main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int i=0;
 char errbuf[PCAP_ERRBUF_SIZE];

 /* Retrieve the device list from the local machine */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL /* auth is not needed */,
  &alldevs, errbuf) == -1)
 {
  fprintf(stderr,
    "Error in pcap_findalldevs_ex: %s\n",
    errbuf);
  exit(1);
 }

 /* Print the list */
 for(d= alldevs; d != NULL; d= d->next)
 {
  printf("%d. %s", ++i, d->name);
  if (d->description)
    printf(" (%s)\n", d->description);
  else
    printf(" (No description available)\n");

```
 }

 if (i == 0)
 {
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return;
 }

 /* We don't need any more the device list. Free it */
 pcap_freealldevs(alldevs);
}
</pre>

  <p>Some comments about this code.</p>

  <p>First of all, <a class="ulink" href="./wpcap/pcap_findalldevs.html" target="_top">pcap_findalldevs_ex()</a>, like
    other libpcap functions, has an <code class="literal">errbuf</code> parameter. This
    parameter points to a string filled by libpcap with a description of the
    error if something goes wrong.</p>

  <p>Second, remember that not all the OSes supported by libpcap provide a
    description
 of the network interfaces, therefore if we want to write a
    portable application, we must consider the case in which
    <code class="literal">description</code> is null: we print the string "No
    description available" in that situation.</p>

  <p>Note finally that we free the list with <a class="ulink" href="wpcap/pcap_findalldevs.html"
target="_top">pcap_freealldevs()</a> once when
    we have finished with it.</p>

  <p>Assuming we have compiled the program, let's try to run it. On a
    particular Windows workstation, the result we optained is</p>

  <pre class="screen">
1. \Device\NPF_{4E273621-5161-46C8-895A-48D0E52A0B83} (Realtek RTL8029(AS) Ethernet Adapter)
2. \Device\NPF_{5D24AE04-C486-4A96-83FB-8B5EC6C7F430} (3Com EtherLink PCI)</pre>

  <p>As you can see, the name of the network adapters (that will be passed
    to libpcap when opening the devices) under Windows are quite unreadable,
    so the parenthetical descriptions can be very helpful.</p>
 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
devdetails"></a>Obtaining advanced information about installed devices</h3></div></div></div>
```

Lesson 1 (the section called “Obtaining the device list”) demonstrated how
to get basic information (i.e. device name and description) about
available adapters. Actually, Npcap provides also other advanced
information. In particular, every <code class="literal">pcap_if</code> structure
returned by pcap_findalldevs_ex()
contains also a list of <code class="literal">pcap_addr</code> structures,
with:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">a list of addresses for that interface.
<li
class="listitem">a list of netmasks (each of which corresponds to an entry in
the addresses list).
<li class="listitem">a list of broadcast addresses (each of which corresponds to an
entry in the addresses list).
<li class="listitem">a list of destination addresses (each of which corresponds to
an entry in the addresses list).

Additionally, <code class="literal">pcap_findalldevs_ex()</code> can also
return remote adapters and a list of pcap files that are located in a
given local folder.

The following sample provides an ifprint() function that prints the
complete contents of a <code class="literal">pcap_if</code> structure. It is
invoked by the program for every entry returned by
<code class="literal">pcap_findalldevs_ex()</code>.

<pre class="programlisting">
```
/* Print all the available information on the given interface */
void ifprint(pcap_if_t *d)
{
 pcap_addr_t *a;
  char ip6str[128];

 /* Name */
 printf("%s\n",d-&gt;name);

 /* Description */
 if (d-&gt;description)
  printf("\tDescription: %s\n",d-&gt;description);

 /* Loopback Address*/
 printf("\tLoopback: %s\n",(d-&gt;flags &amp; PCAP_IF_LOOPBACK)?"yes":"no");

 /* IP addresses */
 for(a=d-&gt;addresses;a;a=a-&gt;next) {
  printf("\tAddress Family: #%d\n",a-&gt;addr-&gt;sa_family);

  switch(a-&gt;addr-&gt;sa_family)
  {
```

```
    case AF_INET:
      printf("\tAddress Family Name: AF_INET\n");
      if (a-&gt;addr)
        printf("\tAddress: %s\n",iptos(((struct sockaddr_in *)a-&gt;addr)-&gt;sin_addr.s_addr));
      if (a-&gt;netmask)
        printf("\tNetmask: %s\n",iptos(((struct sockaddr_in *)a-&gt;netmask)-&gt;sin_addr.s_addr));
      if (a-&gt;broadaddr)
        printf("\tBroadcast Address: %s\n",iptos(((struct sockaddr_in *)a-&gt;broadaddr)-&gt;sin_addr.s_addr));
      if (a-&gt;dstaddr)
        printf("\tDestination Address: %s\n",iptos(((struct sockaddr_in
 *)a-&gt;dstaddr)-&gt;sin_addr.s_addr));
      break;

    case AF_INET6:
      printf("\tAddress Family Name: AF_INET6\n");
      if (a-&gt;addr)
        printf("\tAddress: %s\n", ip6tos(a-&gt;addr, ip6str, sizeof(ip6str)));
     break;

    default:
      printf("\tAddress Family Name: Unknown\n");
      break;
  }
 }
 printf("\n");
}
</pre>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
openadapter"></a>Opening an adapter and capturing the packets</h3></div></div></div>


   <p>Now that we've seen how to obtain an adapter to play with, let's
    start the real job, opening an adapter and capturing some traffic. In
    this lesson we'll write a program that prints some information about each
    packet flowing through the adapter.</p>

   <p>The function that opens a capture device is <a class="ulink" href="wpcap/pcap_open.html"
target="_top">pcap_open()</a>. The parameters,
    <code
 class="literal">snaplen</code>, <code class="literal">flags</code> and
    <code class="literal">to_ms</code> deserve some explanation.</p>

   <p><code class="literal">snaplen</code> specifies the portion of the packet to capture. On
    some OSes (like xBSD and Win32), the packet driver can be configured to
    capture only the initial part of any packet: this decreases the amount of
```

```
    data to copy to the application and therefore improves the efficiency of
    the capture. In this case we use the value 65536 which is higher than the
    greatest MTU that we could encounter. In this manner we ensure that the
    application will always receive the whole packet.</p>

   <p><code class="literal">flags:</code> the most important flag is the one that
    indicates if the adapter will be put in promiscuous mode. In normal
    operation, an adapter only captures packets from the network that are
    destined to it; the packets exchanged by other hosts are therefore
     ignored. Instead, when the adapter is in promiscuous mode it captures all
    packets whether they are destined to it or not. This means that on shared
    media (like non-switched Ethernet), Npcap will be able to capture the
    packets of other hosts. Promiscuous mode is the default for most capture
    applications, so we enable it in the following example.</p>

   <p><code class="literal">to_ms</code> specifies the read timeout, in milliseconds.
    A read on the adapter (for example, with <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_dispatch()</a> or <a class="ulink" href="wpcap/pcap_next_ex.html"
target="_top">pcap_next_ex()</a>) will always
    return after <code class="literal">to_ms</code> milliseconds, even if no packets
    are available from the network. <code class="literal">to_ms</code> also defines the
    interval between statistical reports if the adapter is in statistical
    mode (see the lesson "\ref wpcap_tut9" for information
 about statistical
    mode). Setting <code class="literal">to_ms</code> to 0 means no timeout, a read on
    the adapter never returns if no packets arrive. A -1 timeout on the other
    side causes a read on the adapter to always return immediately.</p>

   <pre class="programlisting">
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

/* prototype of the packet handler */
void packet_handler(
 u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);

int main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];

 /* Load Npcap and its functions. */
```

```
if (!LoadNpcapDlls())
{
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
}

/* Retrieve the device list on the local machine */
if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL, &amp;amp;alldevs, errbuf) == -1)
{
  fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
  exit(1);
}

/* Print
the list */
for(d=alldevs; d; d=d-&gt;next)
{
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
}

if(i==0)
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);

/* Open the device */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
        65536, // portion of the packet to capture
```

```
            // 65536 guarantees that the whole packet will
            // be captured on all the link layers
        PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode

     1000, // read timeout
        NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

 printf("\nlistening on %s...\n", d-&gt;description);

 /* At this point, we don't need any more the device list. Free it */
 pcap_freealldevs(alldevs);

 /* start the capture */
 pcap_loop(adhandle, 0, packet_handler, NULL);

 return 0;
}


/* Callback function invoked by libpcap for every incoming packet */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct tm ltime;
 char timestr[16];
 time_t local_tv_sec;

 /*
  * unused variables
  */
 (VOID)(param);
 (VOID)(pkt_data);

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime,
```

&local_tv_sec);
strftime(timestr, sizeof timestr, "%H:%M:%S", &ltime);

printf("%s,%.6d len:%d\n",
  timestr, header->ts.tv_usec, header->len);
}
</pre>

  Once the adapter is opened, the capture can be started with pcap_dispatch() or pcap_loop(). These two functions are
    very similar, the difference is that <code class="literal">pcap_dispatch()</code>
    returns (although not guaranteed) when the timeout expires while
    <code class="literal">pcap_loop()</code> doesn't return until
    <code class="literal">cnt</code> packets have been captured, so it can block for an
    arbitrary period on an under-utilized network.
    <code class="literal">pcap_loop()</code> is enough for the purpose of this sample,
    while <code class="literal">pcap_dispatch()</code> is normally used in a more
    complex program.

  Both of these functions have a <code class="literal">callback</code> parameter,
    <code class="literal">packet_handler</code>, pointing to a function that will
    receive the packets. This function is invoked by libpcap for every new
    packet coming from the network and receives a generic status
    (corresponding to the <code class="literal">user</code> parameter of pcap_loop() and pcap_dispatch()), a header with some
    information on the packet like the timestamp and the length and the
    actual data of the packet including all the protocol headers. Note that
    the frame CRC is normally not present, because it is removed by the
    network adapter after frame validation. Note also that most adapters
    discard packets with wrong CRCs, therefore Npcap is normally not able
    to capture them.

  The above example
 extracts the timestamp and the length of every
    packet from the <code class="literal">pcap_pkthdr</code> header and prints them on
    the screen.

  Please note that there may be a drawback using pcap_loop() mainly related to the
    fact that the handler is called by the packet capture driver; therefore
    the user application does not have direct control over it. Another
    approach (and to have more readable programs) is to use the pcap_next_ex() function, which is
    presented in the next example (the section called “Capturing the packets without the callback”).

<h3 class="title">Capturing
the packets without the callback</h3>

The example program in this lesson behaves exactly like the previous
program (the section called “Opening an adapter and capturing the packets”), but it uses
pcap_next_ex() instead of
pcap_loop().

The callback-based capture mechanism of pcap_loop() is elegant and it could
be a good choice in some situations. However, handling a callback is
sometimes not practical—it often makes the program more complex
especially in situations with multithreaded applications or C++
classes.

In these cases, pcap_next_ex() retrievs a packet
with a direct call—using <code class="literal">pcap_next_ex()</code>,
packets are received only when the programmer wants them.

The parameters of this function are the same as a capture callback.
It takes an adapter descriptor and a couple of pointers that will be
initialized and returned to the user (one to a
<code class="literal">pcap_pkthdr</code> structure and another to a buffer with the
packet data).

In the following program, we recycle the callback code of the
previous lesson's example and move it inside main() right after the call
to pcap_next_ex().

<pre class="programlisting">
/* Open the device */
if ((adhandle= pcap_open(d->name, // name of the device
    65536, // portion of the packet to capture.
        // 65536 guarantees that the whole
 packet will
        // be captured on all the link layers
    PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
    1000, // read timeout
    NULL, // authentication on the remote machine
    errbuf // error buffer
)) == NULL)

```
{
 fprintf(stderr,
   "\nUnable to open the adapter. %s is not supported by Npcap\n",
   d-&gt;name);
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}

printf("\nlistening on %s...\n", d-&gt;description);

/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);

/* Retrieve the packets */
while((res = pcap_next_ex( adhandle, &amp;header, &amp;pkt_data)) &gt;= 0){

 if(res == 0)
   /* Timeout elapsed */
   continue;

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 printf("%s,%.6d len:%d\n", timestr, header-&gt;ts.tv_usec, header-&gt;len);
}

if(res
 == -1){
 printf("Error reading the packets: %s\n", pcap_geterr(adhandle));
 return -1;
}
</pre>

   <p>Why do we use <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> instead
of the old
     <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_next()</a>? Because
     <code class="literal">pcap_next()</code> has some drawbacks. First of all, it is
     inefficient because it hides the callback method but still relies on
     <code class="literal">pcap_dispatch()</code>. Second, it is not able to detect EOF,
     so it's not very useful when gathering packets from a file.</p>

   <p>Notice also that <code class="literal">pcap_next_ex()</code> returns different
     values for success, timeout elapsed, error and EOF conditions.</p>
 </div>
```

<h3 class="title">Filtering the traffic</h3>

One of the most
powerful features offered by Npcap (and by libpcap as
well) is the filtering engine. It provides a very efficient way to
receive subsets of the network traffic, and is (usually) integrated with
the capture mechanism provided by Npcap. The functions used to filter
packets are pcap_compile()
and pcap_setfilter().

pcap_compile() takes a
string containing a high-level Boolean (filter) expression and produces a
low-level byte code that can be interpreted by the fileter engine in the
packet driver. The syntax of the boolean expression can be found in the
Filtering expression syntax
section of this documentation.

<a class="ulink" href="wpcap/pcap_setfilter.html"
target="_top">pcap_setfilter()
associates a filter with a capture session in the kernel driver. Once
<code class="literal">pcap_setfilter()</code> is called, the associated filter will
be applied to all the packets coming from the network, and all the
conformant packets (i.e., packets for which the Boolean expression
evaluates to true) will be actually copied to the application.

The following code shows how to compile and set a filter. Note that
we must retrieve the netmask from the <code class="literal">pcap_if</code>
structure that describes the adapter, because some filters created by
<code class="literal">pcap_compile()</code> require it.

The filter passed to <code class="literal">pcap_compile()</code> in this code
snippet is "ip and tcp", which means to "keep only the packets that are
both IPv4 and TCP and deliver them to the application".

<pre class="programlisting">
if
(d->addresses != NULL)
/* Retrieve the mask of the first address of the interface */
netmask=((struct sockaddr_in *)(d->addresses->netmask))->sin_addr.S_un.S_addr;
else
/* If the interface is without an address
 * we suppose to be in a C class network */
netmask=0xffffff;

```
//compile the filter
if (pcap_compile(adhandle, &amp;fcode, "ip and tcp", 1, netmask) &lt; 0)
{
 fprintf(stderr,
   "\nUnable to compile the packet filter. Check the syntax.\n");
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}

//set the filter
if (pcap_setfilter(adhandle, &amp;fcode) &lt; 0)
{
 fprintf(stderr,"\nError setting the filter.\n");
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}
</pre>

   <p>If you want to see some code that uses the filtering functions shown
     in this lesson, look at the example presented in the next Lesson, <a class="xref" href="npcap-
tutorial.html#npcap-tutorial-interpreting" title="Interpreting the
 packets">the section called &#8220;Interpreting the packets&#8221;</a>.</p>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
interpreting"></a>Interpreting the packets</h3></div></div></div>


   <p>Now that we are able to capture and filter network traffic, we want
     to put our knowledge to work with a simple "real world"
     application.</p>

   <p>In this lesson we will take code from the previous lessons and use
     these pieces to build a more useful program. the main purpose of the
     current program is to show how the protocol headers of a captured packet
     can be parsed and interpreted. The resulting application, called UDPdump,
     prints a summary of the UDP traffic on our network.</p>

   <p>We have chosen to parse and display the UDP protocol because it is
     more accessible than other protocols such as TCP and consequently is an
     excellent initial example. Let's look at the code:</p>

    <pre class="programlisting">
#include &lt;pcap.h&gt;
#include &lt;Winsock2.h&gt;
```

```
#include &lt;tchar.h&gt;
BOOL LoadNpcapDlls()
{
 _TCHAR npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory: %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, _T("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
  return FALSE;
 }
 return TRUE;
}


/* 4 bytes IP address */
typedef struct ip_address{
 u_char byte1;
 u_char byte2;
 u_char byte3;
 u_char byte4;
}ip_address;

/* IPv4 header */
typedef struct ip_header{
 u_char  ver_ihl; // Version (4 bits) + IP header length (4 bits)
 u_char  tos;     // Type of service
 u_short tlen;    // Total length
 u_short identification; // Identification
 u_short flags_fo; // Flags (3 bits) + Fragment offset (13 bits)
 u_char  ttl;      // Time to live
 u_char  proto;
   // Protocol
 u_short crc;      // Header checksum
 ip_address  saddr; // Source address
 ip_address  daddr; // Destination address
 u_int  op_pad;     // Option + Padding
}ip_header;

/* UDP header*/
typedef struct udp_header{
 u_short sport; // Source port
 u_short dport; // Destination port
 u_short len;   // Datagram length
```

```
 u_short crc;   // Checksum
}udp_header;

/* prototype of the packet handler */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);


int main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];
 u_int netmask;
 char packet_filter[] = "ip and udp";
 struct bpf_program fcode;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Retrieve the device list */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
   NULL, &amp;alldevs, errbuf) == -1)
 {
  fprintf(stderr,"Error
 in pcap_findalldevs: %s\n", errbuf);
  exit(1);
 }

 /* Print the list */
 for(d=alldevs; d; d=d-&gt;next)
 {
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
 }

 if(i==0)
```

```
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);

/* Open the adapter */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
          65536, // portion of the packet to capture.
                // 65536 grants that the whole packet
                // will be captured
on all the MACs.
          PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
          1000, // read timeout
          NULL, // remote authentication
          errbuf // error buffer
          ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Check the link layer. We support only Ethernet for simplicity. */
if(pcap_datalink(adhandle) != DLT_EN10MB)
{
  fprintf(stderr,"\nThis program works only on Ethernet networks.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}
```

```
if(d-&gt;addresses != NULL)
  /* Retrieve the mask of the first address of the interface */
  netmask=((struct sockaddr_in *)(d-&gt;addresses-&gt;netmask))-&gt;sin_addr.S_un.S_addr;
else
  /* If the interface is without addresses
   * we suppose to be in a C class network */
  netmask=0xffffff;



//compile the filter
if (pcap_compile(adhandle, &amp;fcode, packet_filter, 1, netmask) &lt;0 )
{
  fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

//set the filter
if (pcap_setfilter(adhandle, &amp;fcode)&lt;0)
{
  fprintf(stderr,"\nError setting the filter.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

printf("\nlistening on %s...\n", d-&gt;description);

/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);

/* start the capture */
pcap_loop(adhandle, 0, packet_handler, NULL);

return 0;
}

/* Callback function invoked by libpcap for every incoming packet */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct tm ltime;
 char timestr[16];
 ip_header *ih;
 udp_header *uh;
```

```
 u_int
 ip_len;
 u_short sport,dport;
 time_t local_tv_sec;

 /*
  * Unused variable
  */
 (VOID)(param);

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 /* print timestamp and length of the packet */
 printf("%s.%.6d len:%d ", timestr, header-&gt;ts.tv_usec, header-&gt;len);

 /* retireve the position of the ip header */
 ih = (ip_header *) (pkt_data +
   14); //length of ethernet header

 /* retireve the position of the udp header */
 ip_len = (ih-&gt;ver_ihl &amp; 0xf) * 4;
 uh = (udp_header *) ((u_char*)ih + ip_len);

 /* convert from network byte order to host byte order */
 sport = ntohs( uh-&gt;sport );
 dport = ntohs( uh-&gt;dport );

 /* print ip addresses and udp ports */
 printf("%d.%d.%d.%d.%d -&gt; %d.%d.%d.%d.%d\n",
   ih-&gt;saddr.byte1,
   ih-&gt;saddr.byte2,
   ih-&gt;saddr.byte3,
   ih-&gt;saddr.byte4,

   sport,
   ih-&gt;daddr.byte1,
   ih-&gt;daddr.byte2,
   ih-&gt;daddr.byte3,
   ih-&gt;daddr.byte4,
   dport);
}
</pre>

   <p>First of all, we set the filter to "ip and udp". In this way we are
     sure that packet_handler() will receive only UDP packets over IPv4: this
```

simplifies the parsing and increases the efficiency of the
program.

We have also created a couple of structs that describe the IP and UDP
headers. These structs are used by packet_handler() to properly locate
the various header fields.

packet_handler(), although limited to a single protocol dissector
(UDP over IPv4), shows how complex "sniffers" like tcpdump/WinDump decode
the network traffic. Since we aren't interested in the MAC header, we
skip it. For simplicity and before starting the capture, we check the MAC
layer with pcap_datalink()
to make sure that we are dealing
with an Ethernet network. This way we
can be sure that the MAC header is exactly 14 bytes.

The IP header is located just after the MAC header. We will extract
the IP source and destination addresses from the IP header.

Reaching the UDP header is a bit more complicated, because the IP
header doesn't have a fixed length. Therefore, we use the IP header's
length field to know its size. Once we know the location of the UDP
header, we extract the source and destination ports.

The extracted values are printed on the screen, and the result is
something like:
<pre class="screen">
\Device\Packet_{A7FD048A-5D4B-478E-B3C1-34401AC3B72F} (Xircom t 10/100 Adapter)
Enter the interface number (1-2):1
listening on Xircom CardBus Ethernet 10/100 Adapter...
16:13:15.312784 len:87 130.192.31.67.2682 -> 130.192.3.21.53
16:13:15.314796 len:137 130.192.3.21.53 -> 130.192.31.67.2682
16:13:15.322101 len:78 130.192.31.67.2683
-> 130.192.3.21.53</pre>
Each of the final 3 lines represents a different packet.

<h3 class="title">Handling offline dump files</h3>

In this lession we are going to learn how to handle packet capture to
a file (dump to file). Npcap offers a wide range of functions to save
the network traffic to a file and to read the content of
dumps—this lesson will teach how to use all of these
functions.

The format for dump files is the libpcap one. This format contains

the data of the captured packets in binary form and is a standard used by
many network tools including WinDump, Wireshark and Snort.

<h4 class="title">Saving packets to a dump file</h4>

First
of all, let's see how to write packets in libpcap
format.

The following example captures the packets from the selected
interface and saves them on a file whose name is provided by the
user.
<pre class="programlisting">
#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

/* prototype of the packet handler */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);

int main(int argc, char **argv)
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];
 pcap_dumper_t *dumpfile;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Check command line */
 if(argc != 2)
 {
  printf("usage: %s filename", argv[0]);
  return -1;
 }

 /* Retrieve the device list on the local machine */

```
if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL, &amp;amp;alldevs,
errbuf) == -1)
{
  fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
  exit(1);
}

/* Print the list */
for(d=alldevs; d; d=d-&gt;next)
{
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
}

if(i==0)
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);


/* Open the device */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
            65536, // portion of the packet to capture
                  // 65536 guarantees that
the whole packet
                  // will be captured on all the link layers
            PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
            1000, // read timeout
            NULL, // authentication on the remote machine
```

```
            errbuf // error buffer
            ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Open the dump file */
dumpfile = pcap_dump_open(adhandle, argv[1]);

if(dumpfile==NULL)
{
  fprintf(stderr,"\nError opening output file\n");
  return -1;
}

printf("\nlistening on %s... Press Ctrl+C to stop...\n", d-&gt;description);

/* At this point, we no longer need the device list. Free it */
pcap_freealldevs(alldevs);

/* start the capture */
pcap_loop(adhandle, 0, packet_handler, (unsigned char *)dumpfile);

return 0;
}

/* Callback function invoked
 by libpcap for every incoming packet */
void packet_handler(u_char *dumpfile,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 /* save the packet on the dump file */
 pcap_dump(dumpfile, header, pkt_data);
}
</pre>

    <p>As you can see, the structure of the program is very similar to the
      ones we have seen in the previous lessons. The differences
      are:</p>

    <div class="itemizedlist"><ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">a call to <a
class="ulink" href="wpcap/pcap_dump_open.html" target="_top">pcap_dump_open()</a> is issued
```

once the interface is opened. This call opens a dump file and
associates it with the interface.
<li class="listitem">the packets are written to this file with a pcap_dump() from the
packet_handler() callback. The parameters of
<code class="literal">pcap_dump()</code> are in 1-1
correspondence with the
parameters of pcap_handler().

<h4 class="title">Reading packets from a dump file</h4>

Now that we have a dump file available, we can try to read its
content. The following code opens a Npcap/libpcap dump file and
displays every packet contained in the file. The file is opened with
pcap_open_offline(),
then the usual pcap_loop() is
used to sequence through the packets. As you can see, reading packets
from an offline capture is nearly identical to receiving them from a
physical interface.

This example introduces another function:
<code class="literal">pcap_createsrcstr()</code>. This function is required to
create a source string that begins with a marker used to tell Npcap the
type of the source, e.g. "rpcap://" if we are going to open an adapter,
or "file://" if we are going to open a file. This step is not required
when <code class="literal">pcap_findalldevs_ex()</code> is used (the returned
values already contain these strings). However, it is required in this
example because the name of the file is read from the user
input.

<pre class="programlisting">
#include <stdio.h>
#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

#define LINE_LEN 16

void dispatcher_handler(u_char *,
 const struct pcap_pkthdr *,
 const u_char *);

int main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];

```
char source[PCAP_BUF_SIZE];

/* Load Npcap and its functions. */
if (!LoadNpcapDlls())
{
  fprintf(stderr,
"Couldn't load Npcap\n");
  exit(1);
}

if(argc != 2){

  printf("usage: %s filename", argv[0]);
  return -1;

}

/* Create the source string according to the new Npcap syntax */
if ( pcap_createsrcstr(  source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
          NULL,      // remote host
          NULL,      // port on the remote host
          argv[1],    // name of the file we want to open
          errbuf      // error buffer
          ) != 0)
{
  fprintf(stderr,"\nError creating a source string\n");
  return -1;
}

/* Open the capture file */
if ( (fp= pcap_open(source, // name of the device
         65536, // portion of the packet to capture
               // 65536 guarantees that the whole packet
               // will be captured on all the link layers
          PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
          1000, // read timeout

  NULL, // authentication on the remote machine
          errbuf // error buffer
          ) ) == NULL)
{
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return -1;
}

// read and dispatch packets until EOF is reached
pcap_loop(fp, 0, dispatcher_handler, NULL);
```

```
 return 0;
}


void dispatcher_handler(u_char *temp1,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 u_int i=0;

 /*
  * Unused variable
  */
 (VOID)temp1;

 /* print pkt timestamp and pkt len */
 printf("%ld:%ld (%ld)\n", header-&gt;ts.tv_sec, header-&gt;ts.tv_usec, header-&gt;len);

 /* Print the packet */
 for (i=1; (i &lt; header-&gt;caplen + 1 ) ; i++)
 {
   printf("%.2x ", pkt_data[i-1]);
   if ( (i % LINE_LEN) == 0) printf("\n");
 }

 printf("\n\n");

}
</pre>

    <p>The following example has the same purpose of the last one, but
      <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> is used
      instead
 of the <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_loop()</a>
      callback method.</p>

    <pre class="programlisting">
#include &lt;stdio.h&gt;
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

#define LINE_LEN 16

int main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
```

```
char source[PCAP_BUF_SIZE];
struct pcap_pkthdr *header;
const u_char *pkt_data;
u_int i=0;
int res;

/* Load Npcap and its functions. */
if (!LoadNpcapDlls())
{
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
}

if(argc != 2)
{
  printf("usage: %s filename", argv[0]);
  return -1;
}

/* Create the source string according to the new Npcap syntax */
if ( pcap_createsrcstr(  source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
          NULL,      // remote host
          NULL,      // port on the remote host
          argv[1],    // name of
the file we want to open
          errbuf      // error buffer
          ) != 0)
{
  fprintf(stderr,"\nError creating a source string\n");
  return -1;
}

/* Open the capture file */
if ( (fp= pcap_open(source, // name of the device
         65536, // portion of the packet to capture
               // 65536 guarantees that the whole packet
               // will be captured on all the link layers
          PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
          1000, // read timeout
          NULL, // authentication on the remote machine
          errbuf // error buffer
          ) ) == NULL)
{
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return -1;
}
```

```
 /* Retrieve the packets from the file */
 while((res = pcap_next_ex( fp, &amp;header, &amp;pkt_data)) &gt;= 0)
 {
  /* print pkt timestamp and pkt len */
  printf("%ld:%ld (%ld)\n", header-&gt;ts.tv_sec, header-&gt;ts.tv_usec, header-&gt;len);


  /* Print the packet */
  for (i=1; (i &lt; header-&gt;caplen + 1 ) ; i++)
  {
    printf("%.2x ", pkt_data[i-1]);
    if ( (i % LINE_LEN) == 0) printf("\n");
  }

  printf("\n\n");
 }


 if (res == -1)
 {
  printf("Error reading the packets: %s\n", pcap_geterr(fp));
 }

 return 0;
}
</pre>

  </div>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
sending"></a>Sending Packets</h3></div></div></div>

  <p>Although the name <span class="emphasis"><em>Npcap</em></span> indicates clearly that the purpose
    of the library is packet capture, other useful features for raw
    networking are provided. Among them, the user can find a complete set of
    functions to send packets.</p>

  <div class="sect3"><div class="titlepage"><div><div><h4 class="title"><a name="id582117"></a>Sending a
single packet with <code class="literal">pcap_sendpacket()</code></h4></div></div></div>


    <p>The simplest way to send a packet is shown in the following code
     snippet. After opening an adapter, <a class="ulink" href="wpcap/pcap_inject.html"
target="_top">pcap_sendpacket()</a> is called to
     send a hand-crafted packet. <code class="literal">pcap_sendpacket()</code> takes
     as arguments a buffer containing the data to send, the length of the
```

buffer and the adapter that will send it. Notice that the buffer is
sent to the net as is, without any manipulation. This means that the
application has to create the correct protocol headers in order to send
something meaningful.</p>

<pre class="programlisting">
#include <stdlib.h>
#include <stdio.h>

#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

void main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 u_char packet[100];
 int i;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't
 load Npcap\n");
  exit(1);
 }

 /* Check the validity of the command line */
 if (argc != 2)
 {
  printf("usage: %s interface (e.g. 'rpcap://eth0')", argv[0]);
  return;
 }

 /* Open the output device */
 if ((fp= pcap_open(argv[1], // name of the device
         100, // portion of the packet to capture
         PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
         1000, // read timeout
         NULL, // authentication on the remote machine
         errbuf // error buffer
)) == NULL)
 {
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    argv[1]);
  return;
 }

```
/* Supposing to be on ethernet, set mac destination to 1:1:1:1:1:1 */
packet[0]=1;
packet[1]=1;
packet[2]=1;
packet[3]=1;
packet[4]=1;
packet[5]=1;

/* set mac source to 2:2:2:2:2:2 */
packet[6]=2;
packet[7]=2;
packet[8]=2;
packet[9]=2;
packet[10]=2;
packet[11]=2;

/* Fill the rest of the packet
*/
for(i=12;i&lt;100;i++)
{
  packet[i]=(u_char)i;
}

/* Send down the packet */
if (pcap_sendpacket(fp, packet, 100 /* size */) != 0)
{
  fprintf(stderr,"\nError sending the packet: %s\n", pcap_geterr(fp));
  return;
}

return;
}
```

<h4 class="title">Send queues</h4>

While pcap_sendpacket() offers a simple and immediate way to send a single packet, send queues provide an advanced, powerful and optimized mechanism to send a collection of packets. A send queue is a container for a variable number of packets that will be sent to the network. It has a size, that represents the maximum amount of bytes it can store.

A send queue is created calling the
<code class="literal">pcap_sendqueue_alloc()</code>
function, specifying the size
of the new send queue.

Once the send queue is created,
<code class="literal">pcap_sendqueue_queue()</code> can be used to add a packet
to the send queue. This function takes a <code class="literal">pcap_pkthdr</code>
with the timestamp and the length and a buffer with the data of the
packet. These parameters are the same as those received by pcap_next_ex() and
<code class="literal">pcap_handler()</code>, therefore queuing a packet that was
just captured or read from a file is a matter of passing these
parameters to <code class="literal">pcap_sendqueue_queue()</code>.

To transmit a send queue, Npcap provides the
<code class="literal">pcap_sendqueue_transmit()</code> function. Note the third
parameter: if nonzero, the send will be
synchronized,
i.e. the relative timestamps of the
packets will be respected. This operation requires a remarkable amount
of CPU, because the synchronization takes place in the kernel driver
using "busy wait" loops. Although this operation is quite CPU
intensive, it often results in very high precision packet transmissions
(often around few microseconds or less).

Note that transmitting a send queue with
<code class="literal">pcap_sendqueue_transmit()</code> is much more efficient
than performing a series of pcap_sendpacket(), because the
send queue is buffered at kernel level drastically decreasing the
number of context switches.

When a queue is no longer needed, it can be deleted with
<code class="literal">pcap_sendqueue_destroy()</code> that frees all the buffers
associated with the
send queue.

The next program shows how to use send queues. It opens a capture
file with pcap_open_offline(), then
it moves the packets from the file to a properly allocated send queue.
At his point it transmits the queue, synchronizing it if requested by
the user.

Note that the link-layer of the dumpfile is compared with the one
of the interface that will send the packets using pcap_datalink(), and a warning

is printed if they are different—it is important that the capture-file link-layer be the same as the adapter's link layer for otherwise the transmission is pointless.</p>

<pre class="programlisting">
```
#include &lt;stdlib.h&gt;
#include &lt;stdio.h&gt;

#include &lt;pcap.h&gt;

#ifdef _WIN32
#include &lt;tchar.h&gt;
BOOL LoadNpcapDlls()
{
 TCHAR
 npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory: %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, TEXT("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
  return FALSE;
 }
 return TRUE;
}
#endif

void usage();

void main(int argc, char **argv)
{
 pcap_t *indesc,*outdesc;
 char errbuf[PCAP_ERRBUF_SIZE];
 char source[PCAP_BUF_SIZE];
 FILE *capfile;
 int caplen, sync;
 u_int res;
 pcap_send_queue *squeue;
 struct pcap_pkthdr *pktheader;
 u_char *pktdata;
 float cpu_time;
 u_int npacks = 0;
 errno_t fopen_error;
```

```
#ifdef _WIN32
 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }
#endif

 /* Check the validity of the command line */
 if (argc &lt;= 2 || argc &gt;= 5)
 {
  usage();
  return;
 }

 /* Retrieve
 the length of the capture file */
 fopen_error = fopen_s(&amp;capfile, argv[1],"rb");
 if(fopen_error != 0){
  printf("Error opening the file, errno %d.\n", fopen_error);
  return;
 }

 fseek(capfile , 0, SEEK_END);
 caplen= ftell(capfile)- sizeof(struct pcap_file_header);
 fclose(capfile);

 /* Chek if the timestamps must be respected */
 if(argc == 4 &amp;&amp; argv[3][0] == 's')
  sync = TRUE;
 else
  sync = FALSE;

 /* Open the capture */
 /* Create the source string according to the new WinPcap syntax */
 if ( pcap_createsrcstr(
          source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
          NULL,      // remote host
          NULL,      // port on the remote host
          argv[1],    // name of the file we want to open
          errbuf      // error buffer
          ) != 0)
 {
  fprintf(stderr,"\nError creating a source string\n");
  return;
 }
```

```
/*
Open the capture file */
if ( (indesc= pcap_open(source, 65536, PCAP_OPENFLAG_PROMISCUOUS,
                    1000, NULL, errbuf) ) == NULL)
{
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return;
}

/* Open the output adapter */
if ( (outdesc= pcap_open(argv[2], 100, PCAP_OPENFLAG_PROMISCUOUS,
                     1000, NULL, errbuf) ) == NULL)
{
  fprintf(stderr,"\nUnable to open adapter %s.\n", source);
  return;
}

/* Check the MAC type */
if (pcap_datalink(indesc) != pcap_datalink(outdesc))
{
  printf("Warning: the datalink of the capture differs"
         " from the one of the selected interface.\n");
  printf("Press a key to continue, or CTRL+C to stop.\n");
  getchar();
}

/* Allocate a send queue */
squeue = pcap_sendqueue_alloc(caplen);

/* Fill the queue with the packets from the file */
while ((res = pcap_next_ex( indesc, &amp;pktheader, &amp;pktdata)) == 1)
{
  if (pcap_sendqueue_queue(squeue,
pktheader, pktdata) == -1)
  {
    printf("Warning: packet buffer too small, not all the packets will be sent.\n");
    break;
  }

  npacks++;
}

if (res == -1)
{
  printf("Corrupted input file.\n");
  pcap_sendqueue_destroy(squeue);
  return;
```

```
}

/* Transmit the queue */

cpu_time = (float)clock ();

if ((res = pcap_sendqueue_transmit(outdesc, squeue, sync)) &lt; squeue-&gt;len)
{
  printf("An error occurred sending the packets: %s."
         " Only %d bytes were sent\n", pcap_geterr(outdesc), res);
}

cpu_time = (clock() - cpu_time)/CLK_TCK;

printf ("\n\nElapsed time: %5.3f\n", cpu_time);
printf ("\nTotal packets generated = %d", npacks);
printf ("\nAverage packets per second = %d", (int)((double)npacks/cpu_time));
printf ("\n");

/* free the send queue */
pcap_sendqueue_destroy(squeue);

/* Close the input file */
pcap_close(indesc);

/*
 * close the output adapter
 * IMPORTANT: remember to close the adapter, otherwise
there will be no
 * guarantee that all the packets will be sent!
 */
pcap_close(outdesc);

return;
}


void usage()
{

 printf("\nSendcap, sends a libpcap/tcpdump capture file to the net."
        " Copyright (C) 2002 Loris Degioanni.\n");
 printf("\nUsage:\n");
 printf("\t sendcap file_name adapter [s]\n");
 printf("\nParameters:\n");
 printf("\nfile_name: the name of the dump file that will be sent to the network\n");
 printf("\nadapter: the device to use. Use \"WinDump -D\" for a list of valid devices\n");
 printf("\ns: if present, forces the packets to be sent synchronously,"
```

```
        " i.e. respecting the timestamps in the dump file.\n\n");

 exit(0);
}
</pre>
  </div>
 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
statistics"></a>Gathering Statistics on the network traffic</h3></div></div></div>


  <p>This lesson shows another advanced feature of Npcap: the ability to
    collect statistics
 about network traffic. The statistical engine makes
    use of the kernel-level packet filter to efficiently classify the
    incoming packet.</p>

  <p>In order to use this feature, the programmer must open an adapter and
    put it in <span class="emphasis"><em>statistical mode</em></span>. This can be done with
    <code class="literal">pcap_setmode()</code>. In particular,
    <code class="literal">MODE_STAT</code> must be used as the <code class="literal">mode</code>
    argument of this function.</p>

  <p>With statistical mode, making an application that monitors the TCP
    traffic load is a matter of few lines of code. The following sample shows
    how to do it.</p>

  <pre class="programlisting">
#include &lt;stdlib.h&gt;
#include &lt;stdio.h&gt;

#include &lt;pcap.h&gt;

#include &lt;tchar.h&gt;
BOOL LoadNpcapDlls()
{
 _TCHAR npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory:
 %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, _T("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
```

```
  return FALSE;
 }
 return TRUE;
}

void usage();

void dispatcher_handler(u_char *, const struct pcap_pkthdr *, const u_char *);


void main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 struct timeval st_ts;
 u_int netmask;
 struct bpf_program fcode;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Check the validity of the command line */
 if (argc != 2)
 {
  usage();
  return;
 }

 /* Open the output adapter */
 if ( (fp= pcap_open(argv[1], 100, PCAP_OPENFLAG_PROMISCUOUS,
                 1000, NULL, errbuf) ) == NULL)
 {
  fprintf(stderr,"\nUnable to open adapter %s.\n", errbuf);
  return;
 }

 /* Don't care about netmask, it
 won't be used for this filter */
 netmask=0xffffff;

 //compile the filter
 if (pcap_compile(fp, &amp;fcode, "tcp", 1, netmask) &lt;0 )
 {
  fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
```

```
  /* Free the device list */
  return;
 }

 //set the filter
 if (pcap_setfilter(fp, &amp;fcode)&lt;0)
 {
  fprintf(stderr,"\nError setting the filter.\n");
  pcap_close(fp);
  /* Free the device list */
  return;
 }

 /* Put the interface in statstics mode */
 if (pcap_setmode(fp, MODE_STAT)&lt;0)
 {
  fprintf(stderr,"\nError setting the mode.\n");
  pcap_close(fp);
  /* Free the device list */
  return;
 }


 printf("TCP traffic summary:\n");

 /* Start the main loop */
 pcap_loop(fp, 0, dispatcher_handler, (PUCHAR)&amp;st_ts);

 pcap_close(fp);
 return;
}

void dispatcher_handler(u_char *state,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct timeval *old_ts = (struct timeval *)state;
 u_int delay;
  LARGE_INTEGER Bps,Pps;
 struct tm ltime;
 char timestr[16];
 time_t local_tv_sec;

 /* Calculate the delay in microseconds from the last sample. This value
  * is obtained from the timestamp that the associated with the sample. */
 delay = (header-&gt;ts.tv_sec - old_ts-&gt;tv_sec) * 1000000
   - old_ts-&gt;tv_usec + header-&gt;ts.tv_usec;
 /* Get the number of Bits per second */
```

```
 Bps.QuadPart=(((*(LONGLONG*)(pkt_data + 8)) * 8 * 1000000) / (delay));
 /*                                  ^     ^
                                     |     |
                                     |     |
                                     |     |
                    converts bytes in bits --       |
                                                    |
                delay is expressed in microseconds --
 */

 /* Get the number of Packets per second */
 Pps.QuadPart=(((*(LONGLONG*)(pkt_data)) * 1000000) / (delay));

 /* Convert
 the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 /* Print timestamp*/
 printf("%s ", timestr);

 /* Print the samples */
 printf("BPS=%I64u ", Bps.QuadPart);
 printf("PPS=%I64u\n", Pps.QuadPart);

 //store current timestamp
 old_ts-&gt;tv_sec=header-&gt;ts.tv_sec;
 old_ts-&gt;tv_usec=header-&gt;ts.tv_usec;
}



void usage()
{
 printf("\nShows the TCP traffic load, in bits per second and packets per second."
       "\nCopyright (C) 2002 Loris Degioanni.\n");
 printf("\nUsage:\n");
 printf("\t tcptop adapter\n");
 printf("\t You can use \"WinDump -D\" if you don't know the name of your adapters.\n");

 exit(0);
}
</pre>

   <p>Before enabling statistical mode, the user has the option to set a
    filter that defines the subset of network traffic that will be monitored.
    See the <a class="ulink" href="wpcap/pcap-filter.html" target="_top">Filtering
```

expression
syntax documentation for details. If no filter has been set,
all of the traffic will be monitored.

Once
<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">the filter is set
<li class="listitem"><code class="literal">pcap_setmode()</code> is called
<li class="listitem">callback invocation is enabled with pcap_loop()
the interface descriptor starts to work in statistical mode. Notice the
fourth parameter (<code class="literal">to_ms</code>) of pcap_open(): it defines the interval
among the statistical samples. The callback function receives the samples
calculated by the driver every <code class="literal">to_ms</code> milliseconds.
These samples are encapsulated in the second and third
parameters of the
callback function.

Two 64-bit counters are provided: the number of packets and the amount of
bytes received during the last interval.

In the example, the adapter is opened with a timeout of 1000 ms. This
means that dispatcher_handler() is called once per second. At this point
a filter that keeps only tcp packets is compiled and set. Then
<code class="literal">pcap_setmode()</code> and <code class="literal">pcap_loop()</code> are
called. Note that a struct timeval pointer is passed to
<code class="literal">pcap_loop()</code> as the <code class="literal">user</code> parameter.
This structure will be used to store a timestamp in order to calculate
the interval between two samples. dispatcher_handler()uses this interval
to obtain the bits per second and the packets per second and then prints
these values on the screen.

Note finally that this example is by far more efficient
than a
program that captures the packets in the traditional way and calculates
statistics at user-level. Statistical mode requires the minumum amount of
data copies and context switches and therefore the CPU is optimized.
Moreover, a very small amount of memory is required.

<hr><table width="100%" summary="Navigation footer"><tr><td width="40%" align="left">Prev</td><td width="20%" align="center"></td><td width="40%" align="right">Next</td></tr><tr><td width="40%" align="left" valign="top">The Npcap API</td><td width="20%" align="center">Home</td><td width="40%" align="right" valign="top">Npcap internals</td></tr></table> </body></html>

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/docs/npcap-tutorial.html

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2013, Petar Alilovic,
* Faculty of Electrical Engineering and Computing, University of Zagreb
* All rights reserved
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* * Redistributions of source code must retain the above copyright notice,
*  this list of conditions and the following disclaimer.
* * Redistributions in binary form must reproduce the above copyright
*  notice, this list of conditions and the following disclaimer in the
*  documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS
 BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
* DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/nflog.h
No license file was found, but licenses were detected in source scan.

```
/**********************IMPORTANT NPCAP LICENSE TERMS**********************
*
* Npcap (https://npcap.com) is a Windows packet sniffing driver and library
* and is copyright (c) 2013-2022 by Nmap Software LLC ("The Nmap Project").
* All rights reserved.
*
* Even though Npcap source code is publicly available for review, it
* is not open source software and may not be redistributed or used in
* other software without special permission from the Nmap
* Project. The standard (free) version is usually limited to
* installation on five systems. For more details, see the LICENSE
* file included with Npcap and also avaialble at
* https://github.com/nmap/npcap/blob/master/LICENSE. This header file
* summarizes a few important aspects of the Npcap license, but is not
```

* a substitute for that full Npcap license agreement.
*
* We fund the Npcap project by selling two types of commercial licenses to a
* special Npcap
 OEM edition:
*
* 1) The Npcap OEM Redistribution License allows companies distribute Npcap
* OEM within their products. Licensees generally use the Npcap OEM silent
* installer, ensuring a seamless experience for end users. Licensees may
* choose between a perpetual unlimited license or a quarterly term license,
* along with options for commercial support and updates. Prices and details:
* https://npcap.com/oem/redist.html
*
* 2) The Npcap OEM Internal-Use License is for organizations that wish to
* use Npcap OEM internally, without redistribution outside their
* organization. This allows them to bypass the 5-system usage cap of the
* Npcap free edition. It includes commercial support and update options, and
* provides the extra Npcap OEM features such as the silent installer for
* automated deployment. Prices and details:
* https://npcap.com/oem/internal.html
*
* Both of these licenses include updates and support as well as a
* warranty. Npcap OEM also includes
 a silent installer for unattended
* installation. Further details about Npcap OEM are available from
* https://npcap.com/oem/, and you are also welcome to contact us at
* sales@nmap.com to ask any questions or set up a license for your
* organization.
*
* Free and open source software producers are also welcome to contact us for
* redistribution requests. However, we normally recommend that such authors
* instead ask your users to download and install Npcap themselves. It will
* be free for them if they need 5 or fewer copies.
*
* If the Nmap Project (directly or through one of our commercial
* licensing customers) has granted you additional rights to Npcap or
* Npcap OEM, those additional rights take precedence where they
* conflict with the terms of the license agreement.
*
* Since the Npcap source code is available for download and review, users
* sometimes contribute code patches to fix bugs or add new features. By
* sending these changes to the Nmap Project (including
 through direct email
* or our mailing lists or submitting pull requests through our source code
* repository), it is understood unless you specify otherwise that you are
* offering the Nmap Project the unlimited, non-exclusive right to reuse,
* modify, and relicense your code contribution so that we may (but are not
* obligated to) incorporate it into Npcap. If you wish to specify special

```
* license conditions or restrictions on your contributions, just say so when
* you send them.
*
* This software is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
* or FITNESS FOR A PARTICULAR PURPOSE. Warranty rights and commercial
* support are available for the OEM Edition described above.
*
* Other copyright notices and attribution may appear below this license
* header. We have kept those for attribution purposes, but any license terms
* granted by those notices apply only to their original work,
 and not to any
* changes made by the Nmap Project or to this entire file.
*
***************************************************************************/
/*
* Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2007 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino, CACE Technologies
* nor the names of its contributors may be used to endorse or promote
* products derived from this
 software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/
```

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/Packet32.h
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2005 - 2006
*  CACE Technologies, Davis, CA
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of CACE Technologies nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
 INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/
```

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/UserLevelBridge/UserBridge.c
No license file was found, but licenses were detected in source scan.

```
/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California.  All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
```

* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names
 of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
*   @(#)bpf.h   7.1 (Berkeley) 5/7/91
*/
/*
* 208 is reserved
 for an as-yet-unspecified proprietary link-layer
* type, as requested by Will Barker.
*/
/*
* Broadcom Ethernet switches (ROBO switch) 4 bytes proprietary tagging format.
*/
/*
* Marvell (Ethertype) Distributed Switch Architecture proprietary tagging format.
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/dlt.h
No license file was found, but licenses were detected in source scan.

<html><head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><title>Npcap

Reference Guide</title><meta name="generator" content="DocBook XSL Stylesheets V1.79.2"><meta name="description" content="A manual and guide to Npcap, a packet capture and network analysis framework for Windows, for users and software developers. Npcap is a modern, safe, and compatible update to WinPcap."><link rel="home" href="index.html" title="Npcap Reference Guide"><link rel="next" href="npcap-users-guide.html" title="Npcap Users' Guide"></head><body bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF"> <table width="100%" summary="Navigation header"><tr><th colspan="3" align="center">Npcap Reference Guide</th></tr><tr><td width="20%" align="left"></td><th width="60%" align="center"></th><td width="20%" align="right">Next</td></tr></table><hr> <h2 class="title">Npcap Reference Guide</h2> Abstract
A manual and guide to Npcap, a packet capture and network analysis framework for Windows, for users and software developers. Npcap is a modern, safe, and compatible update to WinPcap.
<hr> Table of Contents <dl class="toc"><dt>Introduction</dt><dd><dl><dt>What is Npcap?</dt><dt>Npcap Features</dt><dt>Purpose of this manual</dt><dt>Terminology</dt><dt>Npcap License</dt><dt>Obtaining Npcap</dt><dt>Acknowledgements and copyright</dt></dl></dd><dt>Npcap Users' Guide</dt><dd><dl><dt>Installation</dt><dt>Windows platforms supported</dt><dt>How to use Wireshark to capture raw 802.11 traffic in “Monitor Mode”</dt><dt>Q & A</dt><dt>Reporting Bugs</dt></dl></dd><dt>Developing software with Npcap</dt><dd><dl><dt>Using the Npcap SDK</dt><dt>Examples</dt><dt>Updating WinPcap software to Npcap</dt><dt>How to detect what version Npcap/WinPcap you are using?</dt><dt>For software that want to use Npcap first when Npcap and WinPcap coexist</dt><dt>For software that uses Npcap loopback feature</dt><dt>For software that uses Npcap raw 802.11 feature</dt></dl></dd><dt>The Npcap API</dt><dd><dl><dt>Extensions to libpcap for Windows</dt></dl></dd><dt>Npcap Development Tutorial</dt><dd><dl><dt>Obtaining the device list</dt><dt>Obtaining advanced information about

installed devices</dt><dt>Opening an adapter and capturing the packets</dt><dt>Capturing the packets without the callback</dt><dt>Filtering the
traffic</dt><dt>Interpreting the packets</dt><dt>Handling offline dump files</dt><dt>Sending Packets</dt><dt>Gathering Statistics on the network traffic</dt></dl></dd><dt>Npcap internals</dt><dd><dl><dt>Npcap structure</dt><dt>Npcap driver internals</dt><dt>Further reading</dt></dl></dd></dl>

<h2
class="title" style="clear: both">Introduction</h2>

This Manual describes the programming interface and the source code of
Npcap. It provides detailed descriptions of the functions and structures
exported to programmers, along with complete documentation of the Npcap
internals. Several tutorials and examples are provided as well.

<h3 class="title">What is Npcap?</h3>

Npcap is an architecture for packet capture and network analysis for
Windows operating systems, consisting of a software library and a network
driver.

Most networking applications access the network through widely-used
operating system primitives such as sockets. It is easy to access data on
the network with this approach since the operating system copes with the
low level
details (protocol handling, packet reassembly, etc.) and
provides a familiar interface that is similar to the one used to read and
write files.

Sometimes, however, the “easy way” is not up to the task,
since some applications require direct access to packets on the network.
That is, they need access to the “raw” data on the network
without the interposition of protocol processing by the operating

system.

The purpose of Npcap is to give this kind of access to Windows
applications. It provides facilities to:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">capture raw packets, both the ones destined to the machine where
it's running and the ones exchanged by other hosts (on shared media)
<li class="listitem">filter the packets according to user-specified rules before
dispatching them to the application
<li class="listitem">transmit raw packets to the network
<li class="listitem">gather statistical information on the network traffic

This set of capabilities is obtained by means of a device driver,
which is installed inside the networking portion of the Windows kernel,
plus a couple of DLLs.

All of these features are exported through a powerful programming
interface, easily usable by applications. The main goal of this manual is
to document this interface, with the help of several examples.

<h4 class="title">What kind of programs use Npcap?</h4>

The Npcap programming interface can be used by many types of
network tools for analysis, troubleshooting, security and monitoring.
In particular,
classical tools that rely on Npcap are:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">network and protocol analyzers
<li class="listitem">network monitors
<li class="listitem">traffic loggers
<li class="listitem">traffic generators
<li class="listitem">user-level bridges and routers
<li class="listitem">network intrusion detection systems (NIDS)
<li class="listitem">network scanners
<li class="listitem">security tools

<h4 class="title">What Npcap can't do</h4>

Npcap receives and sends the packets independently from the host
protocols, like TCP/IP. This means that it isn't able to block, filter or
manipulate the traffic generated by other programs on the same machine: it
simply “sniffs”
the packets that transit on the wire. Therefore, it does not
provide the appropriate support for applications like traffic shapers, QoS
schedulers and personal firewalls.

<h3 class="title">Npcap Features</h3>

Npcap has many exciting features that set it above other packet capture solutions:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem"> Built for modern Windows: Npcap is written for Windows 10, Windows 8.1, Windows 8, and Windows 7. Using up-to-date NDIS versions, it allows you to capture traffic without slowing down the network stack. Npcap is implemented as a NDIS 6 Lightweight Filter driver, faster and with less overhead
than the legacy <a class="ulink" href="https://docs.microsoft.com/en-us/previous-versions/windows/hardware/network/ff557149(v=vs.85)"
target="_top">NDIS 5 Protocol Driver
used by WinPcap.
<li class="listitem"> WinPcap compatibility: Npcap is a
drop-in replacement for WinPcap
in most applications.
<li class="listitem"> Updated cross-platform libpcap API:
The libpcap API allows cross-platform packet capture applications
to target Linux, Windows, macOS, BSD, Solaris and others. Npcap includes
the latest version of libpcap,
providing the best solution for compatibility, performance, functionality, and security.
<li class="listitem"> Loopback packet capture and injection: Npcap is able to
see
Windows loopback packets using the
<a class="ulink" href="https://msdn.microsoft.com/en-us/library/windows/desktop/aa366510(v=vs.85).aspx"
target="_top">
Windows Filtering Platform (WFP). Npcap supplies an
interface named “NPF_Loopback”, with the description “Adapter for loopback capture.”
Wireshark users can choose this adapter to capture all loopback traffic the same way as other
non-loopback adapters.
Packet injection works as well with <code class="function">pcap_inject()</code>.
<li class="listitem"> Raw 802.11 Packet Capture Support: Npcap is able to see
802.11 frames instead of emulated Ethernet frames on ordinary wireless

adapters. You need to select the <code class="option">Support raw 802.11 traffic (and monitor
mode) for wireless adapters</code> option in the installation wizard to enable
this feature. When your adapter is in “Monitor

Mode”, Npcap will supply all
802.11 data + control + management packets with Radiotap headers. When
your adapter is in “Managed Mode”, Npcap will only supply Ethernet
packets. Npcap directly supports using Wireshark to capture in “Monitor Mode”.
Npcap also provides the <code class="filename">WlanHelper.exe</code>
tool to manually configure WiFi PHY parameters.
See more details
about this feature in the section called “For software that uses Npcap raw 802.11 feature”.
<li class="listitem"> “Admin-only Mode” Support: Npcap supports restricting its
use to Administrators for safety purpose. If Npcap is installed with
the option “Restrict Npcap driver's access to Administrators only” checked,
only Built-in Administrators may access its features via user software (Nmap, Wireshark, etc).
This provides a level of restriction similar to requiring root access for packet capture on Linux/UNIX.

<h3 class="title">Purpose of this manual</h3>

The purpose of this manual is to provide a comprehensive and easy way
to browse the documentation of the Npcap architecture. You will find
three main sections:

the section called “Npcap Users' Guide” is for end users of Npcap, and
primarily concerns installation options, hardware compatibility, and bug
reporting procedures.

the section called “Developing software with Npcap” is for programmers who need to use
Npcap from an application: it contains information about functions and
data structures exported by the Npcap API, a manual for writing packet
filters, and information on how to include it in an application. A
tutorial with
several code samples is provided as well; it can be used to
learn the basics of the Npcap API using a step-by-step approach, but it
also offers code snippets that demonstrate advanced features.

the section called “Npcap internals” is intended for Npcap developers

and maintainers, or for people who are curious about how this system
works: it provides a general description of the Npcap architecture and
explains how it works. Additionally, it documents the complete device
driver structure, the source code, the Packet.dll interface and the
low-level Npcap API. If you want to understand what happens inside Npcap
or if you need to extend it, this is the section you will want to
read.

<h3 class="title"><a
name="id569495">Terminology</h3>

We call
Npcap an architecture rather than
library because packet capture is a low level
mechanism that requires a strict interaction with the network adapter and
with the operating system, in particular with its networking
implementation, so a simple library is not sufficient.

For consistency with the literature, we will use the term
packet even though
frame is more accurate since the capture process
is done at the data-link layer and the data-link header is included in
the captured data.

<h3 class="title">Npcap
License</h3>

Even though Npcap source code is publicly available for review, it is
not open source software and may not be redistributed without special
permission from the Nmap
Project. The
Npcap
Free License
allows end users to download, install, and use up to 5 copies of
Npcap from our site for
free. Copies which are only used with Nmap, <a
class="ulink" href="https://www.wireshark.org" target="_top">Wireshark, and/or
<a class="ulink" href="https://www.microsoft.com/en-us/microsoft-365/security/identity-defender"
target="_top">Microsoft
Defender for Identity don't count toward this 5-install
limit.

We fund the Npcap project by selling the Npcap OEM
Edition. This special version of Npcap includes enterprise
features such as the silent installer and commercial support as
well as special license rights allowing customers to redistribute
Npcap with their products or to install it on more systems within
their organization
with easy enterprise deployment. We offer two
commercial license types:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">
The Npcap OEM
Redistribution License is for companies that wish to
distribute Npcap OEM within their products (the free Npcap
edition does not allow this). Licensees generally use the
Npcap OEM silent installer, ensuring a seamless experience for
end users. Licensees may choose between a perpetual unlimited
license or an annual term license, along with options for
commercial support and updates.
<li class="listitem">
The Npcap OEM
Internal-Use License is for organizations that wish to
use Npcap OEM internally without redistribution outside their
organization. This allows them to bypass the 5-system usage
cap of
the Npcap free edition. It includes commercial support
and update options, and provides the extra Npcap OEM features
such as the silent installer for enterprise-wide
deployment.

<h3 class="title"><a name="npcap-
download">Obtaining Npcap</h3>

The latest Npcap release can always be found
on the Npcap
website as an executable installer and as a source code
archive.

<h3 class="title"><a name="npcap-guide-
copyright">Acknowledgements and copyright</h3>

Npcap is an update of WinPcap.
It is developed

by the Nmap Project
as a continuation
of the project started by Yang Luo
under Google Summer of Code 2013 and
2015.
It also received many helpful tests from Wireshark
and NetScanTools.

Portions of this guide were adapted from the WinPcap documentation.
Copyright 2002-2005 Politecnico di Torino. Copyright
2005-2010 CACE Technologies. Copyright 2010-2013 Riverbed
Technology. Copyright 2022 Insecure.Com, LLC. All rights
reserved.

<hr><table width="100%" summary="Navigation footer"><tr><td
width="40%" align="left"></td><td width="20%" align="center"></td><td width="40%" align="right"><a
accesskey="n" href="npcap-users-guide.html">Next</td></tr><tr><td width="40%" align="left"
valign="top"></td><td width="20%" align="center"></td><td width="40%" align="right" valign="top">Npcap
Users' Guide</td></tr></table> </body></html>

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/docs/index.html
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   This product includes software developed by the University of
*   California, Berkeley and its contributors.
* 4. Neither the name of the University
 nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/vlan.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of CACE Technologies nor the names of its
* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This sample was contributed by
* Marcin Okraszewski (Marcin.OkraszewskiATpl.compuware.com)
*
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/pcap_fopen/pcap_fopen.cpp
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2006 Paolo Abeni (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
 FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* bluetooth data struct
* By Paolo Abeni <paolo.abeni@email.it>
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/bluetooth.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino, CACE Technologies
* nor the names of its contributors may be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED
BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/tcptop/tcptop.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/iflist/iflist.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/pktdump_ex/pktdump_ex.c
*
 /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/TestPacketSend/TestPacketSend.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/pcap_filter/pcap_filter.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-pcap/UDPdump/udpdump.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/UDPdump/udpdump.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/sendcap/sendcap.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-pcap/iflist/iflist.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-pcap/pcap_filter/pcap_filter.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/TestPacketCapture/TestPacketCapture.c
*
 /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/smp_1/smp_1.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/GetMacAddress/GetMacAddress.c
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Examples-pcap/pktdump_ex/pktdump_ex.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2006 Paolo Abeni (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* Basic USB data struct
* By Paolo Abeni <paolo.abeni@email.it>
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/usb.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the
 Laboratory may be used
*   to endorse or promote products derived from this software without
*   specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
/*
* Remote packet capture mechanisms and extensions from WinPcap:
*
* Copyright
(c) 2002 - 2003
* NetGroup, Politecnico di Torino (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/pcap.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1994, 1996
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the Laboratory may be
 used
*   to endorse or promote products derived from this software without
*   specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/namedb.h
*
 /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap-namedb.h
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features
or use of this software
* must display the following acknowledgement:
* This product includes software developed by the University of
* California, Berkeley and its contributors.
* 4. Neither the name of the University nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/can_socketcan.h
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/ipnet.h
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/sll.h
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap-bpf.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2002 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2009 CACE Technologies, Inc. Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino nor the names of its

* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
 AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/pcap-inttypes.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the
 Laboratory may be used
*   to endorse or promote products derived from this software without
*   specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/funcattrs.h
*
/opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap.h
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/compiler-tests.h
* /opt/cola/permits/1911979005_1708443770.983801/0/npcap-sdk-1-13-zip/Include/pcap/socket.h

1.14 microsoft-wil 20230118-89ecb2b

1.14.1 Available under license :

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE

1.15 abseil 20230125.3

1.15.1 Available under license :

Apache License
Version 2.0, January 2004

https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.16 curl 7.88.1

1.16.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2023, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
dealings
in this Software without prior written authorization of the copyright holder.

1.17 rxcpp 4.1.1

1.17.1 Available under license :

// Copyright (c) 2009 The RE2 Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//   * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//   * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//   * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE
 ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) Microsoft Open Technologies, Inc. All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License
at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.18 libtins 4.4

1.18.1 Available under license :

1.19 thousandeyes-futures 0.9

1.19.1 Available under license :

The MIT License (MIT)

Copyright (c) 2019 ThousandEyes, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.20 libpcap 1.13

1.20.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1994, 1996
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
* This product includes software developed by the Computer Systems

* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the Laboratory may be
 used
*   to endorse or promote products derived from this software without
*   specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/namedb.h
*
 /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap-namedb.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2006 Paolo Abeni (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
 FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* bluetooth data struct
* By Paolo Abeni <paolo.abeni@email.it>
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/bluetooth.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino, CACE Technologies
* nor the names of its contributors may be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED
 BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-pcap/pktdump_ex/pktdump_ex.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/iflist/iflist.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/sendcap/sendcap.c
*
 /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/pcap_filter/pcap_filter.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/UDPdump/udpdump.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/TestPacketSend/TestPacketSend.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/GetMacAddress/GetMacAddress.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-pcap/pcap_filter/pcap_filter.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/PacketDriver/TestPacketCapture/TestPacketCapture.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-pcap/iflist/iflist.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/pktdump_ex/pktdump_ex.c
*
 /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/tcptop/tcptop.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-pcap/UDPdump/udpdump.c
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/smp_1/smp_1.c
No license file was found, but licenses were detected in source scan.

/**********IMPORTANT NPCAP LICENSE TERMS**********
*
* Npcap (https://npcap.com) is a Windows packet sniffing driver and library
* and is copyright (c) 2013-2022 by Nmap Software LLC ("The Nmap Project").
* All rights reserved.
*
* Even though Npcap source code is publicly available for review, it
* is not open source software and may not be redistributed or used in
* other software without special permission from the Nmap
* Project. The standard (free) version is usually limited to
* installation on five systems. For more details, see the LICENSE
* file included with Npcap and also avaialble at
* https://github.com/nmap/npcap/blob/master/LICENSE. This header file
* summarizes a few important aspects of the Npcap license, but is not
* a substitute for that full Npcap license agreement.
*
* We fund the Npcap project by selling two types of commercial licenses to a
* special Npcap
 OEM edition:
*
* 1) The Npcap OEM Redistribution License allows companies distribute Npcap

* OEM within their products. Licensees generally use the Npcap OEM silent
* installer, ensuring a seamless experience for end users. Licensees may
* choose between a perpetual unlimited license or a quarterly term license,
* along with options for commercial support and updates. Prices and details:
* https://npcap.com/oem/redist.html
*
* 2) The Npcap OEM Internal-Use License is for organizations that wish to
* use Npcap OEM internally, without redistribution outside their
* organization. This allows them to bypass the 5-system usage cap of the
* Npcap free edition. It includes commercial support and update options, and
* provides the extra Npcap OEM features such as the silent installer for
* automated deployment. Prices and details:
* https://npcap.com/oem/internal.html
*
* Both of these licenses include updates and support as well as a
* warranty. Npcap OEM also includes
 a silent installer for unattended
* installation. Further details about Npcap OEM are available from
* https://npcap.com/oem/, and you are also welcome to contact us at
* sales@nmap.com to ask any questions or set up a license for your
* organization.
*
* Free and open source software producers are also welcome to contact us for
* redistribution requests. However, we normally recommend that such authors
* instead ask your users to download and install Npcap themselves. It will
* be free for them if they need 5 or fewer copies.
*
* If the Nmap Project (directly or through one of our commercial
* licensing customers) has granted you additional rights to Npcap or
* Npcap OEM, those additional rights take precedence where they
* conflict with the terms of the license agreement.
*
* Since the Npcap source code is available for download and review, users
* sometimes contribute code patches to fix bugs or add new features. By
* sending these changes to the Nmap Project (including
 through direct email
* or our mailing lists or submitting pull requests through our source code
* repository), it is understood unless you specify otherwise that you are
* offering the Nmap Project the unlimited, non-exclusive right to reuse,
* modify, and relicense your code contribution so that we may (but are not
* obligated to) incorporate it into Npcap. If you wish to specify special
* license conditions or restrictions on your contributions, just say so when
* you send them.
*
* This software is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
* or FITNESS FOR A PARTICULAR PURPOSE. Warranty rights and commercial
* support are available for the OEM Edition described above.

*
* Other copyright notices and attribution may appear below this license
* header. We have kept those for attribution purposes, but any license terms
* granted by those notices apply only to their original work,
 and not to any
* changes made by the Nmap Project or to this entire file.
*
**/
/*
* Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2007 CACE Technologies, Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino, CACE Technologies
* nor the names of its contributors may be used to endorse or promote
* products derived from this
 software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/Packet32.h
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997

```
* The Regents of the University of California.  All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names
 of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
*      @(#)bpf.h       7.1 (Berkeley) 5/7/91
*/
/*
* 208 is reserved
 for an as-yet-unspecified proprietary link-layer
* type, as requested by Will Barker.
*/
/*
* Broadcom Ethernet switches (ROBO switch) 4 bytes proprietary tagging format.
*/
/*
* Marvell (Ethertype) Distributed Switch Architecture proprietary tagging format.
*/
```

Found in path(s):

* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/dlt.h

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2006 Paolo Abeni (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
 FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* Basic USB data struct
* By Paolo Abeni <paolo.abeni@email.it>
*/
```

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/usb.h
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2002 - 2005 NetGroup, Politecnico di Torino (Italy)
* Copyright (c) 2005 - 2009 CACE Technologies, Inc. Davis (California)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
```

*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
 AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/pcap-inttypes.h
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   This product includes software developed by the University of
*   California, Berkeley and its contributors.
* 4. Neither the name of the University
 nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.

```
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/vlan.h
No license file was found, but licenses were detected in source scan.

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><title>Npcap
Development Tutorial</title><meta name="generator" content="DocBook XSL Stylesheets V1.79.2"><meta
name="description" content="A step-by-step guide to writing software that uses Npcap to list network adapters,
capture packets, and send network traffic."><link rel="home" href="index.html" title="Npcap Reference
Guide"><link rel="up" href="index.html" title="Npcap Reference Guide"><link rel="prev" href="npcap-api.html"
title="The Npcap API"><link rel="next" href="npcap-internals.html" title="Npcap internals"></head><body
bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF"><div class="navheader"><table
width="100%" summary="Navigation header"><tr><th colspan="3" align="center">Npcap Development
Tutorial</th></tr><tr><td width="20%" align="left"><a accesskey="p" href="npcap-api.html">Prev</a></td><th
 width="60%" align="center"></th><td width="20%" align="right"><a accesskey="n" href="npcap-
internals.html">Next</a></td></tr></table><hr></div><div class="sect1"><div class="titlepage"><div><div><h2
class="title" style="clear: both"><a name="npcap-tutorial"></a>Npcap Development
Tutorial</h2></div><div><div class="abstract"><p class="title"><b>Abstract</b></p>
    <p>A step-by-step guide to writing software that uses Npcap to list
      network adapters, capture packets, and send network traffic.</p>
  </div></div></div></div>



 <p>This section shows how to use the features of the Npcap API. It is
   organized as a tutorial, subdivided into a set of lessons that will
   introduce the reader, in a step-by-step fashion, to program development
   using Npcap, from the basic functions (obtaining the adapter list,
   starting a capture, etc.) to the most advanced ones (handling send queues
   and gathering statistics about network traffic).</p>



 <p>The samples are
```

```
 written in plain C, so a basic knowledge of C
  programming is required. Also, since this is a tutorial about a library
  dealing with "raw" networking packets, good knowledge of networks and
  network protocols is assumed.</p>

 <p>The code in this section is copied from the <a class="xref" href="npcap-devguide.html#npcap-examples"
title="Examples">the section called &#8220;Examples&#8221;</a> in the source
  distribution and the SDK. The code is released under a BSD-3-clause license and
  copyright: NetGroup, Politecnico di Torino (Italy); CACE Technologies,
  Davis (California); and Insecure.com, LLC. Full text of the code license
  can be found in each source file.</p>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
devlist"></a>Obtaining the device list</h3></div></div></div>

  <p>Typically, the first thing that a Npcap-based application does is
    get a list of attached network adapters. Both libpcap and Npcap
 provide
    the <a class="ulink" href="wpcap/pcap_findalldevs.html" target="_top">pcap_findalldevs_ex()</a> function for
this purpose:
    this function returns a linked list of <code class="literal">pcap_if</code> structures, each of which contains
    comprehensive information about an attached adapter. In particular, the
    fields <code class="literal">name</code> and <code class="literal">description</code> contain the name and a
    human readable description, respectively, of the corresponding
    device.</p>

  <p>The following code retrieves the adapter list and shows it on the
    screen, printing an error if no adapters are found.</p>

  <pre class="programlisting">
#include "pcap.h"

main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int i=0;
 char errbuf[PCAP_ERRBUF_SIZE];

 /* Retrieve the device list from the local machine */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL /* auth is not needed */,
  &amp;alldevs, errbuf) == -1)
 {
  fprintf(stderr,
    "Error in pcap_findalldevs_ex: %s\n",
    errbuf);
  exit(1);
```

```
 }

 /* Print the list */
 for(d= alldevs; d != NULL; d= d-&gt;next)
 {
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
 }

 if (i == 0)
 {
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return;
 }

 /* We don't need any more the device list. Free it */
 pcap_freealldevs(alldevs);
}
</pre>

   <p>Some comments about this code.</p>

   <p>First of all, <a class="ulink" href="./wpcap/pcap_findalldevs.html" target="_top">pcap_findalldevs_ex()</a>,
like
     other libpcap functions, has an <code class="literal">errbuf</code> parameter. This
     parameter points to a string filled by libpcap with a description of the
     error if something goes wrong.</p>

   <p>Second, remember that not all the OSes supported by libpcap provide a
     description
 of the network interfaces, therefore if we want to write a
     portable application, we must consider the case in which
     <code class="literal">description</code> is null: we print the string "No
     description available" in that situation.</p>

   <p>Note finally that we free the list with <a class="ulink" href="wpcap/pcap_findalldevs.html"
target="_top">pcap_freealldevs()</a> once when
     we have finished with it.</p>

   <p>Assuming we have compiled the program, let's try to run it. On a
     particular Windows workstation, the result we optained is</p>

   <pre class="screen">
1. \Device\NPF_{4E273621-5161-46C8-895A-48D0E52A0B83} (Realtek RTL8029(AS) Ethernet Adapter)
2. \Device\NPF_{5D24AE04-C486-4A96-83FB-8B5EC6C7F430} (3Com EtherLink PCI)</pre>
```

As you can see, the name of the network adapters (that will be passed
to libpcap when opening the devices) under Windows are quite unreadable,
so the parenthetical descriptions can be very helpful.

<h3 class="title">Obtaining advanced information about installed devices</h3>

Lesson 1 (the section called “Obtaining the device list”) demonstrated how
to get basic information (i.e. device name and description) about
available adapters. Actually, Npcap provides also other advanced
information. In particular, every <code class="literal">pcap_if</code> structure
returned by pcap_findalldevs_ex()
contains also a list of <code class="literal">pcap_addr</code> structures,
with:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">a list of addresses for that interface.
<li class="listitem">a list of netmasks (each of which corresponds to an entry in
the addresses list).
<li class="listitem">a list of broadcast addresses (each of which corresponds to an
entry in the addresses list).
<li class="listitem">a list of destination addresses (each of which corresponds to
an entry in the addresses list).

Additionally, <code class="literal">pcap_findalldevs_ex()</code> can also
return remote adapters and a list of pcap files that are located in a
given local folder.

The following sample provides an ifprint() function that prints the
complete contents of a <code class="literal">pcap_if</code> structure. It is
invoked by the program for every entry returned by
<code class="literal">pcap_findalldevs_ex()</code>.

<pre class="programlisting">
/* Print all the available information on the given interface */
void ifprint(pcap_if_t *d)
{
 pcap_addr_t *a;
  char ip6str[128];

 /* Name */
 printf("%s\n",d->name);

 /* Description */
 if (d->description)
  printf("\tDescription: %s\n",d->description);

```
/* Loopback Address*/
 printf("\tLoopback: %s\n",(d-&gt;flags &amp; PCAP_IF_LOOPBACK)?"yes":"no");

 /* IP addresses */
 for(a=d-&gt;addresses;a;a=a-&gt;next) {
  printf("\tAddress Family: #%d\n",a-&gt;addr-&gt;sa_family);

  switch(a-&gt;addr-&gt;sa_family)
  {
    case AF_INET:
      printf("\tAddress Family Name: AF_INET\n");
      if (a-&gt;addr)
        printf("\tAddress: %s\n",iptos(((struct sockaddr_in *)a-&gt;addr)-&gt;sin_addr.s_addr));
      if (a-&gt;netmask)
        printf("\tNetmask: %s\n",iptos(((struct sockaddr_in *)a-&gt;netmask)-&gt;sin_addr.s_addr));
      if (a-&gt;broadaddr)
        printf("\tBroadcast Address: %s\n",iptos(((struct sockaddr_in *)a-&gt;broadaddr)-&gt;sin_addr.s_addr));
      if (a-&gt;dstaddr)
        printf("\tDestination Address: %s\n",iptos(((struct sockaddr_in
 *)a-&gt;dstaddr)-&gt;sin_addr.s_addr));
      break;

    case AF_INET6:
      printf("\tAddress Family Name: AF_INET6\n");
      if (a-&gt;addr)
        printf("\tAddress: %s\n", ip6tos(a-&gt;addr, ip6str, sizeof(ip6str)));
     break;

    default:
      printf("\tAddress Family Name: Unknown\n");
      break;
  }
 }
 printf("\n");
}
</pre>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
openadapter"></a>Opening an adapter and capturing the packets</h3></div></div></div>


   <p>Now that we've seen how to obtain an adapter to play with, let's
    start the real job, opening an adapter and capturing some traffic. In
    this lesson we'll write a program that prints some information about each
    packet flowing through the adapter.</p>
```

The function that opens a capture device is pcap_open(). The parameters,
<code class="literal">snaplen</code>, <code class="literal">flags</code> and
<code class="literal">to_ms</code> deserve some explanation.

<code class="literal">snaplen</code> specifies the portion of the packet to capture. On some OSes (like xBSD and Win32), the packet driver can be configured to capture only the initial part of any packet: this decreases the amount of data to copy to the application and therefore improves the efficiency of the capture. In this case we use the value 65536 which is higher than the greatest MTU that we could encounter. In this manner we ensure that the application will always receive the whole packet.

<code class="literal">flags:</code> the most important flag is the one that indicates if the adapter will be put in promiscuous mode. In normal operation, an adapter only captures packets from the network that are destined to it; the packets exchanged by other hosts are therefore ignored. Instead, when the adapter is in promiscuous mode it captures all packets whether they are destined to it or not. This means that on shared media (like non-switched Ethernet), Npcap will be able to capture the packets of other hosts. Promiscuous mode is the default for most capture applications, so we enable it in the following example.

<code class="literal">to_ms</code> specifies the read timeout, in milliseconds. A read on the adapter (for example, with pcap_dispatch() or pcap_next_ex()) will always return after <code class="literal">to_ms</code> milliseconds, even if no packets are available from the network. <code class="literal">to_ms</code> also defines the interval between statistical reports if the adapter is in statistical mode (see the lesson "\ref wpcap_tut9" for information about statistical mode). Setting <code class="literal">to_ms</code> to 0 means no timeout, a read on the adapter never returns if no packets arrive. A -1 timeout on the other side causes a read on the adapter to always return immediately.

<pre class="programlisting">
#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

/* prototype of the packet handler */
void packet_handler(
 u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);

```
int main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Retrieve the device list on the local machine */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL, &amp;alldevs, errbuf) == -1)
 {
  fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
  exit(1);
 }

 /* Print
 the list */
 for(d=alldevs; d; d=d-&gt;next)
 {
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
 }

 if(i==0)
 {
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
 }

 printf("Enter the interface number (1-%d):",i);
 scanf_s("%d", &amp;inum);

 if(inum &lt; 1 || inum &gt; i)
 {
  printf("\nInterface number out of range.\n");
  /* Free the device list */
```

```
  pcap_freealldevs(alldevs);
  return -1;
 }

 /* Jump to the selected adapter */
 for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);

 /* Open the device */
 if ( (adhandle= pcap_open(d-&gt;name, // name of the device
        65536, // portion of the packet to capture
               // 65536 guarantees that the whole packet will
               // be captured on all the link layers
        PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode

      1000, // read timeout
        NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
 {
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
 }

 printf("\nlistening on %s...\n", d-&gt;description);

 /* At this point, we don't need any more the device list. Free it */
 pcap_freealldevs(alldevs);

 /* start the capture */
 pcap_loop(adhandle, 0, packet_handler, NULL);

 return 0;
}


/* Callback function invoked by libpcap for every incoming packet */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct tm ltime;
 char timestr[16];
 time_t local_tv_sec;
```

```
 /*
  * unused variables
  */
 (VOID)(param);
 (VOID)(pkt_data);

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime,
 &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 printf("%s,%.6d len:%d\n",
   timestr, header-&gt;ts.tv_usec, header-&gt;len);
}
</pre>

   <p>Once the adapter is opened, the capture can be started with <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_dispatch()</a> or <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_loop()</a>. These two functions are
     very similar, the difference is that <code class="literal">pcap_dispatch()</code>
     returns (although not guaranteed) when the timeout expires while
     <code class="literal">pcap_loop()</code> doesn't return until
     <code class="literal">cnt</code> packets have been captured, so it can block for an
     arbitrary period on an under-utilized network.
     <code class="literal">pcap_loop()</code> is enough for the purpose of this sample,
     while <code class="literal">pcap_dispatch()</code> is normally used in a more
     complex program.</p>

    <p>Both of these functions have a <code class="literal">callback</code> parameter,
     <code class="literal">packet_handler</code>, pointing to a function that will
     receive the packets. This function is invoked by libpcap for every new
     packet coming from the network and receives a generic status
     (corresponding to the <code class="literal">user</code> parameter of <a class="ulink"
href="wpcap/pcap_loop.html" target="_top">pcap_loop()</a> and <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_dispatch()</a>), a header with some
     information on the packet like the timestamp and the length and the
     actual data of the packet including all the protocol headers. Note that
     the frame CRC is normally not present, because it is removed by the
     network adapter after frame validation. Note also that most adapters
     discard packets with wrong CRCs, therefore Npcap is normally not able
     to capture them.</p>

   <p>The above example
 extracts the timestamp and the length of every
     packet from the <code class="literal">pcap_pkthdr</code> header and prints them on
     the screen.</p>
```

Please note that there may be a drawback using pcap_loop() mainly related to the
fact that the handler is called by the packet capture driver; therefore
the user application does not have direct control over it. Another
approach (and to have more readable programs) is to use the pcap_next_ex() function, which is
presented in the next example (the section called “Capturing the packets without the callback”).

<h3 class="title">Capturing
the packets without the callback</h3>

The example program in this lesson behaves exactly like the previous
program (the section called “Opening an adapter and capturing the packets”), but it uses
pcap_next_ex() instead of
pcap_loop().

The callback-based capture mechanism of pcap_loop() is elegant and it could
be a good choice in some situations. However, handling a callback is
sometimes not practical—it often makes the program more complex
especially in situations with multithreaded applications or C++
classes.

In these cases, pcap_next_ex() retrievs a packet
with a direct call—using <code class="literal">pcap_next_ex()</code>,
packets are received only when the programmer wants them.

The parameters of this function are the same as a capture callback.
It takes an adapter descriptor and a couple of pointers that will be
initialized and returned to the user (one to a
<code class="literal">pcap_pkthdr</code> structure and another to a buffer with the
packet data).

In the following program, we recycle the callback code of the
previous lesson's example and move it inside main() right after the call
to pcap_next_ex().

<pre class="programlisting">
/* Open the device */
if ((adhandle= pcap_open(d->name, // name of the device

```
    65536, // portion of the packet to capture.
           // 65536 guarantees that the whole
 packet will
           // be captured on all the link layers
    PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
    1000, // read timeout
    NULL, // authentication on the remote machine
    errbuf // error buffer
 ) ) == NULL)
{
 fprintf(stderr,
  "\nUnable to open the adapter. %s is not supported by Npcap\n",
  d-&gt;name);
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}

printf("\nlistening on %s...\n", d-&gt;description);

/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);

/* Retrieve the packets */
while((res = pcap_next_ex( adhandle, &amp;header, &amp;pkt_data)) &gt;= 0){

 if(res == 0)
  /* Timeout elapsed */
  continue;

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 printf("%s,%.6d len:%d\n", timestr, header-&gt;ts.tv_usec, header-&gt;len);
}

if(res
 == -1){
 printf("Error reading the packets: %s\n", pcap_geterr(adhandle));
 return -1;
}
</pre>

  <p>Why do we use <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> instead
of the old
    <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_next()</a>? Because
```

<code class="literal">pcap_next()</code> has some drawbacks. First of all, it is
inefficient because it hides the callback method but still relies on
<code class="literal">pcap_dispatch()</code>. Second, it is not able to detect EOF,
so it's not very useful when gathering packets from a file.

Notice also that <code class="literal">pcap_next_ex()</code> returns different
values for success, timeout elapsed, error and EOF conditions.

<h3 class="title">Filtering the traffic</h3>

One of the most
powerful features offered by Npcap (and by libpcap as
well) is the filtering engine. It provides a very efficient way to
receive subsets of the network traffic, and is (usually) integrated with
the capture mechanism provided by Npcap. The functions used to filter
packets are pcap_compile()
and pcap_setfilter().

pcap_compile() takes a
string containing a high-level Boolean (filter) expression and produces a
low-level byte code that can be interpreted by the fileter engine in the
packet driver. The syntax of the boolean expression can be found in the
Filtering expression syntax
section of this documentation.

<a class="ulink" href="wpcap/pcap_setfilter.html"
target="_top">pcap_setfilter()
associates a filter with a capture session in the kernel driver. Once
<code class="literal">pcap_setfilter()</code> is called, the associated filter will
be applied to all the packets coming from the network, and all the
conformant packets (i.e., packets for which the Boolean expression
evaluates to true) will be actually copied to the application.

The following code shows how to compile and set a filter. Note that
we must retrieve the netmask from the <code class="literal">pcap_if</code>
structure that describes the adapter, because some filters created by
<code class="literal">pcap_compile()</code> require it.

The filter passed to <code class="literal">pcap_compile()</code> in this code
snippet is "ip and tcp", which means to "keep only the packets that are
both IPv4 and TCP and deliver them to the application".

<pre class="programlisting">
if

```
 (d-&gt;addresses != NULL)
 /* Retrieve the mask of the first address of the interface */
 netmask=((struct sockaddr_in *)(d-&gt;addresses-&gt;netmask))-&gt;sin_addr.S_un.S_addr;
else
 /* If the interface is without an address
  * we suppose to be in a C class network */
 netmask=0xffffff;


//compile the filter
if (pcap_compile(adhandle, &amp;fcode, "ip and tcp", 1, netmask) &lt; 0)
{
 fprintf(stderr,
   "\nUnable to compile the packet filter. Check the syntax.\n");
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}

//set the filter
if (pcap_setfilter(adhandle, &amp;fcode) &lt; 0)
{
 fprintf(stderr,"\nError setting the filter.\n");
 /* Free the device list */
 pcap_freealldevs(alldevs);
 return -1;
}
</pre>

   <p>If you want to see some code that uses the filtering functions shown
     in this lesson, look at the example presented in the next Lesson, <a class="xref" href="npcap-
tutorial.html#npcap-tutorial-interpreting" title="Interpreting the
 packets">the section called &#8220;Interpreting the packets&#8221;</a>.</p>

 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
interpreting"></a>Interpreting the packets</h3></div></div></div>


   <p>Now that we are able to capture and filter network traffic, we want
     to put our knowledge to work with a simple "real world"
     application.</p>

   <p>In this lesson we will take code from the previous lessons and use
     these pieces to build a more useful program. the main purpose of the
     current program is to show how the protocol headers of a captured packet
     can be parsed and interpreted. The resulting application, called UDPdump,
```

prints a summary of the UDP traffic on our network.

We have chosen to parse and display the UDP protocol because it is more accessible than other protocols such as TCP and consequently is an excellent initial example. Let's look at the code:

<pre class="programlisting">
#include <pcap.h>
#include <Winsock2.h>
#include <tchar.h>
BOOL LoadNpcapDlls()
{
 _TCHAR npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory: %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, _T("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
  return FALSE;
 }
 return TRUE;
}

/* 4 bytes IP address */
typedef struct ip_address{
 u_char byte1;
 u_char byte2;
 u_char byte3;
 u_char byte4;
}ip_address;

/* IPv4 header */
typedef struct ip_header{
 u_char ver_ihl; // Version (4 bits) + IP header length (4 bits)
 u_char tos;   // Type of service
 u_short tlen;   // Total length
 u_short identification; // Identification
 u_short flags_fo; // Flags (3 bits) + Fragment offset (13 bits)
 u_char ttl;   // Time to live
 u_char proto;
   // Protocol
 u_short crc;   // Header checksum
 ip_address saddr; // Source address

```
 ip_address  daddr; // Destination address
 u_int  op_pad;     // Option + Padding
}ip_header;

/* UDP header*/
typedef struct udp_header{
 u_short sport; // Source port
 u_short dport; // Destination port
 u_short len;   // Datagram length
 u_short crc;   // Checksum
}udp_header;

/* prototype of the packet handler */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);


int main()
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];
 u_int netmask;
 char packet_filter[] = "ip and udp";
 struct bpf_program fcode;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 /* Retrieve the device list */
 if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
    NULL, &amp;alldevs, errbuf) == -1)
 {
  fprintf(stderr,"Error
 in pcap_findalldevs: %s\n", errbuf);
  exit(1);
 }

 /* Print the list */
 for(d=alldevs; d; d=d-&gt;next)
```

```
{
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
}

if(i==0)
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);

/* Open the adapter */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
            65536, // portion of the packet to capture.
                  // 65536 grants that the whole packet
                  // will be captured
on all the MACs.
            PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
            1000, // read timeout
            NULL, // remote authentication
            errbuf // error buffer
            ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}
```

```
/* Check the link layer. We support only Ethernet for simplicity. */
if(pcap_datalink(adhandle) != DLT_EN10MB)
{
  fprintf(stderr,"\nThis program works only on Ethernet networks.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

if(d-&gt;addresses != NULL)
  /* Retrieve the mask of the first address of the interface */
  netmask=((struct sockaddr_in *)(d-&gt;addresses-&gt;netmask))-&gt;sin_addr.S_un.S_addr;
else
  /* If the interface is without addresses
   * we suppose to be in a C class network */
  netmask=0xffffff;



//compile the filter
if (pcap_compile(adhandle, &amp;fcode, packet_filter, 1, netmask) &lt;0 )
{
  fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

//set the filter
if (pcap_setfilter(adhandle, &amp;fcode)&lt;0)
{
  fprintf(stderr,"\nError setting the filter.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

printf("\nlistening on %s...\n", d-&gt;description);

/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);

/* start the capture */
pcap_loop(adhandle, 0, packet_handler, NULL);

return 0;
}
```

```
/* Callback function invoked by libpcap for every incoming packet */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct tm ltime;
 char timestr[16];
 ip_header *ih;
 udp_header *uh;
 u_int
 ip_len;
 u_short sport,dport;
 time_t local_tv_sec;

 /*
  * Unused variable
  */
 (VOID)(param);

 /* convert the timestamp to readable format */
 local_tv_sec = header-&gt;ts.tv_sec;
 localtime_s(&amp;ltime, &amp;local_tv_sec);
 strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

 /* print timestamp and length of the packet */
 printf("%s.%.6d len:%d ", timestr, header-&gt;ts.tv_usec, header-&gt;len);

 /* retireve the position of the ip header */
 ih = (ip_header *) (pkt_data +
   14); //length of ethernet header

 /* retireve the position of the udp header */
 ip_len = (ih-&gt;ver_ihl &amp; 0xf) * 4;
 uh = (udp_header *) ((u_char*)ih + ip_len);

 /* convert from network byte order to host byte order */
 sport = ntohs( uh-&gt;sport );
 dport = ntohs( uh-&gt;dport );

 /* print ip addresses and udp ports */
 printf("%d.%d.%d.%d.%d -&gt; %d.%d.%d.%d.%d\n",
   ih-&gt;saddr.byte1,
   ih-&gt;saddr.byte2,
   ih-&gt;saddr.byte3,
   ih-&gt;saddr.byte4,

   sport,
   ih-&gt;daddr.byte1,
```

```
  ih-&gt;daddr.byte2,
  ih-&gt;daddr.byte3,
  ih-&gt;daddr.byte4,
  dport);
}
</pre>

  <p>First of all, we set the filter to "ip and udp". In this way we are
    sure that packet_handler() will receive only UDP packets over IPv4: this
    simplifies the parsing and increases the efficiency of the
    program.</p>

  <p>We have also created a couple of structs that describe the IP and UDP
    headers. These structs are used by packet_handler() to properly locate
    the various header fields.</p>

  <p>packet_handler(), although limited to a single protocol dissector
    (UDP over IPv4), shows how complex "sniffers" like tcpdump/WinDump decode
    the network traffic. Since we aren't interested in the MAC header, we
    skip it. For simplicity and before starting the capture, we check the MAC
    layer with <a class="ulink" href="wpcap/pcap_datalink.html" target="_top">pcap_datalink()</a>
    to make sure that we are dealing
 with an Ethernet network. This way we
    can be sure that the MAC header is exactly 14 bytes.</p>

  <p>The IP header is located just after the MAC header. We will extract
    the IP source and destination addresses from the IP header.</p>

  <p>Reaching the UDP header is a bit more complicated, because the IP
    header doesn't have a fixed length. Therefore, we use the IP header's
    length field to know its size. Once we know the location of the UDP
    header, we extract the source and destination ports.</p>

  <p>The extracted values are printed on the screen, and the result is
    something like:</p>
  <pre class="screen">
\Device\Packet_{A7FD048A-5D4B-478E-B3C1-34401AC3B72F} (Xircom t 10/100 Adapter)
Enter the interface number (1-2):1
listening on Xircom CardBus Ethernet 10/100 Adapter...
16:13:15.312784 len:87 130.192.31.67.2682 -&gt; 130.192.3.21.53
16:13:15.314796 len:137 130.192.3.21.53 -&gt; 130.192.31.67.2682
16:13:15.322101 len:78 130.192.31.67.2683
 -&gt; 130.192.3.21.53</pre>
  <p>Each of the final 3 lines represents a different packet.</p>
 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
offline"></a>Handling offline dump files</h3></div></div></div>
```

In this lession we are going to learn how to handle packet capture to
a file (dump to file). Npcap offers a wide range of functions to save
the network traffic to a file and to read the content of
dumps—this lesson will teach how to use all of these
functions.

The format for dump files is the libpcap one. This format contains
the data of the captured packets in binary form and is a standard used by
many network tools including WinDump, Wireshark and Snort.

<h4 class="title">Saving packets to a dump file</h4>

First
of all, let's see how to write packets in libpcap
format.

The following example captures the packets from the selected
interface and saves them on a file whose name is provided by the
user.
<pre class="programlisting">
#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

/* prototype of the packet handler */
void packet_handler(u_char *param,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data);

int main(int argc, char **argv)
{
 pcap_if_t *alldevs;
 pcap_if_t *d;
 int inum;
 int i=0;
 pcap_t *adhandle;
 char errbuf[PCAP_ERRBUF_SIZE];
 pcap_dumper_t *dumpfile;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

```
/* Check command line */
if(argc != 2)
{
  printf("usage: %s filename", argv[0]);
  return -1;
}

/* Retrieve the device list on the local machine */
if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
  NULL, &amp;amp;alldevs,
errbuf) == -1)
{
  fprintf(stderr,"Error in pcap_findalldevs: %s\n", errbuf);
  exit(1);
}

/* Print the list */
for(d=alldevs; d; d=d-&gt;next)
{
  printf("%d. %s", ++i, d-&gt;name);
  if (d-&gt;description)
    printf(" (%s)\n", d-&gt;description);
  else
    printf(" (No description available)\n");
}

if(i==0)
{
  printf("\nNo interfaces found! Make sure Npcap is installed.\n");
  return -1;
}

printf("Enter the interface number (1-%d):",i);
scanf_s("%d", &amp;amp;inum);

if(inum &lt; 1 || inum &gt; i)
{
  printf("\nInterface number out of range.\n");
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Jump to the selected adapter */
for(d=alldevs, i=0; i&lt; inum-1 ;d=d-&gt;next, i++);
```

```
/* Open the device */
if ( (adhandle= pcap_open(d-&gt;name, // name of the device
            65536, // portion of the packet to capture
                // 65536 guarantees that
the whole packet
                // will be captured on all the link layers
            PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
            1000, // read timeout
            NULL, // authentication on the remote machine
            errbuf // error buffer
            ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    d-&gt;name);
  /* Free the device list */
  pcap_freealldevs(alldevs);
  return -1;
}

/* Open the dump file */
dumpfile = pcap_dump_open(adhandle, argv[1]);

if(dumpfile==NULL)
{
  fprintf(stderr,"\nError opening output file\n");
  return -1;
}

printf("\nlistening on %s... Press Ctrl+C to stop...\n", d-&gt;description);

/* At this point, we no longer need the device list. Free it */
pcap_freealldevs(alldevs);

/* start the capture */
pcap_loop(adhandle, 0, packet_handler, (unsigned char *)dumpfile);

return 0;
}

/* Callback function invoked
 by libpcap for every incoming packet */
void packet_handler(u_char *dumpfile,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 /* save the packet on the dump file */
 pcap_dump(dumpfile, header, pkt_data);
```

```
}
</pre>

    <p>As you can see, the structure of the program is very similar to the
      ones we have seen in the previous lessons. The differences
      are:</p>

    <div class="itemizedlist"><ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">a call to <a class="ulink" href="wpcap/pcap_dump_open.html" target="_top">pcap_dump_open()</a> is issued
        once the interface is opened. This call opens a dump file and
        associates it with the interface.</li><li class="listitem">the packets are written to this file with a <a class="ulink" href="wpcap/pcap_dump.html" target="_top">pcap_dump()</a> from the
        packet_handler() callback. The parameters of
        <code class="literal">pcap_dump()</code> are in 1-1
 correspondence with the
        parameters of <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_handler()</a>.</li></ul></div>
  </div>

  <div class="sect3"><div class="titlepage"><div><div><h4 class="title"><a name="npcap-tutorial-offline-reading"></a>Reading packets from a dump file</h4></div></div></div>


    <p>Now that we have a dump file available, we can try to read its
      content. The following code opens a Npcap/libpcap dump file and
      displays every packet contained in the file. The file is opened with
      <a class="ulink" href="wpcap/pcap_open_offline.html" target="_top">pcap_open_offline()</a>,
      then the usual <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_loop()</a> is
      used to sequence through the packets. As you can see, reading packets
      from an offline capture is nearly identical to receiving them from a
      physical interface.</p>

    <p>This example introduces another function:
       <code class="literal">pcap_createsrcstr()</code>. This function is required to
      create a source string that begins with a marker used to tell Npcap the
      type of the source, e.g. "rpcap://" if we are going to open an adapter,
      or "file://" if we are going to open a file. This step is not required
      when <code class="literal">pcap_findalldevs_ex()</code> is used (the returned
      values already contain these strings). However, it is required in this
      example because the name of the file is read from the user
      input.</p>

    <pre class="programlisting">
#include &lt;stdio.h&gt;
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

#define LINE_LEN 16
```

```
void dispatcher_handler(u_char *,
 const struct pcap_pkthdr *,
 const u_char *);

int main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 char source[PCAP_BUF_SIZE];

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr,
 "Couldn't load Npcap\n");
  exit(1);
 }

 if(argc != 2){

  printf("usage: %s filename", argv[0]);
  return -1;

 }

 /* Create the source string according to the new Npcap syntax */
 if ( pcap_createsrcstr(  source, // variable that will keep the source string
           PCAP_SRC_FILE,  // we want to open a file
           NULL,      // remote host
           NULL,      // port on the remote host
           argv[1],    // name of the file we want to open
           errbuf      // error buffer
           ) != 0)
 {
  fprintf(stderr,"\nError creating a source string\n");
  return -1;
 }

 /* Open the capture file */
 if ( (fp= pcap_open(source, // name of the device
         65536, // portion of the packet to capture
              // 65536 guarantees that the whole packet
              // will be captured on all the link layers
           PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
           1000, // read timeout
```

```
  NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
 {
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return -1;
 }

 // read and dispatch packets until EOF is reached
 pcap_loop(fp, 0, dispatcher_handler, NULL);

 return 0;
}


void dispatcher_handler(u_char *temp1,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 u_int i=0;

 /*
  * Unused variable
  */
 (VOID)temp1;

 /* print pkt timestamp and pkt len */
 printf("%ld:%ld (%ld)\n", header-&gt;ts.tv_sec, header-&gt;ts.tv_usec, header-&gt;len);

 /* Print the packet */
 for (i=1; (i &lt; header-&gt;caplen + 1 ) ; i++)
 {
  printf("%.2x ", pkt_data[i-1]);
  if ( (i % LINE_LEN) == 0) printf("\n");
 }

 printf("\n\n");

}
</pre>

    <p>The following example has the same purpose of the last one, but
      <a class="ulink" href="wpcap/pcap_next_ex.html" target="_top">pcap_next_ex()</a> is used
      instead
 of the <a class="ulink" href="wpcap/pcap_loop.html" target="_top">pcap_loop()</a>
      callback method.</p>

    <pre class="programlisting">
```

```
#include &lt;stdio.h&gt;
#include &lt;pcap.h&gt;
#include "misc.h" /* LoadNpcapDlls */

#define LINE_LEN 16

int main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 char source[PCAP_BUF_SIZE];
 struct pcap_pkthdr *header;
 const u_char *pkt_data;
 u_int i=0;
 int res;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }

 if(argc != 2)
 {
  printf("usage: %s filename", argv[0]);
  return -1;
 }

 /* Create the source string according to the new Npcap syntax */
 if ( pcap_createsrcstr(  source, // variable that will keep the source string
          PCAP_SRC_FILE,  // we want to open a file
          NULL,      // remote host
          NULL,      // port on the remote host
          argv[1],    // name of
 the file we want to open
          errbuf      // error buffer
          ) != 0)
 {
  fprintf(stderr,"\nError creating a source string\n");
  return -1;
 }

 /* Open the capture file */
 if ( (fp= pcap_open(source, // name of the device
        65536, // portion of the packet to capture
              // 65536 guarantees that the whole packet
              // will be captured on all the link layers
```

```
        PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode
        1000, // read timeout
        NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
{
  fprintf(stderr,"\nUnable to open the file %s.\n", source);
  return -1;
}

/* Retrieve the packets from the file */
while((res = pcap_next_ex( fp, &amp;header, &amp;pkt_data)) &gt;= 0)
{
  /* print pkt timestamp and pkt len */
  printf("%ld:%ld (%ld)\n", header-&gt;ts.tv_sec, header-&gt;ts.tv_usec, header-&gt;len);


  /* Print the packet */
  for (i=1; (i &lt; header-&gt;caplen + 1 ) ; i++)
  {
    printf("%.2x ", pkt_data[i-1]);
    if ( (i % LINE_LEN) == 0) printf("\n");
  }

  printf("\n\n");
}


if (res == -1)
{
  printf("Error reading the packets: %s\n", pcap_geterr(fp));
}

return 0;
}
</pre>

  </div>

</div>

<div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
sending"></a>Sending Packets</h3></div></div></div>

  <p>Although the name <span class="emphasis"><em>Npcap</em></span> indicates clearly that the purpose
    of the library is packet capture, other useful features for raw
    networking are provided. Among them, the user can find a complete set of
    functions to send packets.</p>
```

<h4 class="title">Sending a single packet with <code class="literal">pcap_sendpacket()</code></h4>

The simplest way to send a packet is shown in the following code snippet. After opening an adapter, pcap_sendpacket() is called to send a hand-crafted packet. <code class="literal">pcap_sendpacket()</code> takes as arguments a buffer containing the data to send, the length of the buffer and the adapter that will send it. Notice that the buffer is sent to the net as is, without any manipulation. This means that the application has to create the correct protocol headers in order to send something meaningful.

<pre class="programlisting">
#include <stdlib.h>
#include <stdio.h>

#include <pcap.h>
#include "misc.h" /* LoadNpcapDlls */

void main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 u_char packet[100];
 int i;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't
 load Npcap\n");
  exit(1);
 }

 /* Check the validity of the command line */
 if (argc != 2)
 {
  printf("usage: %s interface (e.g. 'rpcap://eth0')", argv[0]);
  return;
 }

 /* Open the output device */
 if ((fp= pcap_open(argv[1], // name of the device
         100, // portion of the packet to capture
         PCAP_OPENFLAG_PROMISCUOUS, // promiscuous mode

```
        1000, // read timeout
        NULL, // authentication on the remote machine
        errbuf // error buffer
        ) ) == NULL)
{
  fprintf(stderr,
    "\nUnable to open the adapter. %s is not supported by Npcap\n",
    argv[1]);
  return;
}

/* Supposing to be on ethernet, set mac destination to 1:1:1:1:1:1 */
packet[0]=1;
packet[1]=1;
packet[2]=1;
packet[3]=1;
packet[4]=1;
packet[5]=1;

/* set mac source to 2:2:2:2:2:2 */
packet[6]=2;
packet[7]=2;
packet[8]=2;
packet[9]=2;
packet[10]=2;
packet[11]=2;

/* Fill the rest of the packet
*/
for(i=12;i&lt;100;i++)
{
  packet[i]=(u_char)i;
}

/* Send down the packet */
if (pcap_sendpacket(fp, packet, 100 /* size */) != 0)
{
  fprintf(stderr,"\nError sending the packet: %s\n", pcap_geterr(fp));
  return;
}

return;
}
</pre>

  </div>

  <div class="sect3"><div class="titlepage"><div><div><h4 class="title"><a name="id582137"></a>Send
```

queues</h4>

While pcap_sendpacket()
offers a simple and immediate way to send a single packet,
send queues provide an advanced, powerful and
optimized mechanism to send a collection of packets. A send queue is a
container for a variable number of packets that will be sent to the
network. It has a size, that represents the maximum amount of bytes it
can store.

A send queue is created calling the
<code class="literal">pcap_sendqueue_alloc()</code>
function, specifying the size
of the new send queue.

Once the send queue is created,
<code class="literal">pcap_sendqueue_queue()</code> can be used to add a packet
to the send queue. This function takes a <code class="literal">pcap_pkthdr</code>
with the timestamp and the length and a buffer with the data of the
packet. These parameters are the same as those received by <a class="ulink" href="wpcap/pcap_next_ex.html"
target="_top">pcap_next_ex() and
<code class="literal">pcap_handler()</code>, therefore queuing a packet that was
just captured or read from a file is a matter of passing these
parameters to <code class="literal">pcap_sendqueue_queue()</code>.

To transmit a send queue, Npcap provides the
<code class="literal">pcap_sendqueue_transmit()</code> function. Note the third
parameter: if nonzero, the send will be
synchronized,
i.e. the relative timestamps of the
packets will be respected. This operation requires a remarkable amount
of CPU, because the synchronization takes place in the kernel driver
using "busy wait" loops. Although this operation is quite CPU
intensive, it often results in very high precision packet transmissions
(often around few microseconds or less).

Note that transmitting a send queue with
<code class="literal">pcap_sendqueue_transmit()</code> is much more efficient
than performing a series of <a class="ulink" href="wpcap/pcap_inject.html"
target="_top">pcap_sendpacket(), because the
send queue is buffered at kernel level drastically decreasing the
number of context switches.

When a queue is no longer needed, it can be deleted with
<code class="literal">pcap_sendqueue_destroy()</code> that frees all the buffers
associated with the

send queue.

The next program shows how to use send queues. It opens a capture
file with pcap_open_offline(), then
it moves the packets from the file to a properly allocated send queue.
At his point it transmits the queue, synchronizing it if requested by
the user.

Note that the link-layer of the dumpfile is compared with the one
of the interface that will send the packets using <a class="ulink" href="wpcap/pcap_datalink.html"
target="_top">pcap_datalink(), and a warning
is printed if they are different—it is important that the
capture-file link-layer be the same as the adapter's link layer for
otherwise the transmission is pointless.

<pre class="programlisting">
#include <stdlib.h>
#include <stdio.h>

#include <pcap.h>

#ifdef _WIN32
#include <tchar.h>
BOOL LoadNpcapDlls()
{
 TCHAR
 npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
  fprintf(stderr, "Error in GetSystemDirectory: %x", GetLastError());
  return FALSE;
 }
 _tcscat_s(npcap_dir, 512, TEXT("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
  fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
  return FALSE;
 }
 return TRUE;
}
#endif

void usage();

void main(int argc, char **argv)
{
 pcap_t *indesc,*outdesc;
 char errbuf[PCAP_ERRBUF_SIZE];

```
 char source[PCAP_BUF_SIZE];
 FILE *capfile;
 int caplen, sync;
 u_int res;
 pcap_send_queue *squeue;
 struct pcap_pkthdr *pktheader;
 u_char *pktdata;
 float cpu_time;
 u_int npacks = 0;
 errno_t fopen_error;

#ifdef _WIN32
 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
  fprintf(stderr, "Couldn't load Npcap\n");
  exit(1);
 }
#endif

 /* Check the validity of the command line */
 if (argc &lt;= 2 || argc &gt;= 5)
 {
  usage();
  return;
 }

 /* Retrieve
 the length of the capture file */
 fopen_error = fopen_s(&amp;capfile, argv[1],"rb");
 if(fopen_error != 0){
  printf("Error opening the file, errno %d.\n", fopen_error);
  return;
 }

 fseek(capfile , 0, SEEK_END);
 caplen= ftell(capfile)- sizeof(struct pcap_file_header);
 fclose(capfile);

 /* Chek if the timestamps must be respected */
 if(argc == 4 &amp;&amp; argv[3][0] == 's')
  sync = TRUE;
 else
  sync = FALSE;

 /* Open the capture */
 /* Create the source string according to the new WinPcap syntax */
 if ( pcap_createsrcstr(
```

```
        source, // variable that will keep the source string
        PCAP_SRC_FILE,  // we want to open a file
        NULL,      // remote host
        NULL,      // port on the remote host
        argv[1],    // name of the file we want to open
        errbuf      // error buffer
        ) != 0)
{
 fprintf(stderr,"\nError creating a source string\n");
 return;
}

/*
Open the capture file */
if ( (indesc= pcap_open(source, 65536, PCAP_OPENFLAG_PROMISCUOUS,
                1000, NULL, errbuf) ) == NULL)
{
 fprintf(stderr,"\nUnable to open the file %s.\n", source);
 return;
}

/* Open the output adapter */
if ( (outdesc= pcap_open(argv[2], 100, PCAP_OPENFLAG_PROMISCUOUS,
                 1000, NULL, errbuf) ) == NULL)
{
 fprintf(stderr,"\nUnable to open adapter %s.\n", source);
 return;
}

/* Check the MAC type */
if (pcap_datalink(indesc) != pcap_datalink(outdesc))
{
 printf("Warning: the datalink of the capture differs"
       " from the one of the selected interface.\n");
 printf("Press a key to continue, or CTRL+C to stop.\n");
 getchar();
}

/* Allocate a send queue */
squeue = pcap_sendqueue_alloc(caplen);

/* Fill the queue with the packets from the file */
while ((res = pcap_next_ex( indesc, &amp;pktheader, &amp;pktdata)) == 1)
{
 if (pcap_sendqueue_queue(squeue,
pktheader, pktdata) == -1)
 {
   printf("Warning: packet buffer too small, not all the packets will be sent.\n");
```

```
    break;
  }

  npacks++;
}

if (res == -1)
{
  printf("Corrupted input file.\n");
  pcap_sendqueue_destroy(squeue);
  return;
}

/* Transmit the queue */

cpu_time = (float)clock ();

if ((res = pcap_sendqueue_transmit(outdesc, squeue, sync)) &lt; squeue-&gt;len)
{
  printf("An error occurred sending the packets: %s."
         " Only %d bytes were sent\n", pcap_geterr(outdesc), res);
}

cpu_time = (clock() - cpu_time)/CLK_TCK;

printf ("\n\nElapsed time: %5.3f\n", cpu_time);
printf ("\nTotal packets generated = %d", npacks);
printf ("\nAverage packets per second = %d", (int)((double)npacks/cpu_time));
printf ("\n");

/* free the send queue */
pcap_sendqueue_destroy(squeue);

/* Close the input file */
pcap_close(indesc);

/*
 * close the output adapter
 * IMPORTANT: remember to close the adapter, otherwise
there will be no
 * guarantee that all the packets will be sent!
 */
pcap_close(outdesc);

return;
}
```

```
void usage()
{

 printf("\nSendcap, sends a libpcap/tcpdump capture file to the net."
        " Copyright (C) 2002 Loris Degioanni.\n");
 printf("\nUsage:\n");
 printf("\t sendcap file_name adapter [s]\n");
 printf("\nParameters:\n");
 printf("\nfile_name: the name of the dump file that will be sent to the network\n");
 printf("\nadapter: the device to use. Use \"WinDump -D\" for a list of valid devices\n");
 printf("\ns: if present, forces the packets to be sent synchronously,"
        " i.e. respecting the timestamps in the dump file.\n\n");

 exit(0);
}
</pre>
   </div>
 </div>

 <div class="sect2"><div class="titlepage"><div><div><h3 class="title"><a name="npcap-tutorial-
statistics"></a>Gathering Statistics on the network traffic</h3></div></div></div>


   <p>This lesson shows another advanced feature of Npcap: the ability to
     collect statistics
 about network traffic. The statistical engine makes
     use of the kernel-level packet filter to efficiently classify the
     incoming packet.</p>

   <p>In order to use this feature, the programmer must open an adapter and
     put it in <span class="emphasis"><em>statistical mode</em></span>. This can be done with
     <code class="literal">pcap_setmode()</code>. In particular,
     <code class="literal">MODE_STAT</code> must be used as the <code class="literal">mode</code>
     argument of this function.</p>

   <p>With statistical mode, making an application that monitors the TCP
     traffic load is a matter of few lines of code. The following sample shows
     how to do it.</p>

   <pre class="programlisting">
#include &lt;stdlib.h&gt;
#include &lt;stdio.h&gt;

#include &lt;pcap.h&gt;

#include &lt;tchar.h&gt;
BOOL LoadNpcapDlls()
{
```

```
 _TCHAR npcap_dir[512];
 UINT len;
 len = GetSystemDirectory(npcap_dir, 480);
 if (!len) {
   fprintf(stderr, "Error in GetSystemDirectory:
 %x", GetLastError());
   return FALSE;
 }
 _tcscat_s(npcap_dir, 512, _T("\\Npcap"));
 if (SetDllDirectory(npcap_dir) == 0) {
   fprintf(stderr, "Error in SetDllDirectory: %x", GetLastError());
   return FALSE;
 }
 return TRUE;
}

void usage();

void dispatcher_handler(u_char *, const struct pcap_pkthdr *, const u_char *);


void main(int argc, char **argv)
{
 pcap_t *fp;
 char errbuf[PCAP_ERRBUF_SIZE];
 struct timeval st_ts;
 u_int netmask;
 struct bpf_program fcode;

 /* Load Npcap and its functions. */
 if (!LoadNpcapDlls())
 {
   fprintf(stderr, "Couldn't load Npcap\n");
   exit(1);
 }

 /* Check the validity of the command line */
 if (argc != 2)
 {
   usage();
   return;
 }

 /* Open the output adapter */
 if ( (fp= pcap_open(argv[1], 100, PCAP_OPENFLAG_PROMISCUOUS,
                 1000, NULL, errbuf) ) == NULL)
 {
   fprintf(stderr,"\nUnable to open adapter %s.\n", errbuf);
```

```
  return;
 }

 /* Don't care about netmask, it
 won't be used for this filter */
 netmask=0xffffff;

 //compile the filter
 if (pcap_compile(fp, &amp;fcode, "tcp", 1, netmask) &lt;0 )
 {
  fprintf(stderr,"\nUnable to compile the packet filter. Check the syntax.\n");
  /* Free the device list */
  return;
 }

 //set the filter
 if (pcap_setfilter(fp, &amp;fcode)&lt;0)
 {
  fprintf(stderr,"\nError setting the filter.\n");
  pcap_close(fp);
  /* Free the device list */
  return;
 }

 /* Put the interface in statstics mode */
 if (pcap_setmode(fp, MODE_STAT)&lt;0)
 {
  fprintf(stderr,"\nError setting the mode.\n");
  pcap_close(fp);
  /* Free the device list */
  return;
 }


 printf("TCP traffic summary:\n");

 /* Start the main loop */
 pcap_loop(fp, 0, dispatcher_handler, (PUCHAR)&amp;st_ts);

 pcap_close(fp);
 return;
}

void dispatcher_handler(u_char *state,
 const struct pcap_pkthdr *header,
 const u_char *pkt_data)
{
 struct timeval *old_ts = (struct timeval *)state;
```

```
u_int delay;
 LARGE_INTEGER Bps,Pps;
struct tm ltime;
char timestr[16];
time_t local_tv_sec;

/* Calculate the delay in microseconds from the last sample. This value
 * is obtained from the timestamp that the associated with the sample. */
delay = (header-&gt;ts.tv_sec - old_ts-&gt;tv_sec) * 1000000
  - old_ts-&gt;tv_usec + header-&gt;ts.tv_usec;
/* Get the number of Bits per second */
Bps.QuadPart=(((*(LONGLONG*)(pkt_data + 8)) * 8 * 1000000) / (delay));
/*                                      ^     ^
                                        |     |
                                        |     |
                                        |     |
                    converts bytes in bits --       |
                                                    |
                delay is expressed in microseconds --
*/

/* Get the number of Packets per second */
Pps.QuadPart=(((*(LONGLONG*)(pkt_data)) * 1000000) / (delay));

/* Convert
the timestamp to readable format */
local_tv_sec = header-&gt;ts.tv_sec;
localtime_s(&amp;ltime, &amp;local_tv_sec);
strftime( timestr, sizeof timestr, "%H:%M:%S", &amp;ltime);

/* Print timestamp*/
printf("%s ", timestr);

/* Print the samples */
printf("BPS=%I64u ", Bps.QuadPart);
printf("PPS=%I64u\n", Pps.QuadPart);

//store current timestamp
old_ts-&gt;tv_sec=header-&gt;ts.tv_sec;
old_ts-&gt;tv_usec=header-&gt;ts.tv_usec;
}


void usage()
{
 printf("\nShows the TCP traffic load, in bits per second and packets per second."
       "\nCopyright (C) 2002 Loris Degioanni.\n");
 printf("\nUsage:\n");
```

```
 printf("\t tcptop adapter\n");
 printf("\t You can use \"WinDump -D\" if you don't know the name of your adapters.\n");

 exit(0);
}
</pre>

  <p>Before enabling statistical mode, the user has the option to set a
    filter that defines the subset of network traffic that will be monitored.
    See the <a class="ulink" href="wpcap/pcap-filter.html" target="_top">Filtering
 expression
      syntax</a> documentation for details. If no filter has been set,
    all of the traffic will be monitored.</p>

  <p>Once
    </p><div class="itemizedlist"><ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">the
filter is set</li><li class="listitem"><code class="literal">pcap_setmode()</code> is called</li><li
class="listitem">callback invocation is enabled with <a class="ulink" href="wpcap/pcap_loop.html"
target="_top">pcap_loop()</a></li></ul></div><p>
    the interface descriptor starts to work in statistical mode. Notice the
    fourth parameter (<code class="literal">to_ms</code>) of <a class="ulink" href="wpcap/pcap_open.html"
target="_top">pcap_open()</a>: it defines the interval
    among the statistical samples. The callback function receives the samples
    calculated by the driver every <code class="literal">to_ms</code> milliseconds.
    These samples are encapsulated in the second and third
 parameters of the
    callback function.

    Two 64-bit counters are provided: the number of packets and the amount of
    bytes received during the last interval.</p>

  <p>In the example, the adapter is opened with a timeout of 1000 ms. This
    means that dispatcher_handler() is called once per second. At this point
    a filter that keeps only tcp packets is compiled and set. Then
    <code class="literal">pcap_setmode()</code> and <code class="literal">pcap_loop()</code> are
    called. Note that a struct timeval pointer is passed to
    <code class="literal">pcap_loop()</code> as the <code class="literal">user</code> parameter.
    This structure will be used to store a timestamp in order to calculate
    the interval between two samples.  dispatcher_handler()uses this interval
    to obtain the bits per second and the packets per second and then prints
    these values on the screen.</p>

  <p>Note finally that this example is by far more efficient
 than a
    program that captures the packets in the traditional way and calculates
    statistics at user-level. Statistical mode requires the minumum amount of
    data copies and context switches and therefore the CPU is optimized.
    Moreover, a very small amount of memory is required.</p>
```

<hr><table width="100%" summary="Navigation footer"><tr><td width="40%" align="left">Prev</td><td width="20%" align="center"></td><td width="40%" align="right">Next</td></tr><tr><td width="40%" align="left" valign="top">The Npcap API</td><td width="20%" align="center">Home</td><td width="40%" align="right" valign="top">Npcap internals</td></tr></table> </body></html>

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/docs/npcap-tutorial.html
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2013, Petar Alilovic,
* Faculty of Electrical Engineering and Computing, University of Zagreb
* All rights reserved
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* * Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* * Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS
 BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
* DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/nflog.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008 CACE Technologies, Davis (California)
* All rights reserved.
*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of CACE Technologies nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This sample was contributed by
* Marcin Okraszewski (Marcin.OkraszewskiATpl.compuware.com)
*
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/pcap_fopen/pcap_fopen.cpp
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2005 - 2006
* CACE Technologies, Davis, CA
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of CACE Technologies nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
 INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Examples-remote/UserLevelBridge/UserBridge.c
No license file was found, but licenses were detected in source scan.

<html><head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><title>Npcap Reference Guide</title><meta name="generator" content="DocBook XSL Stylesheets V1.79.2"><meta name="description" content="A manual and guide to Npcap, a packet capture and network analysis framework for Windows, for users and software developers. Npcap is a modern, safe, and compatible update to WinPcap."><link rel="home" href="index.html" title="Npcap Reference Guide"><link rel="next" href="npcap-users-guide.html" title="Npcap Users' Guide"></head><body bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF"> <table width="100%" summary="Navigation header"><tr><th colspan="3" align="center">Npcap Reference Guide</th></tr><tr><td width="20%" align="left"></td><th width="60%" align="center"></th><td width="20%" align="right">Next</td></tr></table><hr> <div
 class="article"> <h2 class="title">Npcap Reference Guide</h2> Abstract
   A manual and guide to Npcap, a packet capture and network analysis framework for Windows, for users and
    software developers. Npcap is a modern, safe, and compatible update to WinPcap.
 <hr> Table of Contents <dl class="toc"><dt>Introduction</dt><dd><dl><dt>What is Npcap?</dt><dt>Npcap Features</dt><dt>Purpose of this manual</dt><dt>Terminology</dt><dt>Npcap

License</dt><dt>Obtaining Npcap</dt><dt>Acknowledgements and copyright</dt></dl></dd><dt>Npcap Users' Guide</dt><dd><dl><dt>Installation</dt><dt>Windows platforms supported</dt><dt>How to use Wireshark to capture raw 802.11 traffic in “Monitor Mode”</dt><dt>Q & A</dt><dt>Reporting Bugs</dt></dl></dd><dt>Developing software with Npcap</dt><dd><dl><dt>Using the Npcap SDK</dt><dt>Examples</dt><dt>Updating WinPcap software to Npcap</dt><dt>How to detect what version Npcap/WinPcap you are using?</dt><dt>For software that want to use Npcap first when Npcap and WinPcap coexist</dt><dt>For software that uses Npcap loopback feature</dt><dt>For software that uses Npcap raw 802.11 feature</dt></dl></dd><dt>The Npcap API</dt><dd><dl><dt>Extensions to libpcap for Windows</dt></dl></dd><dt>Npcap Development Tutorial</dt><dd><dl><dt>Obtaining the device list</dt><dt>Obtaining advanced information about installed devices</dt><dt>Opening an adapter and capturing the packets</dt><dt>Capturing the packets without the callback</dt><dt>Filtering the traffic</dt><dt>Interpreting the packets</dt><dt>Handling offline dump files</dt><dt>Sending Packets</dt><dt>Gathering Statistics on the network traffic</dt></dl></dd><dt>Npcap internals</dt><dd><dl><dt>Npcap structure</dt><dt>Npcap driver internals</dt><dt>Further reading</dt></dl></dd></dl>

<h2 class="title" style="clear: both">Introduction</h2>

This Manual describes the programming interface and the source code of
Npcap. It provides detailed descriptions of the functions and structures
exported to programmers, along with complete documentation of the Npcap
internals. Several tutorials and examples are provided as well.

<h3 class="title">What is Npcap?</h3>

Npcap is an architecture for packet capture and network analysis for
Windows operating systems, consisting of a software library and a network
driver.

Most networking applications access the network through widely-used
operating system primitives such as sockets. It is easy to access data on
the network with this approach since the operating system copes with the
low level
details (protocol handling, packet reassembly, etc.) and
provides a familiar interface that is similar to the one used to read and
write files.

Sometimes, however, the “easy way” is not up to the task,
since some applications require direct access to packets on the network.
That is, they need access to the “raw” data on the network
without the interposition of protocol processing by the operating
system.

The purpose of Npcap is to give this kind of access to Windows
applications. It provides facilities to:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">capture raw packets, both the ones destined to the machine where
it's running and the ones exchanged by other hosts (on shared media)
<li class="listitem">filter the packets according to user-specified rules before
dispatching them to the application
<li class="listitem">transmit raw packets to the network
<li class="listitem">gather statistical information on the network traffic

This set of capabilities is obtained by means of a device driver,
which is installed inside the networking portion of the Windows kernel,
plus a couple of DLLs.

All of these features are exported through a powerful programming
interface, easily usable by applications. The main goal of this manual is
to document this interface, with the help of several examples.

<h4 class="title">What kind of programs use Npcap?</h4>

The Npcap programming interface can be used by many types of
network tools for analysis, troubleshooting, security and monitoring.
In particular,
classical tools that rely on Npcap are:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">network and protocol analyzers
<li class="listitem">network monitors
<li class="listitem">traffic loggers
<li class="listitem">traffic generators
<li class="listitem">user-level bridges and routers
<li class="listitem">network intrusion detection systems (NIDS)
<li class="listitem">network scanners
<li class="listitem">security tools

<h4 class="title">What Npcap can't do</h4>

Npcap receives and sends the packets independently from the host
protocols, like TCP/IP. This means that it isn't able to block, filter or
manipulate the traffic generated by other programs on the same machine: it
simply “sniffs”
the packets that transit on the wire. Therefore, it does not
provide the appropriate support for applications like traffic shapers, QoS
schedulers and personal firewalls.

<h3 class="title">Npcap Features</h3>

Npcap has many exciting features that set it above other packet capture solutions:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem"> Built for modern Windows: Npcap is written for Windows 10, Windows 8.1, Windows 8, and Windows 7. Using up-to-date NDIS versions, it allows you to capture traffic without slowing down the network stack. Npcap is implemented as a NDIS 6 Lightweight Filter driver, faster and with less overhead
than the legacy <a class="ulink" href="https://docs.microsoft.com/en-us/previous-versions/windows/hardware/network/ff557149(v=vs.85)"
target="_top">NDIS 5 Protocol Driver
used by WinPcap.
<li class="listitem"> WinPcap compatibility: Npcap is a
drop-in replacement for WinPcap

in most applications.
<li class="listitem"> Updated cross-platform libpcap API:
The libpcap API allows cross-platform packet capture applications
to target Linux, Windows, macOS, BSD, Solaris and others. Npcap includes
the latest version of libpcap,
providing the best solution for compatibility, performance, functionality, and security.
<li class="listitem"> Loopback packet capture and injection: Npcap is able to
see
Windows loopback packets using the

Windows Filtering Platform (WFP). Npcap supplies an
interface named “NPF_Loopback”, with the description “Adapter for loopback capture.”
Wireshark users can choose this adapter to capture all loopback traffic the same way as other
non-loopback adapters.
Packet injection works as well with <code class="function">pcap_inject()</code>.
<li class="listitem"> Raw 802.11 Packet Capture Support: Npcap is able to see
802.11 frames instead of emulated Ethernet frames on ordinary wireless

adapters. You need to select the <code class="option">Support raw 802.11 traffic (and monitor
mode) for wireless adapters</code> option in the installation wizard to enable
this feature. When your adapter is in “Monitor Mode”, Npcap will supply all
802.11 data + control + management packets with Radiotap headers. When
your adapter is in “Managed Mode”, Npcap will only supply Ethernet
packets. Npcap directly supports using Wireshark to capture in “Monitor Mode”.
Npcap also provides the <code class="filename">WlanHelper.exe</code>
tool to manually configure WiFi PHY parameters.
See more details
about this feature in the section called “For software that uses Npcap raw 802.11 feature”.
<li class="listitem"> “Admin-only Mode” Support: Npcap supports restricting its
use to Administrators for safety purpose. If Npcap is installed with
the option “Restrict Npcap driver's access to Administrators only” checked,
only Built-in Administrators may access its features via user software (Nmap, Wireshark, etc).
This provides a level of restriction similar to requiring root access for packet capture on

Linux/UNIX.

<h3
class="title">Purpose of this manual</h3>

The purpose of this manual is to provide a comprehensive and easy way
to browse the documentation of the Npcap architecture. You will find
three main sections:

the section called “Npcap Users' Guide” is for end users of Npcap, and
primarily concerns installation options, hardware compatibility, and bug
reporting procedures.

the section called “Developing software with Npcap” is for programmers who need to use
Npcap from an application: it contains information about functions and
data structures exported by the Npcap API, a manual for writing packet
filters, and information on how to include it in an application. A
tutorial with
several code samples is provided as well; it can be used to
learn the basics of the Npcap API using a step-by-step approach, but it
also offers code snippets that demonstrate advanced features.

the section called “Npcap internals” is intended for Npcap developers
and maintainers, or for people who are curious about how this system
works: it provides a general description of the Npcap architecture and
explains how it works. Additionally, it documents the complete device
driver structure, the source code, the Packet.dll interface and the
low-level Npcap API. If you want to understand what happens inside Npcap
or if you need to extend it, this is the section you will want to
read.

<h3 class="title">Terminology</h3>

We call
Npcap an architecture rather than
library because packet capture is a low level
mechanism that requires a strict interaction with the network adapter and
with the operating system, in particular with its networking
implementation, so a simple library is not sufficient.

For consistency with the literature, we will use the term
packet even though
frame is more accurate since the capture process
is done at the data-link layer and the data-link header is included in
the captured data.

<h3 class="title">Npcap License</h3>

Even though Npcap source code is publicly available for review, it is
not open source software and may not be redistributed without special
permission from the Nmap
Project. The
Npcap
Free License
allows end users to download, install, and use up to 5 copies of
Npcap from our site for
free. Copies which are only used with Nmap, Wireshark, and/or
Microsoft
Defender for Identity don't count toward this 5-install
limit.

We fund the Npcap project by selling the Npcap OEM
Edition. This special version of Npcap includes enterprise
features such as the silent installer and commercial support as
well as special license rights allowing customers to redistribute
Npcap with their products or to install it on more systems within
their organization
with easy enterprise deployment. We offer two
commercial license types:

<ul class="itemizedlist" style="list-style-type: disc; "><li class="listitem">
The Npcap OEM
Redistribution License is for companies that wish to
distribute Npcap OEM within their products (the free Npcap
edition does not allow this). Licensees generally use the
Npcap OEM silent installer, ensuring a seamless experience for
end users. Licensees may choose between a perpetual unlimited
license or an annual term license, along with options for
commercial support and updates.
<li class="listitem">

The Npcap OEM
Internal-Use License is for organizations that wish to
use Npcap OEM internally without redistribution outside their
organization. This allows them to bypass the 5-system usage
cap of
the Npcap free edition. It includes commercial support
and update options, and provides the extra Npcap OEM features
such as the silent installer for enterprise-wide
deployment.

<h3 class="title">Obtaining Npcap</h3>

The latest Npcap release can always be found
on the Npcap
website as an executable installer and as a source code
archive.

<h3 class="title">Acknowledgements and copyright</h3>

Npcap is an update of WinPcap.
It is developed
by the Nmap Project
as a continuation
of the project started by Yang Luo
under Google Summer of Code 2013 and
2015.
It also received many helpful tests from Wireshark
and NetScanTools.

Portions of this guide were adapted from the WinPcap documentation.

<hr><table width="100%" summary="Navigation footer"><tr><td width="40%" align="left"></td><td width="20%" align="center"></td><td width="40%" align="right">Next</td></tr><tr><td width="40%" align="left" valign="top"></td><td width="20%" align="center"></td><td width="40%" align="right" valign="top">Npcap Users' Guide</td></tr></table> </body></html>

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/docs/index.html
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names
 of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
*   @(#)bpf.h   7.1 (Berkeley) 5/7/91
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/bpf.h
No license file was found, but licenses were detected in source scan.

/*-
* Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* This code is derived from the Stanford/CMU enet packet filter,
* (net/enet.c) distributed as part of 4.3BSD, and code contributed
* to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
* Berkeley Laboratory.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features
 or use of this software
*   must display the following acknowledgement:
*   This product includes software developed by the University of
*   California, Berkeley and its contributors.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/sll.h
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/can_socketcan.h
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/ipnet.h
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap-bpf.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the
Laboratory may be used
* to endorse or promote products derived from this software without
* specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
/*

* Remote packet capture mechanisms and extensions from WinPcap:
*
* Copyright
(c) 2002 - 2003
* NetGroup, Politecnico di Torino (Italy)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the Politecnico di Torino nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/pcap.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1993, 1994, 1995, 1996, 1997
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
* This product includes software developed by the Computer Systems
* Engineering Group at Lawrence Berkeley Laboratory.
* 4. Neither the name of the University nor of the
 Laboratory may be used
*   to endorse or promote products derived from this software without
*   specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/socket.h
*
 /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap.h
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/compiler-tests.h
* /opt/cola/permits/1911993923_1708442971.280755/0/npcap-sdk-1-13-zip/Include/pcap/funcattrs.h

1.21 log4cplus 2.0.7

1.21.1 Available under license :

Log4cplus license
==========

Each file of log4cplus source is licensed using either two clause BSD
license or Apache license 2.0. Log4cplus is derived work from log4j.

Threadpool code is licensed under separate license.

Two clause BSD license

Apache 2.0 license

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on

behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion
in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE,
NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has
been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Log4j license
==========

Apache License; Version 2.0, January 2004; http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall
mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software
source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind
by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the
copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation,

if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission
of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer
failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work
==========

To
apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced
with your own identifying information. (Don't include the brackets!)
The text should be enclosed in the appropriate comment syntax for the
file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the
copyright notice for easier identification within third-party
archives.

Copyright 1999-2009 [Contributors to log4cplus project]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied.
See the License for the specific language governing permissions and
limitations under the License.

Threadpool license
==========

Copyright (c) 2012-2015 Jakob Progsch

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

3. This notice may not be removed or altered
from any source
distribution.

Modified for log4cplus, copyright (c) 2014-2015 Vclav Zeman.