

# SiLK Acceptance Tests (SiLK-3.14.0)

CERT Network Situational Awareness

November 17, 2016

### 1 Introduction

SiLK, the System for Internet-Level Knowledge, is a collection of traffic analysis tools to facilitate security analysis of large networks. The SiLK tool suite supports the efficient collection, storage and analysis of network flow data, enabling network security analysts to rapidly query large historical traffic data sets.

The tools in SiLK suite can be grouped into two categories:

- The packing tools are responsible for collecting flow records, converting them to the SiLK format, categorizing them, and storing them in the data repository.
- The analysis tools read SiLK flow records from the data repository and can display, sort, or group the flow records by various attributes and compute the flow volume of each group.

This document describes the testing procedures used to verify that the tools in the SiLK suite are implemented correctly and work as advertised.

#### 1.1 Structure of this document

This document begins with a general description of SiLK and of some conventions used in the tests.

The remainder of the document consists of the tests themselves, broken down by functional area. Each test is referenced by the requirement it tests, and is broken down into the following sections:

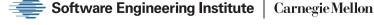
**Prerequisites** If the test requires some conditions to be satisfied that are outside the scope of the test, they will be mentioned here. This section may not be present if there are no special prerequisites for running the test.

**Preparation** Steps to conduct prior to the test. Some of these steps may be unnecessary to repeat between tests. If something goes wrong during this phase, the test is considered impossible to run due to error.

**Procedure** Steps to conduct during the test. These steps should be performed in order each time the test is run.

**Expected results** The tester should verify that these items occur at the appropriate points in the test procedure. If they do not, the test is considered a failure.

© 2009–2016 Carnegie Mellon University



# 2 Testing SiLK Analysis Tools

The tests for the SiLK Analysis Tools are included with the SiLK-3.14.0 source distribution. The tests are invoked by typing make check in an application build directory or at the top of the build tree to run the tests for all applications.

The data used to test is applications is created by a Perl script that generates text. This text is piped into the rwtuc application to create the SiLK flow records that are used for the tests.

The tests for an application invoke the application with various combinations of its options. Some tests are used to confirm that the application properly fails, for example, when incorrect or conflicting options are specified. Other tests confirm that the output of the application is correct. The output is assumed to be correct if the MD5 hash of the output matches an expected value. The expected value is determined or verified either by using unrelated SiLK applications or by directly processing the output produced by the Perl script that creates the text that was piped to rwtuc.

The tests for an application do not attempt an exhaustive permutation of all options, as that would require an extraordinary amount of time for the tests to complete. Knowledge of the software's source code is used to select options that exercise the majority of the application's functionality. When possible, unrelated options are used simultaneously to exercise multiple parts of the source code.

### 2.1 Prerequisites

The tests of the SiLK Analysis Tools require that Perl be installed on the system, and that the Digest::MD5 Perl module is installed.

### 2.2 Preparation

The tests assume you have configured and built SiLK. The tests in this section use the application binaries as they exist in the build tree. The tests do not require that you install SiLK prior to running the tests.

#### 2.3 Procedure

- 1. Go to the top of the directory where you built SiLK.
- 2. Type make check.

The full lists of tests that make check runs are listed in Section 9.

### 2.4 Expected results

The tests will take several minutes to run.

During the tests, you may see the following sorts of output.

• The following indicates a test that successfully passed.

```
PASS: tests/rwstats-version.pl
```

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/rwstats-sip24-top-pkt-p2.pl.log.

```
FAIL: tests/rwstats-sip24-top-pkt-p2.pl
```

• The following indicates that the test was skipped. For this skipped test, information about why the test was skipped may be available in the file tests/rwcut-icmp-type.pl.log.

```
SKIP: tests/rwcut-icmp-type.pl
```

A test can be skipped for two reasons.

- 1. The test is not applicable. For example, there is no need to test IPv6 functionality if SiLK was not compiled with IPv6 support.
- 2. A file or application that the test requires is not present. This can occur if you fail to build the SiLK tools prior to testing, so that make check is building the tools and testing them. Some tests use other tools in SiLK suite, and the tests will be skipped if the required tools are not available.
- The following indicates that no tests exist for the applications or libraries in the named directory.

```
Making check in plugins make[2]: Nothing to be done for 'all'.
```

When you run make check from the top-level directory and all tests are successful, you should see output similar to the following once all processing stops, and the exit status of make should be 0.

```
make[2]: Nothing to be done for 'check'.
make[2]: Nothing to be done for 'check-am'.
```

If one or more tests in a directory (e.g., src/rwcut) fails, make will stop processing once it finishes running the tests in that particular directory, and make will exit with a non-zero status. The end of the output will resemble

\_\_\_\_\_\_

In each directory that make visits, a summary of the results of running the tests in that directory is displayed. The summary resembles the following:

\_\_\_\_\_\_

\_\_\_\_\_

where

# ERROR: 0

**TOTAL** is the number of tests that were run

**PASS** is the number of tests that passed

**SKIP** is the number of tests that were skipped

 $\mathbf{XFAIL}$  will always be 0

FAIL is the number of tests that failed

**XPASS** will always be 0

 $\mathbf{ERROR}$  is the number of tests that had a fatal error

When the sum of PASS and SKIP equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

In each directory, details about why a test was skipped or why a test failed can be found in the test-suite.log file in that directory as well as in the \*.log files in the tests subdirectory of that directory.

# 3 Testing rwscanquery

rwscan is an application that reads SiLK flow records representing incoming traffic, attempts to find external hosts that are scanning the monitored network, and produces textual output describing what it found. Although the design concept of rwscan has it running periodically on the SiLK data files and inserting its results into a database, rwscan operates like most SiLK analysis tools: it is a self-contained program that reads SiLK flow records from the files listed on the command line or from the standard input and it produces textual output.

rwscanquery is a script that queries the database populated by the results from invocations of rwscan. Depending on the report that the user requests, rwscanquery will create textual output, binary IPset files, or files of SiLK flow records.

One of report options available from rwscanquery allows an analyst to provide a time window and an IPset of internal hosts to determine what external hosts scanned those internal hosts during the time window. However, the results from the rwscan program include the (external) hosts that are scanning the network, but they do not include the (internal) hosts that were the target of a scan. Thus, to produce its report, the rwscanquery program first asks the database for the external hosts that were scanning during the time window, then it uses rwfilter to find flow records from the scanning IPs that targeted the IPs in the IPset file provided by the analyst.

There are additional report options in rwscanquery that operate similarly. For all of these report types to produce output, rwscanquery must invoke other SiLK analysis tools (e.g., rwsetbuild, rwfilter), and rwscanquery requires access to a SiLK data repository.

The tests for rwscan and rwscanquery are included with the SiLK-3.14.0 source distribution. The tests are invoked by typing make check in the src/rwscan directory. (If you type make check at the top of the build tree, the rwscan tests will be invoked as make recursively descends into each directory.)

The tests are written in Perl. The tests use fictional data and they will confirm that rwscan finds potential scanners and that rwscanquery can query a SiLK data repository to produce the various reports that it supports.

Since rwscan is a self-contained program and the results from rwscan are textual, the results from one rwscan invocation are easy to compare with previous invocations. Ensuring that rwscanquery is operating correctly is a greater challenge, since it invokes other tools. This document describes the tests that check the behavior of rwscanquery.

## 3.1 Prerequisites

The tests of rwscan and rwscanquery require that the following tools are installed on the system:

- the sqlite3 program
- Perl 5.6 or later
- the Perl module Digest::MD5
- the Perl module DBD::SQLite

### 3.2 Preparation

The test scripts assume you have configured and built rwscan, rwscanquery, and all the libraries they require. The scripts use the application binary as it exists in the build tree, and the scripts do not require that you install SiLK prior to running the tests.

During the test, a temporary directory is created, and files and subdirectories are created in this directory. The directory is created in the location specified by the TMPDIR environment variable, or in /tmp when the TMPDIR environment variable is not set.

#### 3.3 Procedure

1. Go to src/rwscan subdirectory in the directory tree where you built SiLK.

2. Type make check. This will invoke the tests that check the behaviors of all the applications in the src/rwscan directory, including rwscan and rwscanquery.

The following behaviors are tested.

- 1. Verify that rwscan query produces the expected textual output when running over fictional data.
- 2. Write the results from rwscan into a SQLite database.
- 3. Verify that rwscanquery can query the SQLite database and export textual output that matches the results from rwscan.
- 4. Verify that rwscanquery can produce textual output summarizing the scan volume seen per day.
- 5. Verify that rwscanquery can query the SQLite database for a particular scanning subnet and write the result as text.
- 6. Verify that rwscanquery can query the SQLite database for scanning IP addresses contained in an IPset. The results from rwscanquery are written as text.
- 7. Verify that rwscanquery can query the SQLite database over a particular time window and write the result as text.
- 8. Verify that rwscanquery can query the SQLite database and write an IPset file containing the scanning IPs for all records in the database.
- 9. Verify that rwscanquery can query the SQLite database and write an IPset file containing the scanning IPs that targeted a subnet of internal IP addresses.
- 10. Verify that rwscanquery can query the SQLite database and write an IPset file containing the scanning IPs that targeted internal IP addresses listed in an IPset file.
- 11. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing incoming records that originated from all scanning IPs.
- 12. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing incoming records that originated from (external) scanning IPs specified by one subnet that targeted (internal) IPs specified by another subnet.
- 13. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing incoming records that originated from scanning IPs listed in an IPset file that targeted IPs specified by a subnet.
- 14. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing incoming records that originated from scanning IPs specified by a subnet that targeted IPs listed in an IPset file.
- 15. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing outgoing records that originated from internal IPs and that may have been responses to activity by the scanning IPs for all internal IPs and scanning IPs.
- 16. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing outgoing records that originated from internal IPs specified by a subnet and that may have been responses to activity by scanning IPs listed in an IPset file.
- 17. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing outgoing records that originated from internal IPs specified by a subnet and that may have been responses to activity by scanning IPs specified by a another subnet.

18. Verify that rwscanquery can query the SQLite database and write a SiLK flow file containing outgoing records that may have been responses from internal IPs listed in an IPset file and that may have been responses to activity by scanning IPs specified by a subnet.

### 3.4 Expected results

The tests may take several minutes to run.

During the tests, you may see the following sorts of output.

• The following indicates a test that successfully passed.

```
PASS: tests/rwscanquery-sqlite.pl
```

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/rwscanquery-sqlite.pl.log.

```
FAIL: tests/rwscanquery-sqlite.pl
```

• The following indicates that the test was skipped. For this skipped test, information about why the test was skipped may be available in the file tests/rwscanquery-sqlite.pl.log.

```
SKIP: tests/rwscanquery-sqlite.pl
```

A test will be skipped if a file or application that the test requires is not present. This can occur if the prerequisites described above are not available, or it can occur if you fail to build the SiLK tools prior to testing, so that make check is building the tools and testing them. Some tests use other tools in SiLK suite, and the tests will be skipped if the required tools are not available.

Once all processing stops, you should see output similar to the following to summarize the results of running the tests.

Testsuite summary for SiLK 3.6.0

# TOTAL: 13

# PASS: 12

# XFAIL: 0

# FAIL: 0

# XPASS: 0

# ERROR: 0

where

**TOTAL** is the number of tests that were run

PASS is the number of tests that passed

**SKIP** is the number of tests that were skipped

**XFAIL** will always be 0

FAIL is the number of tests that failed

**XPASS** will always be 0

**ERROR** is the number of tests that had a fatal error

When the sum of PASS and SKIP equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

Details about why a test was skipped or why a test failed can be found in the test-suite.log file in the src/rwscanquery directory as well as in the \*.log files in the src/rwscanquery/tests directory.

### 4 Testing rwsender and rwreceiver

rwsender is a daemon which transfers files over the network to one or more rwreceiver daemons. An rwreceiver may accept files from multiple rwsenders. Either rwsender or rwreceiver may act as the server and accept connections from rwreceiver or rwsender processes acting as clients. The connection between rwsender and rwreceiver may be encrypted using GnuTLS. rwsender and rwreceiver do not require the files they transfer to have any particular format; they treat the contents of the files as a stream of bytes.

The tests will determine whether rwsender can successfully send files to rwreceiver processes, and whether an rwreceiver can successfully receive files from rwsender processes. If SiLK was configured with GnuTLS support, tests will also be conducted using GnuTLS.

Tests are included with the SiLK-3.14.0 distribution that run tests on rwsender and rwreceiver. To run the tests, go into the src/sendrcv directory and type make check. (If you type make check at the top of the build tree, the tests will be invoked as make recursively descends into each directory.) The tests invoke the daemons, have them connect, send files, and shut down. Some of the tests will involve shutting down one side of the connection during file transfer to verify that the other side handles that situation correctly.

### 4.1 Prerequisites

The tests of rwsender and rwreceiver require that the following tools are installed on the system:

- Python 2.6 or later
- Perl 5.6 or later
- the Perl module Digest::MD5

### 4.2 Preparation

The test script assumes you have configured and built rwsender, rwreceiver, and all the libraries they require. The script uses the application binaries as they exist in the build tree, and the script does not require that you install SiLK prior to running the tests.

During many of the tests, a temporary directory is created, and files and subdirectories are created in this directory. The directory is created in the location specified by the TMPDIR environment variable, or in /tmp when the TMPDIR environment variable is not set.

#### 4.3 Procedure

- 1. Go to src/sendrcv subdirectory in the directory tree where you built SiLK.
- 2. Type make check. This will invoke some basic checks on rwsender and rwreceiver and then invoke the scripts that attempt to connect them.

The rwsender/rwreceiver tests check the following behaviors:

- 1. Simple connection to 127.0.0.1. With rwreceiver acting as a server and rwsender acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection when connecting as the IPv4 localhost address, and shut down cleanly.
- 2. Simple connection to localhost. This test is similar to the previous, except the connection is made using "localhost".
- 3. **Simple connection to ::1.** This test is similar to the previous, except the connection is made using the IPv6 localhost address. This test is skipped when IPv6 networking support is not available.
- 4. **GnuTLS connection.** With rwreceiver acting as a server and rwsender acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection using GnuTLS, and shut down cleanly. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 5. **Stop** rwreceiver server. With rwreceiver acting as a server and rwsender acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwreceiver a SIGTERM, causing it to shut down cleanly. Restart rwreceiver and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.
- 6. Stop rwreceiver server when using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 7. Stop rwsender server. With rwsender acting as a server and rwreceiver acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwsender a SIGTERM, causing it to shut down cleanly. Restart rwsender and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.
- 8. Stop rwsender server when using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.

9. Stop rwreceiver client. With rwsender acting as a server and rwreceiver acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwreceiver a SIGTERM, causing it to shut down cleanly. Restart rwreceiver and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.

- 10. **Stop** rwreceiver client using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 11. **Stop** rwsender client. With rwreceiver acting as a server and rwsender acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwsender a SIGTERM, causing it to shut down cleanly. Restart rwsender and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.
- 12. **Stop rwsender client when using GnuTLS.** This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 13. Kill rwreceiver server. With rwreceiver acting as a server and rwsender acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwreceiver a SIGKILL, causing it to abruptly shut down. Check whether rwsender handles the sudden loss of connectivity. Restart rwreceiver and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.
- 14. Kill rwreceiver server when using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 15. Kill rwsender server. With rwsender acting as a server and rwreceiver acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwsender a SIGKILL, causing it to abruptly shut down. Check whether rwreceiver handles the sudden loss of connectivity. Restart rwsender and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.
- 16. Kill rwsender server when using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 17. Kill rwreceiver client. With rwsender acting as a server and rwreceiver acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwreceiver a SIGKILL, causing it to abruptly shut down. Check whether rwsender handles the sudden loss of connectivity. Restart rwreceiver and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.
- 18. Kill rwreceiver client when using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 19. Kill rwsender client. With rwreceiver acting as a server and rwsender acting as a client, check whether rwsender and rwreceiver start correctly, establish a connection, and begin to transfer files. During file transfer, send rwsender a

SIGKILL, causing it to abruptly shut down. Check whether rwreceiver handles the sudden loss of connectivity. Restart rwsender and verify that the connection is reestablished and that file transfer resumes. Finally, check whether rwsender and rwreceiver shut down cleanly.

- 20. Kill rwsender client when using GnuTLS. This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 21. **Multiple connections.** Start two rwreceiver processes acting as clients and two rwsender processes acting as servers. Check whether each of the rwreceiver clients establish a connection with each of the rwsender servers. Verify that files from each rwsender are sent to each rwreceiver. Check whether all four daemons shut down cleanly.
- 22. **Multiple connections when using GnuTLS.** This test is similar to the previous, except the connections are made with GnuTLS. This test uses certificates that are included in the SiLK source code. This test is skipped when GnuTLS support is not available.
- 23. **Filtering.** Start two rwreceiver processes acting as clients and a single rwsender process acting as a server. Check whether each of the rwreceiver clients establish a connection with rwsender. Use filtering rules on rwsender so that a subset of the files are sent to each rwreceiver. Verify that the correct files are sent. Check whether all three daemons shut down cleanly.
- 24. **Post processing.** Start rwreceiver acting as a server and rwsender acting as a client. Establish a connection and successfully transfer files. For each file, verify that the command specified rwreceiver's --post-command switch is executed. Check whether the daemons shut down cleanly.

### 4.4 Expected results

The tests may take several minutes to run.

During the tests, you will see the following sorts of output.

• The following indicates a test that successfully passed.

PASS: tests/sendrcv-testConnectOnlyIPv4Addr.pl

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/sendrcv-testConnectOnlyIPv6Addr.pl.log.

FAIL: tests/sendrcv-testConnectOnlyIPv6Addr.pl

• The following indicates that the test was skipped. For this skipped test, information about why the test was skipped may be available in the file tests/sendrcv-testConnectOnlyTLS.pl.log.

SKIP: tests/sendrcv-testConnectOnlyTLS.pl

A test can be skipped for the following reason:

1. The test is not applicable. For example, there is no need to test GnuTLS functionality if SiLK was not compiled with GnuTLS support.

Once all processing stops, you should see output similar to the following to summarize the results of running the tests.

Testsuite summary for SiLK 3.6.0

# TOTAL: 30

# PASS: 20

# SKIP: 10

# XFAIL: 0

\_\_\_\_\_

where

# FAIL: 0 # XPASS: 0 # ERROR: 0

**TOTAL** is the number of tests that were run

PASS is the number of tests that passed

**SKIP** is the number of tests that were skipped

**XFAIL** will always be 0

**FAIL** is the number of tests that failed

**XPASS** will always be 0

**ERROR** is the number of tests that had a fatal error

When the sum of PASS and SKIP equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

Details about why a test was skipped or why a test failed can be found in the test-suite.log file in the src/sendrcv directory as well as in the \*.log files in the src/sendrcv/tests directory.

A note on tests that fail: The Python code that drives the test makes heavy use of Python threads, and there have been instances where a test fails due to errors in Python, not because of errors in the rwsender or rwreceiver daemons.

# 5 Testing rwflowappend

The rwflowappend daemon is used to support multiple copies of the data store, or to allow the data to be stored on a machine separate from the machine where rwflowpack is running. Typically an rwsender-rwreceiver pair is used to move the data files from rwflowpack to rwflowappend. For testing purposes, the method used to inject files into rwflowappend is immaterial.

The tests for rwflowappend are included with the SiLK-3.14.0 source distribution. The tests are invoked by typing make check-rwflowappend in the src/rwflowpack directory. (If you type make check at the top of the build tree, the rwflowappend tests will be invoked as make recursively descends into each directory.)

The tests are written in a combination of Perl and Python. The tests will confirm that the rwflowappend daemon can start, process files, and terminate cleanly. The tests also confirm that rwflowappend handles unusual input files correctly.

### 5.1 Prerequisites

The tests of rwflowappend require that the following tools are installed on the system:

- Python 2.6 or later
- Perl 5.6 or later
- the Perl module Digest::MD5

### 5.2 Preparation

The test scripts assume you have configured and built rwflowappend and all the libraries it requires. The scripts use the application binary as it exists in the build tree, and the scripts do not require that you install SiLK prior to running the tests.

During many of the tests, a temporary directory is created, and files and subdirectories are created in this directory. The directory is created in the location specified by the TMPDIR environment variable, or in /tmp when the TMPDIR environment variable is not set.

#### 5.3 Procedure

- 1. Go to src/rwflowpack subdirectory in the directory tree where you built SiLK.
- 2. Type make check-rwflowappend. This will invoke the tests that check the behavior of rwflowappend.

The rwflowappend tests check the following behaviors:

- 1. **Append IPv4.** Check whether rwflowappend properly handles two files that exist in its incoming directory when rwflowappend is invoked. rwflowappend will create a new hourly data file, and append the second file to that hourly file. Both input files will be moved to the archive directory. When rwflowappend receives a signal, it should shut down cleanly. This test uses input files that contain only IPv4 data.
- 2. **Append IPv6.** This test is similar to the previous, except it uses a data file that contains IPv6 data. This test is only invoked when SiLK has been compiled with IPv6 support.
- 3. Post processing. Check whether rwflowappend properly handles the --hour-file-command and --post-command switches to notice a new hourly file and to process an incoming file after rwflowappend has processed it. This test is similar to the "Append IPv4" test; in addition, the --hour-file-command will write the name of the hourly file to a text file, and the --post-command will copy the incoming files to a separate location. When rwflowappend receives a signal, it should shut down cleanly.

4. **Time window.** Check whether rwflowappend properly handles the --reject-hours-past and --reject-hours-future switches. Files containing records with start times before the --reject-hours-past or after the --reject-hours-future times are stored in the error directory. All other files should appear in the archive directory and corresponding data files should be created. When rwflowappend receives a signal, it should shut down cleanly.

- 5. Bad input. Check whether rwflowappend properly handles unusual files in its incoming directory. One file is a SiLK data file that contains no records; rwflowappend should move this file to the archive directory and not create an hourly data file. The second unusual file is a file that does not contain the SiLK file header. rwflowappend should move this file into its error directory.
- 6. Many input files. Check whether rwflowappend properly handles combining about 16,925 incremental files into 432 hourly files. The incremental files exist in rwflowappend's incoming directory when it is invoked. To create an each hourly file, rwflowappend will combine approximately 39 incremental files. The input files will be deleted. When rwflowappend receives a signal, it should shut down cleanly. To create the incremental files, the test runs rwflowpack in sending mode which creates 432 incremental files, and then the test runs rwsplit on each of those files. This test uses input files that contain only IPv4 data.

### 5.4 Expected results

The tests may take several minutes to run.

During the tests, you may see the following sorts of output.

• The following indicates a test that successfully passed.

PASS: tests/rwflowappend-version.pl

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/rwflowappend-append-ipv4.pl.log.

FAIL: tests/rwflowappend-append-ipv4.pl

• The following indicates that the test was skipped. For this skipped test, information about why the test was skipped may be available in the file tests/rwflowappend-append-ipv6.pl.log.

SKIP: tests/rwflowappend-append-ipv6.pl

A test can be skipped for two reasons.

- 1. The test is not applicable. For example, there is no need to test IPv6 functionality if SiLK was not compiled with IPv6 support.
- 2. A file or application that the test requires is not present. This can occur if you fail to build the SiLK tools prior to testing, so that make check is building the tools and testing them. Some tests use other tools in SiLK suite, and the tests will be skipped if the required tools are not available.

Once all processing stops, you should see output similar to the following to summarize the results of running the tests.

\_\_\_\_\_\_

Testsuite summary for SiLK 3.6.0

# TOTAL: 8
# PASS: 7
# SKIP: 1
# XFAIL: 0
# FAIL: 0
# ERROR: 0

where

**TOTAL** is the number of tests that were run

PASS is the number of tests that passed

**SKIP** is the number of tests that were skipped

**XFAIL** will always be 0

FAIL is the number of tests that failed

**XPASS** will always be 0

**ERROR** is the number of tests that had a fatal error

When the sum of PASS and SKIP equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

Details about why a test was skipped or why a test failed can be found in the test-suite.log file in the src/rwflowpack directory as well as in the \*.log files in the src/rwflowpack/tests directory.

# 6 Testing flowcap

The flowcap daemon listens on user-specified network ports to collect NetFlow v5 and/or IPFIX flow records that are created by flow generators. Examples of flow generators include routers and software that processes packet capture (libpcap) data. flowcap converts the flow records into a SiLK format and stores the records in temporary files. These files are later processed by rwflowpack. The typical way to transfer files from flowcap to rwflowpack is via an rwsender-rwreceiver pair, though the administrator is free to use other software (such as scp or rsync).

At a minimum, the tests will determine whether flowcap can receive NetFlow v5 packets when listening on an IPv4 port. If flowcap was built with libfixbuf support, tests will be run to test receiving IPFIX packets. If IPv6 networking support is enabled, tests will be conducted with flowcap listening on an IPv6 port. If SiLK is built with support for storing IPv6 flow records, a test is run that sends IPFIX packets containing IPv6 addresses to flowcap listening on an IPv6 port.

The tests for flowcap are included with the SiLK-3.14.0 source distribution. The tests are invoked by typing make check in the src/flowcap directory. (If you type make check at the top of the build tree, the flowcap tests will be invoked as make recursively descends into each directory.)

The tests are written in a combination of Perl and Python. The tests will confirm that the flowcap daemon can start, read data from the network, write the data into files, and terminate cleanly. Verifying that the files produced by flowcap are consistent is sufficient; it is not necessary in these tests to confirm that rwflowpack can process the files.

### 6.1 Prerequisites

The tests of flowcap require that the following tools are installed on the system:

- Python 2.6 or later
- Perl 5.6 or later
- the Perl module Digest::MD5
- the Perl module Socket6

### 6.2 Preparation

The test scripts assume you have configured and built flowcap and all the libraries it requires. The scripts use the application binary as it exists in the build tree, and the scripts do not require that you install SiLK prior to running the tests.

During many of the tests, a temporary directory is created, and files and subdirectories are created in this directory. The directory is created in the location specified by the TMPDIR environment variable, or in /tmp when the TMPDIR environment variable is not set.

#### 6.3 Procedure

- 1. Go to src/flowcap subdirectory in the directory tree where you built SiLK.
- 2. Type make check. This will invoke the scripts that check the behavior of flowcap.

The flowcap tests check the following behaviors:

- 1. Collect NetFlow v5 records when listening as 127.0.0.1. Check whether flowcap properly starts, accepts NetFlow v5 UDP packets on a UDP port bound to the IPv4 localhost address, converts the NetFlow v5 packets to SiLK flow records, and shuts down cleanly.
- 2. Collect NetFlow v5 records when listening as any host. Check whether flowcap properly starts, accepts NetFlow v5 UDP packets from the IPv4 localhost address when listening on a UDP port bound to the any address, converts the NetFlow v5 packets to SiLK flow records, and shuts down cleanly.

3. Collect NetFlow v5 records when listening as ::1. Check whether flowcap properly starts, accepts NetFlow v5 UDP packets from the IPv6 localhost when listening on a UDP port bound to the IPv6 localhost address, converts the NetFlow v5 packets to SiLK flow records, and shuts down cleanly

- 4. Collect IPFIX records when listening as 127.0.0.1. Check whether flowcap properly starts, accepts IPFIX packets on a TCP port bound to the IPv4 localhost address, converts the IPFIX packets to SiLK flow records, and shuts down cleanly. The IPFIX packets contain only IPv4 addresses.
- 5. Collect IPFIX records when listening as any host. Check whether flowcap properly starts, accepts IPFIX packets from the IPv4 localhost address when listening on a TCP port bound to the any addresses, converts the IPFIX packets to SiLK flow records, and shuts down cleanly. The IPFIX packets contain only IPv4 addresses.
- 6. Collect IPFIX records when listening as ::1. Check whether flowcap properly starts, accepts IPFIX packets from the IPv6 localhost address when listening on a TCP port bound to the IPv6 localhost address, converts the IPFIX packets to SiLK flow records, and shuts down cleanly. The IPFIX packets contain only IPv4 addresses.
- 7. Collect IPv6 IPFIX records when listening as ::1. Check whether flowcap properly starts, accepts IPFIX packets from the IPv6 localhost address when listening on a TCP port bound to the IPv6 localhost address, converts the IPFIX packets to SiLK flow records, and shuts down cleanly. The IPFIX packets contain only IPv6 addresses.

### 6.4 Expected results

The tests may take several minutes to run.

During the tests, you will see the following sorts of output.

• The following indicates a test that successfully passed.

PASS: tests/flowcap-version.pl

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/flowcap-append-ipv4.pl.log.

FAIL: tests/flowcap-append-ipv4.pl

• The following indicates that the test was skipped. For this skipped test, information about why the test was skipped may be available in the file tests/flowcap-ipfix.pl.log.

SKIP: tests/flowcap-ipfix.pl

A test can be skipped for two reasons.

- 1. The test is not applicable. For example, there is no need to test IPFIX functionality if SiLK was not compiled with IPFIX support.
- 2. A file or application that the test requires is not present. This can occur if you fail to build the SiLK tools prior to testing, so that make check is building the tools and testing them. Some tests use other tools in SiLK suite, and the tests will be skipped if the required tools are not available.

Once all processing stops, you should see output similar to the following to summarize the results of running the tests.

\_\_\_\_\_\_

Testsuite summary for SiLK 3.6.0

\_\_\_\_\_\_

# TOTAL: 10
# PASS: 9
# SKIP: 1
# XFAIL: 0
# FAIL: 0
# XPASS: 0

# ERROR: 0

where

 $\mathbf{TOTAL}$  is the number of tests that were run

**PASS** is the number of tests that passed

**SKIP** is the number of tests that were skipped

**XFAIL** will always be 0

**FAIL** is the number of tests that failed

**XPASS** will always be 0

**ERROR** is the number of tests that had a fatal error

When the sum of PASS and SKIP equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

Details about why a test was skipped or why a test failed can be found in the test-suite.log file in the src/flowcap directory as well as in the \*.log files in the src/flowcap/tests directory.

# 7 Testing rwflowpack

rwflowpack is the heart of the SiLK packing system. It may either collect NetFlow v5 and/or IPFIX flow records itself (similar to flowcap), or it may process the following types of files:

- files created by flowcap
- files containing NetFlow v5 PDUs, such as those created by NetFlow Collector
- files generated by the yaf program which contain IPFIX flow records

• files containing SiLK flow records generated by other SiLK applications

rwflowpack is responsible for deciding how and where each flow record gets written into the data store. rwflowpack splits the flow data by hour and chooses a *flowtype* (also called a *class/type* pair) for the record according to "packing logic". The packing logic normally categorizes data as incoming or outgoing, and it chooses an appropriate file format for the data.

There are four input-modes for rwflowpack.

- In "stream" input mode, rwflowpack opens an input "stream" for every *probe* listed in the sensor.conf file. These streams can be network ports where rwflowpack will read NetFlow v5 or IPFIX records, or they can directories that are routinely polled for files containing NetFlow v5 PDUs, IPFIX records, or SiLK files.
- In "fcfiles" input mode, rwflowpack polls a directory for files created by flowcap. In this mode, the probe definitions in the sensor.conf file are ignored, and instead rwflowpack uses the probe name written into each file's header.
- In "pdufile" input mode, rwflowpack reads NetFlow v5 PDUs from a single file specified on the command line, then rwflowpack exits.
- In "respool" input mode, rwflowpack does not recategorize the data; instead, rwflowpack reads SiLK flow files and puts each record into a flow file using the sensor and class/type values that already exist on the record.

There are two output-modes for rwflowpack. In the first, rwflowpack writes the data directly to the data store; this is called "local-storage" mode. In the second (called "sending" mode), rwflowpack stores the flow records in temporary files, and an rwflowappend process is responsible for writing the flow records into the data store. Typically rwflowpack and rwflowappend are running on separate machines, and an rwsender-rwreceiver pair is used to transfer the temporary files between the machines.

The tests for rwflowpack are included with the SiLK-3.14.0 source distribution. The tests are invoked by typing make check-rwflowpack in the src/rwflowpack directory. (If you type make check at the top of the build tree, the rwflowpack tests will be invoked as make recursively descends into each directory.)

The tests are written in a combination of Perl and Python. The tests will confirm that the rwflowpack daemon can start, process files, and terminate cleanly. The tests also confirm that rwflowpack handles unusual input files correctly.

### 7.1 Prerequisites

The tests of rwflowpack require that the following tools are installed on the system:

- Python 2.6 or later
- Perl 5.6 or later
- the Perl module Digest::MD5
- the Perl module Socket6

### 7.2 Preparation

The test scripts assume you have configured and built rwflowpack and all the libraries it requires. The scripts use the application binary as it exists in the build tree, and the scripts do not require that you install SiLK prior to running the tests.

During many of the tests, a temporary directory is created, and files and subdirectories are created in this directory. The directory is created in the location specified by the TMPDIR environment variable, or in /tmp when the TMPDIR environment variable is not set.

#### 7.3 Procedure

- 1. Go to src/rwflowpack subdirectory in the directory tree where you built SiLK.
- 2. Type make check-rwflowpack. This will invoke the tests that check the behavior of rwflowpack.

The rwflowpack tests check the following behaviors (unless otherwise stated, rwflowpack is running in "stream" input mode and "local-storage" output mode):

- 1. **Sensor configuration.** Verify that rwflowpack correctly parses a sensor configuration file that contains both valid and invalid probe and sensor definitions.
- 2. Pack SiLK IPv4 file. Check whether rwflowpack starts, uses its directory poller to find a file (that was present when rwflowpack was started), reads the SiLK flow records from the file, creates files and directories in its data directory, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data.
- 3. Pack SiLK IPv6 file. This test is similar to the previous, except it uses a data file that contains IPv6 data. This tests is only invoked when SiLK has been compiled with IPv6 support.
- 4. **Directory polling check.** This test is similar to the previous test, except the file is put into the polling directory after rwflowpack has started. This ensures that the directory poller works as expected.
- 5. Pack IPFIX IPv4 file. Check whether rwflowpack starts, uses its directory poller to find a file, reads the IPFIX records from the file, creates files and directories in its data directory, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data. This test is only invoked when SiLK has been compiled with IPFIX support.
- 6. Pack IPFIX IPv6 file. This test is similar to the previous, except it uses a data file that contains IPv6 data. This test is only invoked when SiLK has been compiled with both IPFIX and IPv6 support.
- 7. Pack IPFIX from network (127.0.0.1). Check whether rwflowpack starts, reads IPFIX records on a TCP socket bound to an IPv4 address, creates files and directories in its data directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data. This test is only invoked when SiLK has been compiled with IPFIX support.
- 8. Pack IPFIX from network (::1). This test is similar to the previous, except rwflowpack binds to an IPv6 address. This test is only invoked when SiLK has been compiled with both IPFIX and IPv6 networking support. This test requires the Perl Socket6 module.

9. Pack NetFlow v5 file. Check whether rwflowpack starts, uses its directory poller to find a file, reads the NetFlow v5 PDU records from the file, creates files and directories in its data directory, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test also verifies that the --packing-logic switch works as expected.

- 10. Run in "sending" output-mode. Check whether rwflowpack starts, uses its directory poller to find a file, reads the SiLK flow records from the file, creates files in its sending directory, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data. This test also verifies that the --pack-interfaces switch causes the "in" and "out" fields to appear in the output files.
- 11. Run in "sending" output-mode and apply a command. Check whether rwflowpack starts, uses its directory poller to find incoming files, reads the SiLK flow records from the files, creates files in its sending directory, moves the incoming file to its archive directory, invokes a command on the incoming files after moving them to the archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data.
- 12. Run in "fcfiles" input-mode. Check whether rwflowpack starts, finds a file in its incoming directory, reads the probe name and flowcap records from the file, creates files and directories in its data directory, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data.
- 13. Run in "respool" input-mode. Check whether rwflowpack starts, uses its directory poller to find incoming files, reads the SiLK flow records from the files, creates files and directories in its data directory based on the sensor and class/type data that exists on the flow records, moves the input files to the archive directory, and exits cleanly. This test uses an input file that contains only IPv4 data.
- 14. Run in "pdufile" input-mode. Check whether rwflowpack starts, reads the NetFlow v5 PDUs from a file specified on the command line, creates files and directories in its data directory, moves the PDU file to its archive directory, and exits cleanly.
- 15. Packing multiple streams. Check whether rwflowpack properly starts, polls two directories (containing SiLK flow files) and listens on two network ports (collecting NetFlow v5 PDUs), and exits cleanly. The test creates data files for three sensors, where the first sensor contains the data from one poll directory and one network port, the second sensor contains the remaining poll directory, and the third sensor contains the renaming network port.
- 16. Packing multiple streams. Check whether rwflowpack properly starts, polls two directories containing SiLK flow files and two other directories containing files of NetFlow v5 PDUs, and exits cleanly. The test creates data files for three sensors, where the first sensor contains the data from one SiLK directory and one NetFlow v5 directory, the second sensor contains the remaining SiLK directory, and the third sensor contains the remaining NetFlow v5 directory.
- 17. **Discarding flows matching CIDR block.** Check whether rwflowpack properly starts, uses its directory poller to find a file, reads the SiLK flow records from the file, discards records that have a source or destination IP in a particular CIDR block, creates files and directories in its data directory containing the remaining flows, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data.
- 18. **Discarding flows not matching CIDR block.** This test is similar to the previous, except flow records that do not match the specified CIDR block are discarded.
- 19. Categorizing and discarding flows matching an IPv4 IPset. Check whether rwflowpack properly starts, uses its directory poller to find a file, reads the SiLK flow records from the file, discards records that have a source or destination IP in an IPset containing IPv4 addresses, properly categorizes each flow by comparing its source and destination IP to an IPv4 IPset, creates files and directories in its data directory, moves the incoming file to its archive directory, and exits cleanly when it receives a signal. This test uses an input file that contains only IPv4 data.

20. Categorizing and discarding flows matching an IPv6 IPset. This test is similar to the previous, except the flow records and the IPset files contain IPv6 addresses.

- 21. Bad SiLK input files. Check whether rwflowpack properly handles unusual files in a directory it is polling for SiLK files. One test file is a SiLK data file that contains no records; rwflowpack should move this file to the archive directory and not create any hourly data files. Another file is one that does not contain the SiLK file header. rwflowpack should move this file into its error directory.
- 22. Bad flowcap input files. Check whether rwflowpack, running in "fcfiles" input mode, properly handles unusual files its incoming directory. The first test file is a flowcap file that contains no records; rwflowpack should move this file to the archive directory and not create any hourly data files. Another file is one that does not contain the SiLK file header. rwflowpack should move this file into its error directory. The final test file is a SiLK data file that does not contain the proper header; rwflowpack should move this file into the error directory.
- 23. Bad NetFlow input files. Check whether rwflowpack properly handles unusual files in a directory it is polling for NetFlow v5 files. rwflowpack should treat all these files as invalid and move them to the error directory. The checks include (1) a file that has the correct header and is the correct size but contains a record count of zero, (2) a file that has the correct header but it too small, (3) a file that claims it is NetFlow v8, and (4) a file containing plain text.
- 24. Bad IPFIX input files. Check whether rwflowpack properly handles unusual files in a directory it is polling for IPFIX files. rwflowpack should treat all these files as invalid and move them to the error directory. The checks include a file that has the correct header but contains no records, and a file containing plain text. This test is only invoked when SiLK has been compiled with IPFIX support.

### 7.4 Expected results

The tests may take several minutes to run.

During the tests, you may see the following sorts of output.

• The following indicates a test that successfully passed.

PASS: tests/rwflowpack-version.pl

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/rwflowpack-pack-silk.pl.log.

FAIL: tests/rwflowpack-pack-silk.pl

• The following indicates that the test was skipped. For this skipped test, information about why the test was skipped may be available in the file tests/rwflowpack-pack-silk-ipv6.pl.log.

SKIP: tests/rwflowpack-pack-silk-ipv6.pl

A test can be skipped for two reasons.

1. The test is not applicable. For example, there is no need to test IPv6 functionality if SiLK was not compiled with IPv6 support.

2. A file or application that the test requires is not present. This can occur if you fail to build the SiLK tools prior to testing, so that make check is building the tools and testing them. Some tests use other tools in SiLK suite, and the tests will be skipped if the required tools are not available.

Once all processing stops, you should see output similar to the following to summarize the results of running the tests.

\_\_\_\_\_\_

Testsuite summary for SiLK 3.6.0

\_\_\_\_\_

# PASS: 24 # SKIP: 1 # XFAIL: 0 # FAIL: 0 # XPASS: 0 # ERROR: 0

# TOTAL: 25

\_\_\_\_\_

where

**TOTAL** is the number of tests that were run

PASS is the number of tests that passed

**SKIP** is the number of tests that were skipped

**XFAIL** will always be 0

**FAIL** is the number of tests that failed

XPASS will always be 0

**ERROR** is the number of tests that had a fatal error

When the sum of PASS and SKIP equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

Details about why a test was skipped or why a test failed can be found in the test-suite.log file in the src/rwflowpack directory as well as in the \*.log files in the src/rwflowpack/tests directory.

# 8 Testing rwpollexec

The rwpollexec daemon is used to run a user-defined command on files that appear in a directory which rwpollexec periodically examines for new files. rwpollexec is intended to provide a stand-alone program that operates similarly to the --post-command argument available on rwflowappend.

The tests for rwpollexec are included with the SiLK-3.14.0 source distribution. The tests are invoked by typing make check in the src/rwpollexec directory. (If you type make check at the top of the build tree, the rwpollexec tests will be invoked as make recursively descends into each directory.)

The tests are written in a combination of Perl and Python. The tests will confirm that the rwpollexec daemon can start, notices files, execute subprocesses on those files, send signals to subprocesses, properly dispose of files, and terminate cleanly.

### 8.1 Prerequisites

The tests of rwpollexec require that the following tools are installed on the system:

- Python 2.6 or later
- Perl 5.6 or later
- the Perl module Digest::MD5

### 8.2 Preparation

The test scripts assume you have configured and built rwpollexec and all the libraries it requires. The scripts use the application binary as it exists in the build tree, and the scripts do not require that you install SiLK prior to running the tests.

During many of the tests, a temporary directory is created, and files and subdirectories are created in this directory. The directory is created in the location specified by the TMPDIR environment variable, or in /tmp when the TMPDIR environment variable is not set.

### 8.3 Procedure

- 1. Go to src/rwpollexec subdirectory in the directory tree where you built SiLK.
- 2. Type make check. This will invoke the scripts that check the behavior of rwpollexec.

The rwpollexec tests check the following behaviors:

- 1. Handle processes that exit successfully. Check whether rwpollexec properly handles the case when it invokes a command that completes successfully (exit status is 0). The command is invoked sequentially on each of the two files that exist in rwpollexec's incoming directory when rwpollexec is invoked. rwpollexec should move the files to the archive directory once the command is completed. When rwpollexec receives a signal, it should shut down cleanly.
- 2. Handle processes that exit unsuccessfully. This test is similar to the previous, except in this test the command does not complete successfully (i.e., exits with a non-zero status). In this case, the files should be put into the error directory. When rwpollexec receives a signal, it should shut down cleanly.
- 3. Handle processes that exit due to a signal. This test is similar to the first, except in this test the command is terminated due to a signal. In this case, the files should be put into the error directory. When rwpollexec receives a signal, it should shut down cleanly.

4. Handle "slow" processes. Check whether rwpollexec properly handles subprocesses that do not exit after a period of time, where the subprocesses will exit when they receive a SIGTERM. When rwpollexec is invoked, there are two files in its incoming directory. rwpollexec invokes a command on one file, but the command does not exit. rwpollexec sends a SIGTERM to the command, at which point the command exits with a status of 0. rwpollexec repeats the steps for the second file, and the command exits with a status of 1. The first file should appear in the archive directory, and the second in the error directory. When rwpollexec receives a signal, it should shut down cleanly.

- 5. Handle "hanging" processes. This test is similar to the previous, except the subprocesses do not exit when they receive a SIGTERM. Once rwpollexec sends the SIGTERM and the process fails to exit, rwpollexec sends a SIGKILL to terminate the subprocess. Both input files should be moved to the error directory. When rwpollexec receives a signal, it should shut down cleanly.
- 6. Handle many types of processes sequentially. This test is a combination of all of the above tests. rwpollexec invokes a command on each of the 12 files that exists in its incoming directory. rwpollexec does not invoke the command on the next file until the current command terminates. The command either exits on its own, or rwpollexec must send a signal to the process to terminate it. The input files will be moved to the archive or error directory as appropriate. When rwpollexec receives a signal, it should shut down cleanly.
- 7. **No archive directory.** This test is identical to the previous test, except the archive directory is not used. For this test, files whose commands exit successfully should be removed from the file system.
- 8. Handle many types of processes simultaneously. This test is similar to the previous test, except rwpollexec is allowed to invoke 4 subprocesses simultaneously. The input files will be moved to the archive or error directory as appropriate. When rwpollexec receives a signal, it should shut down cleanly.

### 8.4 Expected results

The tests may take several minutes to run.

During the tests, you may see the following sorts of output.

• The following indicates a test that successfully passed.

PASS: tests/rwpollexec-version.pl

• The following indicates a test has failed. For this failed test, information about why the test failed may be available in the file tests/rwpollexec-killed.pl.log.

FAIL: tests/rwpollexec-killed.pl

Once all processing stops, you should see output similar to the following to summarize the results of running the tests.

Testsuite summary for SiLK 3.6.0

# TOTAL: 11

```
# PASS: 11
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
```

where

**TOTAL** is the number of tests that were run

**PASS** is the number of tests that passed

SKIP is the number of tests that were skipped, and should always be 0 for rwpollexec tests

**XFAIL** will always be 0

FAIL is the number of tests that failed

XPASS will always be 0

ERROR is the number of tests that had a fatal error

When PASS equals TOTAL, the make command exits with a status of 0 to indicate that no test failed.

If either FAIL or ERROR is non-zero, one or more tests failed and the return status of make will be non-zero.

Details about why a test failed can be found in the test-suite.log file in the src/rwpollexec directory as well as in the \*.log files in the src/rwpollexec/tests directory.

# 9 Detail of Analysis Tool Testing

This section provides a detailed listing of the tests that will be invoked when you follow the instructions in Section 2.

### 9.1 Simple help check

The following tests verify the --help switch works.

```
flowcap --help
num2dot --help
rwaddrcount --help
rwappend --help
```

```
rwbag --help
rwbagbuild --help
rwbagcat --help
rwbagtool --help
rwcat --help
rwcompare --help
rwallformats --help
rwrtd2split --help
rwcount --help
rwcut --help
rwfileinfo --help
rwfglob --help
rwfilter --help
rwflowappend --help
rwflowpack --help
rwguess --help
rwpackchecker --help
rwpdu2silk --help
rwgroup --help
rwidsquery --help
rwipaexport --help
```

```
rwipaimport --help
rwipfix2silk --help
rwp2yaf2silk --help
rwsilk2ipfix --help
rwmatch --help
rwnetmask --help
rwgeoip2ccmap --help
rwip2cc --help
rwpmapbuild --help
rwpmapcat --help
rwpmaplookup --help
rwpollexec --help
rwrandomizeip --help
rwrecgenerator --help
rwresolve --help
rwscan --help
rwscanquery --help
rwset --help
rwsetbuild --help
rwsetcat --help
rwsetmember --help
```

```
rwsettool --help
mapsid --help
rwsiteinfo --help
rwcombine --help
rwdedupe --help
rwsort --help
rwsplit --help
rwstats --help
rwstats --legacy-help
rwswapbytes --help
rwtotal --help
rwtuc --help
rwuniq --help
rwreceiver --help
rwsender --help
9.2
      Simple version check
The following tests verify the --version switch works.
flowcap --version
```

num2dot --version

rwaddrcount --version

rwappend --version

rwbag --version rwbagbuild --version rwbagcat --version rwbagtool --version rwcat --version rwcompare --version rwallformats --version  ${\tt rwrtd2split} \ {\tt --version}$ rwcount --version rwcut --version rwfileinfo --version rwfglob --version rwfilter --version rwflowappend --version rwflowpack --version rwguess --version rwpackchecker --version  ${\tt rwpdu2silk} \ {\tt --version}$ rwgroup --version rwidsquery --version rwipaexport --version

rwipaimport --version rwipfix2silk --version rwp2yaf2silk --version rwsilk2ipfix --version rwmatch --version rwnetmask --version rwgeoip2ccmap --version rwip2cc --version rwpmapbuild --version rwpmapcat --version rwpmaplookup --version rwpollexec --version rwrandomizeip --version rwrecgenerator --version rwresolve --version rwscan --version rwscanquery --version rwset --version rwsetbuild --version rwsetcat --version rwsetmember --version

```
rwsettool --version

mapsid --version

rwsiteinfo --version

rwcombine --version

rwdedupe --version

rwsort --version

rwsplit --version

rwstats --version

rwstats --version

rwtotal --version

rwtotal --version

rwtuc --version

rwtuc --version

rwtuc --version
```

## 9.3 Command without arguments

The following tests verify the application does not crash when invoked with no switches or arguments. Most of these tests will result in the application exiting with a non-zero exit status.

flowcap
rwaddrcount
rwappend
rwbag

rwbagbuild

rwbagcat	
rwbagtool	
rwcat	
rwcompare	
rwallformats	
rwrtd2split	
rwcount	
rwcut	
rwfileinfo	
rwfglob	
rwfilter	
rwflowappend	
rwflowpack	
rwguess	
rwpackchecker	
rwpdu2silk	
rwgroup	
rwidsquery	
rwipaexport	
rwipaimport	
rwipfix2silk	

rwp2yaf2silk	
rwsilk2ipfix	
rwmatch	
rwnetmask	
rwgeoip2ccmap	
rwip2cc	
rwpmapbuild	
rwpmapcat	
rwpmaplookup	
rwpollexec	
rwrandomizeip	
rwrecgenerator	
rwscan	
rwset	
rwsetbuild	
rwsetcat	
rwsetmember	
rwsettool	
mapsid	
rwsiteinfo	
rwcombine	

rwdedupe

rwsort

rwsplit

rwstats

rwswapbytes

rwtotal

rwtuc

rwuniq

rwreceiver

rwsender

# 9.4 Command with null input

The following tests verify the application does not crash when invoked with completely empty (null) input. Most of these tests will result in the application exiting with a non-zero exit status.

```
rwcompare data.rwf /dev/null
rwcount </dev/null</pre>
rwcut </dev/null
rwfilter --input-pipe=/dev/null --all=/dev/null
rwgroup --id-fields=3 /dev/null
rwnetmask --sip=prefix-length=24 </dev/null</pre>
rwgeoip2ccmap </dev/null</pre>
rwip2cc </dev/null</pre>
rwpmapbuild </dev/null</pre>
rwpmapcat </dev/null</pre>
rwpmaplookup </dev/null</pre>
rwrandomizeip </dev/null</pre>
rwrecgenerator </dev/null</pre>
rwresolve </dev/null</pre>
rwscan --scan-mode=2 /dev/null
rwset --sip-file=/dev/null </dev/null</pre>
rwsetbuild </dev/null >/dev/null
rwsetcat </dev/null</pre>
rwsetmember 10.x.x.x </dev/null</pre>
rwsettool </dev/null >/dev/null
rwcombine --ignore-fields=1 </dev/null</pre>
```

## 9.5 Command with empty SiLK file

The following tests verify the application works correctly when invoked with a SiLK file that contains no data section.

```
rwfilter --proto=0- --pass=stdout empty.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwgroup --id-fields=3 empty.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsilk2ipfix empty.rwf
| rwipfix2silk -
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwmatch --relate=1,2 empty.rwf empty.rwf -
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwnetmask --nhip-prefix=16 empty.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwrandomizeip --seed=38901 empty.rwf -
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwscan --scan-mode=2 empty.rwf
rwsetbuild /dev/null /tmp/rwscan-empty-input-emptyset
&& rwscan --trw-sip-set=/tmp/rwscan-empty-input-emptyset empty.rwf
rwset --sip-file=stdout empty.rwf
| rwsetcat
rwsetbuild /dev/null
| rwsetcat --net=v4:T,13,17,20/10,14,18
rwsetbuild /dev/null
| rwsetcat --net=v6:T,13,17,20/10,14,18
rwcombine empty.rwf --output-path=/dev/null
        --print-statistics=stdout
rwcombine empty.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
```

```
rwdedupe empty.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
cat /dev/null
| rwsort --fields=9,1 --presorted-input --xargs=-
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --field=9,1 --presorted-input empty.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --field=9,1 empty.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsplit --basename=/tmp/rwsplit-empty-input-empty_input
        --flow-limit=100 empty.rwf
cat /dev/null
| rwstats --fields=dip --count=10 --top --ipv6-policy=ignore
        --presorted-input --xargs=-
rwstats --fields=dip --count=10 --top --ipv6-policy=ignore
        --presorted-input empty.rwf
rwstats --fields=dip --count=10 --top
        --ipv6-policy=ignore empty.rwf
rwswapbytes --big-endian empty.rwf -
| rwfileinfo --no-title --field=byte-order,count-records -
rwswapbytes --little-endian empty.rwf -
| rwfileinfo --no-title --field=byte-order,count-records -
rwswapbytes --little-endian empty.rwf -
                                                                 ١
| rwswapbytes --swap-endian - -
| rwfileinfo --no-title --field=byte-order,count-records -
rwtotal --sport empty.rwf
cat /dev/null
| rwuniq --fields=sport --presorted-input -xargs=-
rwuniq --fields=sport --presorted-input empty.rwf
rwuniq --fields=sport --sort-output empty.rwf
```

## 9.6 Checking successful exit status

The following tests perform a variety of checks. In all cases, the application should exit with a zero exit status.

```
mapsid 99999

mapsid 9999

mapsid S9999

rwcombine empty.rwf

rwdedupe empty.rwf
```

## 9.7 Checking for non-zero exit status

The following tests perform a variety of checks for error conditions. In all cases, the application should exit with a non-zero exit status.

```
rwaddrcount --print-recs

rwaddrcount empty.rwf

touch /tmp/rwappend-null-output-in
&& rwappend --create /tmp/rwappend-null-output-in empty.rwf

rwappend stdout empty.rwf >/dev/null

rwbag --sport-flows=/dev/null

rwbag empty.rwf

rwbagcat --mincounter=101 --maxcounter=99 /dev/null

rwbagcat --minkey=101 --maxkey=99 /dev/null

rwbagtool --mincounter=101 --maxcounter=99 /dev/null

rwbagtool --minkey=101 --maxkey=99 /dev/null

rwbagtool --minkey=101 --maxkey=99 /dev/null
```

```
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:16 --end-date=2009/02/12:14
rwfilter --fail=/dev/null empty.rwf
rwfilter --pass=/dev/null empty.rwf
rwfilter --print-stats empty.rwf
rwfilter --all=/dev/null
rwfilter --proto=1 empty.rwf
rwidsquery --intype=fast
rwidsquery --intype=rule --dry-run
                                                                 ١
        /tmp/rwidsquery-rule-no-date-rule 2>&1
rwipaexport /dev/null
rwipaexport --catalog=my-cat --time=2009/02/14:00:00 /dev/null
rwipaimport /dev/null
rwset --sip=- empty.rwf
| rwipaimport --catalog=my-cat --description=my-description
        --start-time=2009/02/12:00:00
        --end-time=2009/02/14:23:59:59 -
rwnetmask --sip=prefix-length=24
rwnetmask empty.rwf
rwrandomizeip empty.rwf </dev/null</pre>
rwset --sip-file=/dev/null
rwset empty.rwf
rwset --sip-file=stdout empty.rwf
| rwsetmember 10.x.x.x
./rwsettool --sample set1-v4.set >/dev/null
```

```
rwcombine --ignore-fields=1
rwdedupe --ignore-fields=1
rwsort --fields=1
rwsort empty.rwf
rwsplit --flow-limit=100 empty.rwf
rwsplit --basename=/tmp/rwsplit-missing-limit-missing_limit empty.rwf
rwsplit --basename=/tmp/rwsplit-multiple-limit-multiple_limit \
        --ip-limit=200 --flow-limit=900 empty.rwf
rwstats --fields=sip --count=10
rwstats empty.rwf
rwswapbytes --big-endian empty.rwf
rwswapbytes --big-endian
rwswapbytes empty.rwf /dev/null
rwtotal --sport --dport empty.rwf
rwtotal --sport
rwtotal empty.rwf
rwuniq --fields=1
rwuniq empty.rwf
```

## 9.8 Perform a checksum of the output–success

The following tests perform a variety of checks. The output of the command is gathered and compared to a known good checksum (MD5). In all cases, the application should exit with a zero exit status.

```
rwcut --fields=1,2 --no-title --ipv6-policy=ignore
        --ip-format=decimal --no-final-delimiter data.rwf
| num2dot --ip-fields=2,1
rwcut --fields=1,3,2,4,5 --no-title
        --ipv6-policy=ignore data.rwf
rwaddrcount --print-rec --sort-ips --column-separator=/
       --no-final-delimiter data.rwf
rwaddrcount --print-stat --output-path=/dev/null
        --copy-input=stdout data.rwf
| rwaddrcount --print-stat
rwaddrcount --print-rec --sort-ips --delimited=, data.rwf
rwaddrcount --use-dest --print-rec --sort-ips data.rwf
rwaddrcount --use-dest --print-stat data.rwf
rwaddrcount --print-rec --sort-ips --ip-format=decimal
        --max-byte=2000 data.rwf
rwaddrcount --use-dest --print-rec --sort-ips
        --max-packet=20 data.rwf
rwaddrcount --print-ips --sort-ips --ip-format=zero-padded
        --max-record=10 data.rwf
rwaddrcount --print-rec --sort-ips --ip-format=decimal
        --min-byte=2000 data.rwf
rwaddrcount --use-dest --print-rec --sort-ips
        --min-packet=20 data.rwf
rwaddrcount --print-ips --sort-ips --ip-format=zero-padded
        --min-record=10 data.rwf
rwaddrcount --print-rec --use-dest --sort-ips data.rwf
rwaddrcount --print-rec --sort-ips --no-columns
                                                                \
        --no-title data.rwf
rwaddrcount --print-ips --sort-ips --no-title data.rwf
```

```
rwaddrcount --print-rec data.rwf
| sort
rwaddrcount --set-file=stdout data.rwf
| rwsetcat
rwaddrcount --print-stat data.rwf
cat data.rwf
| rwaddrcount --print-rec --use-dest --sort-ips
cp data.rwf /tmp/rwappend-create-exists-out
&& rwappend --create
        /tmp/rwappend-create-exists-out empty.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output /tmp/rwappend-create-exists-out
rwappend --create=data.rwf
       /tmp/rwappend-create-template-out data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output /tmp/rwappend-create-template-out
rwappend --create /tmp/rwappend-create-out empty.rwf data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output /tmp/rwappend-create-out
rwcat --byte-order=little empty.rwf >
        /tmp/rwappend-multiple-file-little-out
&& rwappend
        /tmp/rwappend-multiple-file-little-out empty.rwf data.rwf empty.rwf \
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output /tmp/rwappend-multiple-file-little-out
rwcat --byte-order=big empty.rwf >
        /tmp/rwappend-one-file-big-out
&& rwappend /tmp/rwappend-one-file-big-out data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output /tmp/rwappend-one-file-big-out
rwbag --sport-flows=/dev/null --copy-input=stdout data.rwf
| rwbag --sport-flows=-
| rwbagcat --key-format=decimal
rwbag --bag-file=dip-country,sum-packets,- data-v6.rwf
| rwbagcat --delimited
```

```
rwbag --dip-bytes=stdout data-v6.rwf
                                                                \
| rwbagcat
rwbag --dip-bytes=stdout data.rwf
| rwbagcat
rwbag --dip-flows=stdout data-v6.rwf
| rwbagcat --key-format=zero-padded
rwbag --dip-flows=stdout data.rwf
| rwbagcat --key-format=zero-padded
rwbag --dip-packets=stdout data-v6.rwf
| rwbagcat
rwbag --dip-packets=stdout data.rwf
| rwbagcat --key-format=decimal
rwbag --pmap-file=ip-map-v6.pmap
        --bag-file=dip-pmap:service-host,bytes,- data-v6.rwf
| rwbagcat --pmap-file=ip-map-v6.pmap
rwbag --dport-bytes=- data.rwf
| rwbagcat --key-format=decimal --no-final-delimiter -
rwbag --dport-flow=stdout data.rwf
| rwbagcat --key-format=decimal --delimited
rwbag --dport-flow=stdout data.rwf
| rwbagcat --key-format=decimal
rwbag --dport-packets=stdout data.rwf
| rwbagcat --key-format=decimal --no-columns
rwbag --pmap-file=service-port:proto-port-map.pmap
        --bag-file=dport-pmap:service-port,packets,- data.rwf
| rwbagcat --pmap-file=service-port:proto-port-map.pmap
rwbag --bag-file=flags,records,- data.rwf
                                                                ١
| rwbagcat
rwbag --sport-flow=stdout empty.rwf data-v6.rwf empty.rwf data.rwf \
| rwbagcat --key-format=decimal
rwbag --sport-flow=stdout empty.rwf data-v6.rwf data-v6.rwf
| rwbagcat --key-format=decimal
```

```
rwbag --sport-flow=stdout data.rwf empty.rwf data.rwf
| rwbagcat --key-format=decimal
rwbag --proto-bytes=- data.rwf
| rwbagcat --key-format=decimal --minkey=1 --maxkey=20
        --zero-counts
rwbag --proto-flow=stdout data.rwf
| rwbagcat --key-format=decimal --minkey=1
rwbag --proto-packets=stdout data.rwf
| rwbagcat --key-format=decimal --maxkey=17
rwbag --bag-file=sip-country,bytes,- data.rwf
| rwbagcat
rwbag --bag-file=sensor,sum-packets,- data.rwf
| rwbagcat --delimited
rwbag --sip-bytes=stdout data-v6.rwf
| rwbagcat
rwbag --sip-bytes=stdout data.rwf
| rwbagcat
rwbag --sip-flows=/dev/null --sip-packets=stdout data-v6.rwf
| rwbagcat
rwbag --sip-flows=/dev/null --sip-packets=stdout data.rwf
                                                                 \
| rwbagcat
rwbag --sip-flows=stdout data-v6.rwf
| rwbagcat
rwbag --sip-flows=stdout data.rwf
| rwbagcat
rwbag --sip-packets=stdout --sip-bytes=/dev/null data-v6.rwf
| rwbagcat
rwbag --sip-packets=stdout --sip-bytes=/dev/null data.rwf
| rwbagcat
rwbag --sip-packets=stdout data-v6.rwf
| rwbagcat
```

```
rwbag --sip-packets=stdout data.rwf
| rwbagcat
rwbag --pmap-file=ip-map.pmap
        --bag-file=sip-pmap:service-host,flows,- data.rwf
| rwbagcat --pmap-file=ip-map.pmap
rwbag --sport-bytes=- data.rwf
| rwbagcat --key-format=decimal --delimited=, -
rwbag --sport-flow=stdout data.rwf
| rwbagcat --key-format=decimal --column-separator=,
rwbag --sport-packets=stdout data.rwf
| rwbagcat --key-format=decimal --column-separator=,
        --no-final-delim
cat data.rwf
| rwbag --sport-flows=stdout
| rwbagcat --key-format=decimal
rwbag --bag-file=stime,sum-bytes,stdout data.rwf
| rwbagcat --key-format=iso-time
rwuniq --fields=sport --flows --no-title
        --delimited=, data.rwf
| rwbagbuild --bag-input=stdin --delimiter=,
| rwbagcat --key-format=decimal
rwuniq --fields=sip --flows --no-title data-v6.rwf
| rwbagbuild --bag-input=stdin
| rwbagcat --key-format=decimal
rwuniq --fields=sport --flows --no-title data.rwf
| rwbagbuild --bag-input=stdin
| rwbagcat --key-format=decimal
rwcut --delimited --fields=dip,packets --no-title data-v6.rwf
| rwbagbuild --bag-input=- --key-type=dip-country
| rwbagcat --delimited
rwcut --fields=dip,bytes --no-title data-v6.rwf
| rwbagbuild --bag-input=- --key-type=dip-pmap
        --pmap-file=ip-map-v6.pmap
| rwbagcat --pmap-file=ip-map-v6.pmap
```

```
rwcut --fields=protocol,dport,packets
       --column-sep=, --no-title data.rwf
| rwbagbuild --pmap-file=service-port:proto-port-map.pmap
       --delimiter=, --bag-input=- --key-type=dport-pmap
| rwbagcat --pmap-file=service-port:proto-port-map.pmap
rwcut --integer-tcp-flags --fields=flags
       --delimited --no-title data.rwf
| rwbagbuild --bag-input=- --key-type=flags
| rwbagcat
rwset --sip-file=stdout data.rwf
| rwbagbuild --set-input=stdin --output-path=stdout
| rwbagcat
echo 65535,100
| rwbagbuild --bag-input=stdin --delimiter=, --key-type=sport
        --counter-type=sum-bytes
| rwbagcat
echo 65536,100
| rwbagbuild --bag-input=stdin --delimiter=, --key-type=dport
       --counter-type=sum-bytes
| rwbagcat
echo 255,100
| rwbagbuild --bag-input=stdin --delimiter=,
       --key-type=protocol --counter-type=sum-bytes
| rwbagcat
echo 256,100
| rwbagbuild --bag-input=stdin --delimiter=,
        --key-type=protocol --counter-type=sum-bytes
| rwbagcat
rwcut --no-columns --fields=sip, bytes --no-title data.rwf
| rwbagbuild --bag-input=- --key-type=sip-country
| rwbagcat
rwcut --integer-sensor --fields=sensor,packets
        --no-title data.rwf
| rwbagbuild --bag-input=stdin --key-type=sensor
| rwbagcat --delimited
rwset --sip-file=stdout data.rwf
| rwbagbuild --set-input=stdin --default-count=200
| rwbagcat
```

```
rwset --dip-file=stdout data-v6.rwf
| rwbagbuild --set-input=- --key-type=dip-country
| rwbagcat
rwset --sip-file=- data.rwf
| rwbagbuild --pmap-file=ip-map.pmap --set-input=stdin
        --key-type=sip-pmap
| rwbagcat --pmap-file=ip-map.pmap
rwset --sip-file=stdout data-v6.rwf
| rwbagbuild --set-input=stdin
| rwbagcat
rwset --sip-file=stdout data.rwf
| rwbagbuild --set-input=stdin
| rwbagcat
rwcut --no-final-delimiter --fields=sip --no-title data.rwf
| rwbagbuild --pmap-file=ip-map.pmap --bag-input=-
        --key-type=sip-pmap
                                                                \
| rwbagcat --pmap-file=ip-map.pmap
rwcut --timestamp-format=epoch,no-msec
        --fields=stime,bytes --no-title data.rwf
| rwbagbuild --bag-input=- --key-type=stime
| rwbagcat --key-format=iso-time
rwbag --sip-flows=stdout data-v6.rwf
| rwbagcat --key-format=decimal --bin-ips=binary
rwbag --sip-flows=stdout data.rwf
| rwbagcat --key-format=decimal --bin-ips=binary
rwbag --sip-flows=stdout data-v6.rwf
| rwbagcat --key-format=decimal --bin-ips=decimal
rwbag --sip-flows=stdout data.rwf
| rwbagcat --key-format=decimal --bin-ips=decimal
rwbag --sip-flows=stdout data-v6.rwf
| rwbagcat --key-format=decimal --bin-ips
rwbag --sip-flows=stdout data.rwf
| rwbagcat --key-format=decimal --bin-ips
rwbag --sip-flows=stdout data.rwf
| rwbagcat --network-structure=12TS,12
```

```
rwbag --sip-flows=stdout data.rwf
                                                                \
| rwbagcat --network-structure=ATS
rwbag --sip-flows=stdout data.rwf
| rwbagcat --network-structure
rwbag --bag-file=proto,packet,stdout data.rwf
| rwbagcat --sort-counter=decreasing
cat data.rwf | rwbag --bag-file=proto,packet,-
| rwbagcat --sort-counter=increasing
rwbag --bag-file=sipv4,byte,stdout data.rwf
| rwbagcat --key-format=zero-padded --sort-counter
rwbag --sport-bytes=stdout data.rwf
| rwbagcat --key-format=decimal --mincounter=2000
rwbag --sport-flows=stdout data.rwf
| rwbagcat --key-format=decimal --mincounter=10
rwbag --sport-packets=stdout data.rwf
| rwbagcat --key-format=decimal --mincounter=20
rwbag --sport-bytes=stdout data.rwf
                                                                \
| rwbagcat --key-format=decimal --maxcounter=2000
rwbag --sport-flows=stdout data.rwf
| rwbagcat --key-format=decimal --maxcounter=10
rwbag --sport-packets=stdout data.rwf
| rwbagcat --key-format=decimal --maxcounter=20
rwbagtool --add bag1-v4.bag bag2-v4.bag
| rwbagcat
rwbagtool --add bag1-v6.bag bag2-v6.bag
| rwbagcat
rwbagtool --add bag2-v4.bag bag1-v4.bag
| rwbagcat
rwbagtool --add bag2-v6.bag bag1-v6.bag
| rwbagcat
```

```
rwbagtool --add bag1-v4.bag bag2-v4.bag
| rwbagtool --subtract - bag1-v4.bag bag2-v4.bag
| rwbagcat
rwbagtool --add bag1-v6.bag bag2-v6.bag
| rwbagtool --subtract - bag1-v6.bag bag2-v6.bag
| rwbagcat
rwbag --sport-flows=stdout data.rwf
| rwbagtool --add stdin
| rwbagcat --key-format=decimal
rwbagtool --compare=eq bag1-v4.bag bag3-v4.bag
| rwbagcat
rwbagtool --compare=eq bag1-v6.bag bag3-v6.bag
| rwbagcat
rwbagtool --compare=ge bag1-v4.bag bag3-v4.bag
| rwbagcat
rwbagtool --compare=ge bag1-v6.bag bag3-v6.bag
| rwbagcat
rwbagtool --compare=ge bag2-v4.bag bag1-v4.bag
| rwbagcat
rwbagtool --compare=ge bag2-v6.bag bag1-v6.bag
| rwbagcat
rwbagtool --compare=le bag1-v4.bag bag2-v4.bag
| rwbagcat
rwbagtool --compare=le bag1-v6.bag bag2-v6.bag
| rwbagcat
echo 10.4.0.0/14
| rwsetbuild
| rwbagtool --complement-intersect=- bag2-v4.bag
| rwbagcat
echo 2001:db8:a:4::/62
| rwsetbuild
| rwbagtool --complement-intersect=- bag2-v6.bag
| rwbagcat
```

```
rwbag --sip-flows=stdout data-v6.rwf
| rwbagtool --coverset --ipset-record-version=4
| rwsetcat
rwbag --sip-flows=stdout data.rwf
| rwbagtool --coverset --ipset-record-version=4
| rwsetcat
rwbag --sip-flows=stdout data-v6.rwf
| rwbagtool --coverset
| rwsetcat
rwbag --sip-flows=stdout data.rwf
| rwbagtool --coverset
| rwsetcat
rwbagtool --divide bag1-v4.bag bag3-v4.bag
| rwbagcat
rwbagtool --divide bag1-v6.bag bag3-v6.bag
| rwbagcat
echo 10.4.0.0/14
| rwsetbuild
| rwbagtool --intersect=- bag2-v4.bag
| rwbagcat
echo 2001:db8:a:4::/62
| rwsetbuild
| rwbagtool --intersect=- bag2-v6.bag
| rwbagcat
rwbag --sip-flows=stdout data-v6.rwf
| rwbagtool --invert
| rwbagcat --key-format=decimal
rwbag --sip-flows=stdout data.rwf
| rwbagtool --invert
| rwbagcat --key-format=decimal
rwbag --sport-flows=stdout data.rwf
| rwbagtool --maxcounter=10
| rwbagcat --key-format=decimal
rwbagtool --maximize bag3-v4.bag bag1-v4.bag
| rwbagcat
```

```
rwbagtool --maximize bag3-v6.bag bag1-v6.bag
                                                                \
| rwbagcat
rwbag --sport-flows=stdout data.rwf
| rwbagtool --maxkey=1024
| rwbagcat --key-format=decimal
rwbag --sport-flows=stdout data.rwf
| rwbagtool --mincounter=10
| rwbagcat --key-format=decimal
rwbagtool --minimize bag1-v4.bag bag3-v4.bag
| rwbagcat
rwbagtool --minimize bag1-v6.bag bag3-v6.bag
| rwbagcat
rwbag --sport-flows=stdout data.rwf
| rwbagtool --minkey=1024
| rwbagcat --key-format=decimal
rwbag --sport-flows=stdout data.rwf
| rwbagtool --add --output-path=stdout
| rwbagcat --key-format=decimal
rwbagtool --scalar-multiply=2 bag1-v4.bag
| rwbagcat
rwbagtool --scalar-multiply=2 bag1-v6.bag
| rwbagcat
rwbagtool --subtract bag1-v4.bag bag2-v4.bag
| rwbagcat
rwbagtool --subtract bag1-v6.bag bag2-v6.bag
| rwbagcat
rwbagtool --subtract bag2-v4.bag bag1-v4.bag
| rwbagcat
rwbagtool --subtract bag2-v6.bag bag1-v6.bag
| rwbagcat
rwcat --byte-order=big data-v6.rwf
| rwcut --fields=1-15,20,21,26-29 --timestamp-format=epoch
        --delimited
```

```
rwcat --byte-order=big data.rwf
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
rwcat --byte-order=little data-v6.rwf
| rwcut --fields=1-15,20,21,26-29 --timestamp-format=epoch
        --delimited
rwcat --byte-order=little data.rwf
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
rwcat empty.rwf data.rwf empty.rwf
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
cat data.rwf
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
rwcat --note-add='my command line note' empty.rwf
| rwfileinfo --fields=7,14 -
echo 'my stdin note'
| rwcat --note-file-add=- empty.rwf
| rwfileinfo --fields=7,14 -
rwcat data-v6.rwf
| rwcut --fields=1-15,20,21,26-29 --timestamp-format=epoch
        --delimited
rwcat data.rwf
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
cat data.rwf
| rwcat
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
```

```
ls -1 empty.rwf data.rwf empty.rwf
| rwcat --xargs=stdin
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
ls -1 empty.rwf data.rwf empty.rwf
| rwcat --xargs
| rwcut --fields=1-15,20,21,26-29 --ipv6-policy=ignore
        --timestamp-format=epoch --ip-format=decimal
        --delimited
rwcat --byte-order=big data.rwf
| rwcompare data.rwf -
rwcat --byte-order=big
        --output-path=/tmp/rwcompare-big-big data.rwf
&& rwcompare data.rwf /tmp/rwcompare-big-big
rwcat --byte-order=little data.rwf
| rwcompare - data.rwf
rwfilter --stime=2009/02/13:20:00-2009/02/13:20 --sensor=S2
        --proto=6 --aport=80,8080,443 --pass=stdout data.rwf
| rwallformats --no-invocation --basename=/tmp/sk-teststmp
&& md5 /tmp/sk-teststmp*
rwcount --bin-size=1 --load-scheme=1 data.rwf
rwcount --bin-size=1800 --load-scheme=middle-spike data.rwf
rwcount --bin-size=1800
                                                                \
        --load-scheme=time-proportional data.rwf
rwcount --bin-size=1800 --load-scheme=maximum-volume data.rwf
rwcount --bin-size=1800 --load-scheme=minimum-volume data.rwf
rwcount --bin-size=30 --load-scheme=2 data.rwf
rwcount --bin-size=3600 --load-scheme=end-spike
        --bin-slots data.rwf
rwcount --bin-size=3600 --no-title data.rwf
```

```
rwcount --bin-size=86400 --load-scheme=start-spike
        --timestamp-format=epoch data.rwf
rwcount --bin-size=900 --load-scheme=3 data.rwf
rwcount --bin-size=3600 --load-scheme=1 --column-separator=/
        --no-final-delimiter data.rwf
rwcount --bin-size=3600 --load-scheme=1
        --output-path=/dev/null --copy-input=stdout data.rwf
| rwcount --bin-size=86400 --load-scheme=1
        --timestamp-format=epoch
rwcount --bin-size=3600 --load-scheme=1 --delimited=, data.rwf
rwcount --bin-size=3600 --load-scheme=0
                                                                \
        --end-time=2009/02/14T19:30:00 data.rwf
rwcount --bin-size=3600 --load-scheme=1
        --timestamp-format=default data.rwf
rwcount --bin-size=3600 --load-scheme=1
        --timestamp-format=m/d/y data.rwf
rwcount --bin-size=0.500 --skip-zero --load-scheme=1
        --start-time=2009/02/14T20:00:00 data.rwf
rwcount --bin-size=0.1 --load-scheme=2 data.rwf
rwcount --bin-size=3600
        --load-scheme=1 empty.rwf data.rwf data-v6.rwf empty.rwf
rwcount --bin-size=3600
        --load-scheme=1 data-v6.rwf empty.rwf data-v6.rwf empty.rwf
rwcount --bin-size=3600
        --load-scheme=1 empty.rwf data.rwf empty.rwf data.rwf
rwcount --bin-size=3600 --load-scheme=1 --no-columns
        --no-title data.rwf
rwcount data.rwf
rwsort --fields=stime --reverse data.rwf
| rwcount --load-scheme=1
```

```
rwcount --bin-size=3600 --load-scheme=0
        --start-epoch=2009/02/12T20:00:00
        --end-epoch=2009/02/13T20:00:00 data.rwf
rwcount --bin-size=3600 --load-scheme=0 --skip-zero
        --start-time=2009/02/11T20:30:00 data.rwf
rwcount --bin-size=604800 --load-scheme=0
        --start-time=2009/02/10:00:00:00 data.rwf
rwcount --bin-size=3600 --load-scheme=bin-uniform
        --start-time=2009/02/12T20:30:00 data.rwf
cat data.rwf
                                                                \
| rwcount --bin-size=3600 --load-scheme=1
rwcut --fields=1-5 --ipv6-policy=force data.rwf
rwcut --fields=1-5 --ipv6-policy=ignore data.rwf
rwcut --fields=1-5 data-v6.rwf
rwcut --fields=stype,sip,dtype,dip,dtype --delimited
        --num-recs=10000 data.rwf
rwcut --all-fields --delimited data-v6.rwf
rwcut --all-fields --delimited data.rwf
rwcut --fields=7,6 --column-separator=/ data.rwf
rwcut --fields=5,4,3 --column-separator=, --no-columns data.rwf
rwcut --fields=5 --output-path=/dev/null
        --copy-input=stdout data.rwf
| rwcut --fields=5
rwcut --fields=sip,scc,dip,dcc data-v6.rwf
rwcut --fields=sip,scc,dip,dcc --ipv6=ignore data.rwf
rwcut --delimited data.rwf
rwcut --fields=2 --delimited --ip-format=zero-padded data.rwf
```

```
rwcut --dry-run --ipv6-policy=ignore data.rwf
rwcut --fields=8,initialFlags,sessionFlags data.rwf
rwcut --plugin=flowrate.so
        --fields=bytes,packets,dur,pckts/sec,bytes/sec,bytes/packet,payload-bytes,payload-rate data.rwf
rwfilter --proto=58 --pass=- data-v6.rwf
                                                                \
| rwcut --fields=4,5 --icmp-type-and-code
rwfilter --proto=1 --pass=- data.rwf
| rwcut --fields=4,5 --icmp-type-and-code
rwfilter --proto=58 --pass=- data-v6.rwf
| rwcut --fields=icmpTypeCode
rwfilter --proto=1 --pass=- data.rwf
| rwcut --fields=icmpTypeCode
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwcut --plugin=int-ext-fields.so --delimited
        --fields=ext-ip,ext-port,int-ip,int-port,proto,type
        data.rwf
rwcut --plugin=int-ext-fields.so --delimited
        --incoming-flowtypes=all/in,all/inweb
        --outgoing-flowtypes=all/out,all/outweb
        --fields=ext-ip,ext-port,int-ip,int-port,proto,type
        data-v6.rwf
rwcut --plugin=int-ext-fields.so --delimited
        --incoming-flowtypes=all/in,all/inweb
        --outgoing-flowtypes=all/out,all/outweb
        --fields=ext-ip,ext-port,int-ip,int-port,proto,type
        data.rwf
rwcut --fields=9,11 --timestamp-format=default data.rwf
rwcut --fields=9,11 --timestamp-format=m/d/y,no-msec data.rwf
rwcut --fields=attributes,application data.rwf
rwcut --fields=5 --delimited data-v6.rwf data.rwf
```

```
rwcut --fields=5 --delimited data.rwf data.rwf
rwcut --fields=5,4,3 --no-columns data.rwf
rwcut --fields=5,4,3 --no-final-delimiter data.rwf
rwcut --fields=5,4,3 --no-title < data.rwf</pre>
rwcut --pmap-file=servhost:ip-map-v6.pmap
        --fields=dst-servhost data-v6.rwf
rwcut --pmap-file=servhost:ip-map.pmap
        --fields=dst-servhost data.rwf
rwcut --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --fields=src-service-host,src-service-port,src-service-host,src-service-port data-v6.rwf
rwcut --pmap-file=service-port:proto-port-map.pmap
                                                                 \
        --pmap-file=ip-map.pmap
        --fields=src-service-host,src-service-port,src-service-host,src-service-port data.rwf
rwcut --pmap-file=proto-port-map.pmap
                                                                 ١
        --fields=sval,dval data.rwf
rwcut --pmap-file=ip-map-v6.pmap
        --fields=src-service-host data-v6.rwf
rwcut --pmap-file=ip-map.pmap
        --fields=src-service-host data.rwf
rwcut --python-file=pysilk-plugin.py
        --fields=scc,py-scc,dcc,py-dcc
        --num-recs=10000 data.rwf
rwcut --python-file=pysilk-plugin.py --fields=3-5,lower_port
        --num-recs=10000 data.rwf
rwcut --python-file=pysilk-plugin.py
        --fields=lower_port,lower_port data.rwf
rwcut --python-file=pysilk-plugin.py
        --fields=sip,dip,sport,dport,server_ipv6
        --num-recs=10000 data-v6.rwf
```

```
rwcut --python-file=pysilk-plugin.py
        --fields=sip,dip,server_ip,sport,dport,lower_port_simple,protocol,proto_name \
        --num-recs=10000 --delimited=, data.rwf
rwcut --fields=3-5 --num-recs=3000 data.rwf
rwcut --fields=9,10 --timestamp-format=epoch --num-recs=3000
        --start-rec-num=2000 data.rwf
rwcut --fields=12 --integer-sensor --num-recs=3000
        --end-rec-num=2000 data.rwf
rwcut --fields=sip,dip --delimited=, --num-recs=3000
        --end-rec-num=20000 data.rwf
rwcut --fields=sport,dport --start-rec-num=30000
        --end-rec-num=40000 data.rwf
rwcut --fields=in,out,nhip --delimited=,
        --tail-recs=2000 data.rwf
rwcut --fields=class,type,sensor --tail-recs=2000
        --num-recs=1000 data.rwf
rwcut --fields=dip,dport,sip,sport --delimited
        --tail-recs=1000 --num-recs=2000 data.rwf
rwcut --fields=1 --delimited --ip-format=decimal data.rwf
rwcut --fields=sensor,class,type data.rwf
rwcut --plugin=skplugin-test.so --ipv6-policy=ignore
        --no-columns
        --fields=bytes,copy-bytes,text-bytes,quant-bytes,sip,copy-sipv4,copy-sip data.rwf
cat data.rwf
| rwcut --fields=3-8
rwcut --fields=9 --timestamp-format=epoch
        --no-final-delimiter data.rwf
rwcut --fields=9-11 data.rwf
rwfileinfo --fields=1,5-6 --no-title data.rwf
```

```
cat data.rwf
| rwfileinfo --fields=count-records -
rwfileinfo --fields=command-lines, version data.rwf
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --class=all 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --flowtypes=all/in,all/outweb 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --sensors=4,6-8,10 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --sensors=S4,S6,S7,S8,S10 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/13 --no-summary 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12
        --end-date=2009/02/12:14 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13:00
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234483200
        --sensors=S13 --type=out 2>&1
```

rwfileinfo --fields=count-records data.rwf

```
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13T14:15:16
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234534516
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13 --end-date=2009/02/13
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13:00 --end-date=2009/02/13
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234483200 --end-date=2009/02/13
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13T14:15:16 --end-date=2009/02/13
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234534516 --end-date=2009/02/13
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13 --end-date=2009/02/14
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13:00 --end-date=2009/02/14
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234483200 --end-date=2009/02/14
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13T14:15:16 --end-date=2009/02/14
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234534516 --end-date=2009/02/14
        --sensors=S13 --type=out 2>&1
```

```
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13 --end-date=1234569600
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13:00 --end-date=1234569600
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234483200 --end-date=1234569600
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13T14:15:16 --end-date=1234569600
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234534516 --end-date=1234569600
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13 --end-date=2009/02/13T15:16:17
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13:00
        --end-date=2009/02/13T15:16:17
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234483200 --end-date=2009/02/13T15:16:17 \
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13T14:15:16
        --end-date=2009/02/13T15:16:17
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234534516 --end-date=2009/02/13T15:16:17 \
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13 --end-date=1234538177
        --sensors=S13 --type=out 2>&1
```

```
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13:00 --end-date=1234538177
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234483200 --end-date=1234538177
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=2009/02/13T14:15:16 --end-date=1234538177
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing --no-summary
        --start-date=1234534516 --end-date=1234538177
        --sensors=S13 --type=out 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --type=inweb --sensor=S12 2>&1
rwfglob --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --type=out 2>&1
rwfilter --active-time=2009/02/13:00:00-2009/02/13:00:05
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --any-address=2001:db8:c0:a8::x:c1-ff,c0:x
        --fail=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --any-address=192.168.192-255.x
        --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --any-cidr=2001:db8:c0:a8::c0:0/107,2001:db8:c0:a8::e0:0/107 \
        --fail=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --any-cidr=192.168.192.0/18 --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
rwfilter --pmap-file=ip-map-v6.pmap
        --pmap-any-service-host=dhcp
        --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --pmap-file=ip-map.pmap --pmap-any-service-host=dhcp
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
echo 192.168.192-255.x
| rwsetbuild - -
| rwfilter --anyset=- --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --aport=25 --proto=6 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --application=80 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --attributes=T/T --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --bytes-per-packet=39-60 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --bytes=1-100 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --dcc=xg,xj,xq --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --dcc=xg,xj,xq --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --pmap-file=ip-map-v6.pmap
        --pmap-dst-service-host='internal,internal services'
        --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
```

```
rwfilter --pmap-file=ip-map.pmap
        --pmap-dst-service-host='internal,internal services'
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --dport=25 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --pmap-file=service:proto-port-map.pmap
        --pmap-dst-service=TCP/HTTP,TCP/HTTPS
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --dtype=2 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --duration=1-5 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --etime=2009/02/13:00:00-2009/02/13:00:05
        --pass=stdout data.rwf
 \  \  \, | \  \, {\tt rwcat} \  \, -\hbox{\tt -compression-method=none} \  \, -\hbox{\tt -byte-order=little} \\
        --ipv4-output
rwfilter --data-rootdir=. --print-missing
        --start-date=2009/02/12:12 --end-date=2009/02/12:14
        --sensors=S4,S6,S7,S8,S10 --type=in,outweb
        --all=/dev/null 2>&1
rwfilter --flags-all=R/R --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --flags-init=S/SA --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --flags-session=/F,C/C --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
rwfilter --plugin=flowrate.so --bytes-per-second=100-
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --plugin=flowrate.so --proto=17
        --print-volume-statistics=stdout data.rwf
rwfilter --plugin=flowrate.so --packets-per-second=100-1000
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --icmp-code=3 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --icmp-type=3 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --input-index=10 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --ip-version=4 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --proto=17 --print-volume-statistics=stdout data.rwf
rwfilter --proto=17 --max-fail=200 --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --proto=17 --max-pass=100 --max-fail=200
        --pass=/tmp/rwfilter-max-pass-fail-pass
        --fail=/tmp/rwfilter-max-pass-fail-fail data.rwf
&& rwcut --fields=1-10 --ipv6-policy=ignore
        /tmp/rwfilter-max-pass-fail-pass
        /tmp/rwfilter-max-pass-fail-fail
rwfilter --proto=17 --max-pass=100 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
rwfilter --proto=17 --pass=stdout data.rwf data.rwf data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
       --timestamp-format=epoch
       --values=bytes, packets, records, stime, etime
       --sort-output --delimited --no-titles
rwfilter --not-any-address=2001:db8:c0:a8:x:x:c0-ff:x
       --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --not-any-address=192.168.255,192-254.x
       --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
rwfilter --not-any-cidr=2001:db8:c0:a8::c0:0/106
       --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --not-any-cidr=192.168.192.0/19,192.168.224.0/20,192.168.240.0/21,192.168.248.0/22,192.168.252.0/23,192.168.254.0/2
       --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
echo 192.168.255,192-254.x
| rwsetbuild - -
| rwfilter --not-anyset=- --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
rwfilter --not-saddr=x:x:a:fc-ff::0-ffff:0,1-fab,fad-ffff,fac
       --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --not-saddr=10.252-255.0-255.0,1-100,102-255,101
       --pass=stdout data.rwf
 \  \  \, | \  \, {\tt rwcat --compression-method=none \,\, --byte-order=little} \\
       --ipv4-output
rwfilter --not-scidr=2001:db8:a:fc::/62
       --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
--pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
```

--ipv4-output

```
echo 10.252-255.x.x
| rwsetbuild - -
| rwfilter --not-sipset=- --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --output-index=10 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --packets=1-50 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --plugin=flowrate.so --payload-bytes=0-1000
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --plugin=flowrate.so --payload-rate=1000.4-2000.9
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --proto=17 --print-statistics --print-filenames
        --pass=/dev/null data.rwf 2>&1
rwfilter --proto=17
                                                                ١
        --print-volume-statistics=stdout data-v6.rwf
rwfilter --proto=17 --print-volume-statistics=stdout data.rwf
rwfilter --proto=17 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --python-expr='rec.sport==rec.dport'
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --python-file=pysilk-plugin.py
        --print-volume data.rwf 2>&1
rwfilter --proto=17 --print-volume-statistics=stdout data.rwf
```

```
rwfilter --saddress=2001:db8:a:fc-ff::x:x
        --fail=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --saddress=10.252-255.x.x --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --scc=xz --dcc=xz --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --scc=xz --dcc=xz --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --scc=xa,xb,xc --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --scc=xa,xb,xc --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --scidr=2001:db8:a:fc::/63,2001:db8:a:fe::/63
        --fail=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --scidr=10.252.0.0/15,10.254.0.0/15
        --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --pmap-file=ip-map-v6.pmap
        --pmap-src-service-host=ntp --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --pmap-file=ip-map.pmap --pmap-src-service-host=ntp
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
echo 10.252-255.x.x
| rwsetbuild - -
| rwfilter --sipset=- --fail=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
rwfilter --sport=25 --dport=25 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --pmap-file=proto-port-map.pmap
        --pmap-sport-proto=UDP/DHCP --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --sport=25 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter data.rwf data.rwf --all-dest=stdout
| rwfilter --input-pipe=- --proto=17 --pass=stdout
| rwuniq --fields=1-5 --ipv6-policy=ignore
        \verb|--timestamp-format=epoch|\\
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-titles
rwfilter --stime=2009/02/13:00:00-2009/02/13:00:05
       --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --stype=1 --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --threads=4 --proto=17
        --pass=stdout data.rwf data.rwf data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-titles
echo 25,6
| rwfilter --tuple-file=- --tuple-delim=,
        --tuple-fields=sport,proto --tuple-direction=both
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
echo 25,6
| rwfilter --tuple-file=- --tuple-delim=,
        --tuple-fields=sport,proto --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
echo 25,6
| rwfilter --tuple-file=- --tuple-delim=,
        --tuple-fields=sport,proto --tuple-direction=reverse
        --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --pmap-file=service:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap --pmap-any-service=UDP/NTP
        --pmap-any-service-host=ntp --pass=stdout data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwfilter --pmap-file=service:proto-port-map.pmap
        --pmap-file=ip-map.pmap --pmap-any-service=UDP/NTP
        --pmap-any-service-host=ntp --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --type=in --pass=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
ls -1 data.rwf data.rwf data.rwf
| rwfilter --xargs=stdin --proto=17 --pass=stdout
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-titles
rwguess --print-all small.pdu
rwguess small.pdu
rwguess --top=2 small.pdu
rwpackchecker --print-all data.rwf empty.rwf
rwpackchecker --value max-tcp-bpp=5000
        --allowable-count max-tcp-bpp=2 data.rwf
rwpdu2silk small.pdu
| rwcat --byte-order=big --ipv4-output --compression=none
rwsort --fields=dtype data.rwf
| rwgroup --id-fields=dtype
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
rwsort --fields=stype data.rwf
| rwgroup --id-fields=stype
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=dcc data.rwf
| rwgroup --id-fields=dcc
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=scc data.rwf
| rwgroup --id-fields=scc
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --plugin=flowrate.so --fields=bytes/sec data.rwf
| rwgroup --plugin=flowrate.so --id-fields=bytes/sec
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --plugin=flowrate.so --fields=payload-bytes data.rwf
| rwgroup --plugin=flowrate.so --id-fields=payload-bytes
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --plugin=flowrate.so --fields=pckts/sec data.rwf
| rwgroup --plugin=flowrate.so --id-fields=pckts/sec
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=5,1,3,2,4 data.rwf
| rwgroup --id-fields=5,1,3,2,4
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes,packets,records,stime,etime
        --sort-output --delimited --no-title
rwsort --fields=1 data-v6.rwf
| rwgroup --delta-field=1 --delta-value=64
| rwcat --compression-method=none --byte-order=little
rwsort --fields=1 data.rwf
| rwgroup --delta-field=1 --delta-value=16
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
```

```
rwsort --fields=1,2,9 data-v6.rwf
| rwgroup --id-fields=1,2 --delta-field=9 --delta-value=15
       --summarize --rec-threshold=5
| rwcat --compression-method=none --byte-order=little
rwsort --fields=1,2,9 data.rwf
| rwgroup --id-fields=1,2 --delta-field=9 --delta-value=15
       --summarize --rec-threshold=5
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
rwsort --fields=1,2,9 data.rwf
| rwgroup --id-fields=1,2 --delta-field=9 --delta-value=15
        --summarize
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
rwsort --fields=1,2,9 data.rwf
| rwgroup --id-fields=1,2 --delta-field=9 --delta-value=15
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
rwsort --fields=1,2,9 data-v6.rwf
| rwgroup --id-fields=1,2
| rwcat --compression-method=none --byte-order=little
rwsort --fields=1,2,9 data.rwf
| rwgroup --id-fields=1,2
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwsort --pmap-file=servhost:ip-map-v6.pmap
       --fields=dst-servhost
| rwgroup --pmap-file=servhost:ip-map-v6.pmap
       --id-fields=dst-servhost
--ipv4-output
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=servhost:ip-map.pmap
       --fields=dst-servhost
| rwgroup --pmap-file=servhost:ip-map.pmap
       --id-fields=dst-servhost
| rwcat --compression-method=none --byte-order=little
       --ipv4-output
```

```
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwsort --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --fields=src-service-host,src-service-port
| rwgroup --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --id-fields=src-service-host,src-service-port
| rwcat --compression-method=none --byte-order=little
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map.pmap
        --fields=src-service-host,src-service-port
| rwgroup --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map.pmap
        --id-fields=src-service-host,src-service-port
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=proto-port-map.pmap --fields=sval
| rwgroup --pmap-file=proto-port-map.pmap --id-fields=sval
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwsort --pmap-file=ip-map-v6.pmap --fields=src-service-host
| rwgroup --pmap-file=ip-map-v6.pmap
        --id-fields=src-service-host
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=ip-map.pmap --fields=src-service-host
| rwgroup --pmap-file=ip-map.pmap
        --id-fields=src-service-host
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=3,4,9 data.rwf
| rwgroup --id-fields=3,4 --delta-field=9 --delta-value=15
        --objective
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --python-file=pysilk-plugin.py
        --fields=lower_port data.rwf
| rwgroup --python-file=pysilk-plugin.py
```

```
--id-fields=lower_port
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
cat data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwsort --fields=3 data.rwf
| rwgroup --id-fields=3 --rec-threshold=20
        --group-offset=0.1.0.0
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=3 data.rwf
| rwgroup --id-fields=3
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwidsquery --intype=fast --year=2009 --dry-run
        /tmp/rwidsquery-fast-fast 2>&1
rwidsquery --intype=full --year=2009 --dry-run
        /tmp/rwidsquery-full-full 2>&1
rwidsquery --intype=rule --start-date=2009/02/11:10
        --end-date=2009/02/11:12 --dry-run
        /tmp/rwidsquery-rule-rule 2>&1
rwsilk2ipfix data-v6.rwf
| rwipfix2silk --silk-output=/dev/null
        --log-destination=stderr --print-stat 2>&1
rwsilk2ipfix data.rwf
| rwipfix2silk --silk-output=/dev/null
        --log-destination=stderr --print-stat 2>&1
rwsilk2ipfix data-v6.rwf --ipfix-output=/dev/null
        --print-stat 2>&1
rwsilk2ipfix data.rwf --ipfix-output=/dev/null
        --print-stat 2>&1
rwsilk2ipfix data-v6.rwf
| rwipfix2silk --silk-output=stdout
| rwcat --compression-method=none --byte-order=little
```

```
rwsilk2ipfix data.rwf
| rwipfix2silk --silk-output=stdout
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsilk2ipfix empty.rwf data.rwf empty.rwf
| rwipfix2silk
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
cat data.rwf
| rwsilk2ipfix --ipfix-output=stdout
| rwipfix2silk
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --daddr=192.168.x.x --dport=0-1024
        --pass=stdout data.rwf
| rwsort --fields=1,4,2,3,5,9
        --output-path=/tmp/rwmatch-int-server-incoming
&& rwfilter --saddr=192.168.x.x --sport=0-1024
        --pass=stdout data.rwf
| rwsort --fields=2,3,1,4,5,9
        --output-path=/tmp/rwmatch-int-server-outgoing
&& rwmatch --ipv6-policy=asv4 --time-delta=2.5
        --symmetric-del --relative-del --relate=1,2
        --relate=4,3 --relate=2,1 --relate=3,4 --relate=5,5
        /tmp/rwmatch-int-server-incoming
        /tmp/rwmatch-int-server-outgoing -
| rwcut --plugin=cutmatch.so --ipv6-policy=asv4
        --fields=match,sip,sport,dip,dport,proto,type
rwfilter --daddr=2001:db8:c0:a8::/64 --sport=0-1024
        --pass=stdout data-v6.rwf
| rwsort --fields=1,4,2,3,5,9
        --output-path=/tmp/rwmatch-ext-server-v6-incoming
&& rwfilter --saddr=2001:db8:c0:a8::/64 --dport=0-1024
        --pass=stdout data-v6.rwf
| rwsort --fields=2,3,1,4,5,9
        --output-path=/tmp/rwmatch-ext-server-v6-outgoing
&& rwmatch --time-delta=2.5 --symmetric-del --relative-del
        --relate=2,1 --relate=3,4 --relate=1,2 --relate=4,3
        --relate=5,5 /tmp/rwmatch-ext-server-v6-outgoing
        /tmp/rwmatch-ext-server-v6-incoming -
| rwcat --compression-method=none --byte-order=little
rwfilter --daddr=192.168.x.x --sport=0-1024
        --pass=stdout data.rwf
```

```
| rwsort --fields=1,4,2,3,5,9
        --output-path=/tmp/rwmatch-ext-server-incoming
&& rwfilter --saddr=192.168.x.x --dport=0-1024
        --pass=stdout data.rwf
| rwsort --fields=2,3,1,4,5,9
        --output-path=/tmp/rwmatch-ext-server-outgoing
&& rwmatch --time-delta=2.5 --symmetric-del --relative-del
        --relate=2,1 --relate=3,4 --relate=1,2 --relate=4,3
        --relate=5,5 /tmp/rwmatch-ext-server-outgoing
        /tmp/rwmatch-ext-server-incoming -
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --daddr=2001:db8:c0:a8::/64 --dport=0-1024
        --pass=stdout data-v6.rwf
| rwsort --fields=1,4,2,3,5,9
        --output-path=/tmp/rwmatch-int-server-v6-incoming
&& rwfilter --saddr=2001:db8:c0:a8::/64 --sport=0-1024
        --pass=stdout data-v6.rwf
| rwsort --fields=2,3,1,4,5,9
        --output-path=/tmp/rwmatch-int-server-v6-outgoing
&& rwmatch --time-delta=2.5 --symmetric-del --relative-del
        --relate=1,2 --relate=4,3 --relate=2,1 --relate=3,4
        --relate=5,5 /tmp/rwmatch-int-server-v6-incoming
        /tmp/rwmatch-int-server-v6-outgoing -
| rwcat --compression-method=none --byte-order=little
rwfilter --daddr=192.168.x.x --dport=0-1024
        --pass=stdout data.rwf
| rwsort --fields=1,4,2,3,5,9
        --output-path=/tmp/rwmatch-int-server-incoming
&& rwfilter --saddr=192.168.x.x --sport=0-1024
        --pass=stdout data.rwf
| rwsort --fields=2,3,1,4,5,9
        --output-path=/tmp/rwmatch-int-server-outgoing
&& rwmatch --ipv6-policy=asv4 --time-delta=2.5
        --symmetric-del --relative-del --relate=1,2
        --relate=4,3 --relate=2,1 --relate=3,4 --relate=5,5
        /tmp/rwmatch-int-server-incoming
        /tmp/rwmatch-int-server-outgoing -
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwnetmask --6dip-prefix=64 --6sip-prefix=120 data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwnetmask --6sip-prefix-length=120 data-v6.rwf
                                                                ١
| rwcat --compression-method=none --byte-order=little
```

```
rwnetmask --dip-prefix=16 --sip-prefix=24 data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwnetmask --sip-prefix-length=24 data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
cat data.rwf
| rwnetmask --sip-prefix-length=24
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwip2cc --map-file=fake-cc.pmap --print-ips=0
        --address=10.10.10.10
rwip2cc --map-file=fake-cc.pmap --print-ips=1
        --address=10.10.10.10
rwip2cc --map-file=fake-cc.pmap --address=10.10.10.10
rwcut --fields=sip --ipv6-policy=ignore --no-title
        --delimited data.rwf
| rwip2cc --input-file=-
echo 10.10.10.10
| rwip2cc --map-file=fake-cc.pmap --input-file=- --delimited=,
echo 10.10.10.10
                                                                ١
| rwip2cc --input-file=-
echo 10.10.10.10
| rwip2cc --map-file=fake-cc.pmap --input-file=-
        --integer-ips --column-separator=/
echo 10.10.10.10
| rwip2cc --map-file=fake-cc.pmap --input-file=- --no-columns
echo 10.10.10.10
| rwip2cc --map-file=fake-cc.pmap --input-file=- --print-ips=0
| rwip2cc --map-file=fake-cc.pmap --input-file=- --print-ips=1
echo 10.10.10.10
| rwip2cc --map-file=fake-cc.pmap --input-file=-
        --zero-pad-ips --no-final-delimiter
```

```
echo 10.10.10.10
                                                                \
| rwip2cc --map-file=fake-cc.pmap --input-file=-
/usr/bin/env SILK_ADDRESS_TYPES=address_types.pmap
                                                                \
rwpmapcat --no-cidr --address-types
rwpmapcat --no-cidr --address-types=address_types.pmap
rwpmapcat --no-cidr fake-cc.pmap
rwpmapcat --no-cidr fake-cc-v6.pmap
rwpmapcat --no-cidr --country-codes=fake-cc-v6.pmap
/usr/bin/env SILK_COUNTRY_CODES=fake-cc.pmap
                                                                \
rwpmapcat --no-cidr --country-codes
rwpmapcat --delimited=, --no-cidr --map-file ip-map.pmap
rwpmapcat --ip-label-to-ignore=:: ip-map-v6.pmap
rwpmapcat --ip-label-to-ignore=0.0.0.0 ip-map.pmap
rwpmapcat --ignore-label=external ip-map.pmap
                                                                ١
rwpmapcat --ip-format=decimal --no-columns
        --map-file ip-map.pmap
rwpmapcat --output-type=labels --map-file ip-map.pmap
rwpmapcat --left-justify-labels ip-map.pmap
rwpmapcat --output-type=mapname --map-file ip-map.pmap
rwpmapcat --no-cidr-blocks ip-map-v6.pmap
rwpmapcat --no-cidr-blocks --map-file ip-map.pmap
rwpmapcat --output-type=type --no-titles ip-map.pmap
rwpmapcat ip-map-v6.pmap
rwpmapcat --ip-format=zero-padded
        --output-type=ranges ip-map.pmap
```

```
rwpmapcat ip-map.pmap
rwpmapcat --ignore-label=unknown proto-port-map.pmap
rwpmapcat --output-type=labels --no-title proto-port-map.pmap
rwpmapcat --output-type=mapname proto-port-map.pmap
rwpmapcat --no-titles proto-port-map.pmap
rwpmapcat --output-type=type,mapname
                                                                \
        --map-file=proto-port-map.pmap
rwpmapcat --column-sep=, --map-file=proto-port-map.pmap
cat ip-map.pmap
                                                                ١
| rwpmapcat --map-file=- --no-cidr
cat ip-map.pmap
| rwpmapcat --no-cidr
rwpmaplookup --country-codes=fake-cc.pmap --no-title
        --fields=block,key,value
        --no-files 10.10.10.10 10.200.200.200
rwpmaplookup --map-file=ip-map.pmap --no-title
        --fields=block,key,value
        --no-files 172.16.17.18 172.30.31.32
rwpmaplookup --map-file=ip-map-v6.pmap --no-title
        --fields=block,key,value
        --no-files 2001:db8:ac:10::11:12 2001:db8:ac:1e::1f:20
rwpmaplookup --map-file=proto-port-map.pmap --no-title
                                                                ١
        --fields=block,key,value --no-files 17/0 6/0
echo 6/22 > /tmp/rwpmaplookup-files-proto-port-file1
; echo 6/25 > /tmp/rwpmaplookup-files-proto-port-file2
; echo 6/80 > /tmp/rwpmaplookup-files-proto-port-file3
; rwpmaplookup --map-file=proto-port-map.pmap
        /tmp/rwpmaplookup-files-proto-port-file1
        /tmp/rwpmaplookup-files-proto-port-file2
        /tmp/rwpmaplookup-files-proto-port-file3
```

```
rwcut --fields=sip --ipv6-policy=ignore --no-title
        --num-rec=1000 --delimited data.rwf
| rwsetbuild
| /usr/bin/env SILK_ADDRESS_TYPES=address_types.pmap
  rwpmaplookup --ipset-files --address-types --no-final-delim
rwcut --fields=sip --no-title --start-rec=1000 --num-rec=1000
        --delimited data-v6.rwf
| rwsetbuild
| rwpmaplookup --ipset-files --delimited
        --country-codes=fake-cc-v6.pmap --fields=value,input
rwcut --fields=sip --ipv6-policy=ignore --no-title
        --start-rec=1000 --num-rec=1000 --delimited data.rwf
| rwsetbuild
| rwpmaplookup --country-codes=fake-cc.pmap
        --fields=value,input --delimited --ipset-files
echo 192.168.72.72
| rwsetbuild
| rwpmaplookup --ipset-files --map-file=ip-map.pmap
        --ip-format=decimal --fields=key, value, input
rwcut --fields=sip --no-title --num-rec=200
        --delimited data-v6.rwf
| rwsetbuild - /tmp/rwpmaplookup-ipset-ip-v6-file1
&& rwpmaplookup --map-file=ip-map-v6.pmap
        --ip-format=zero-padded --fields=key,value,input
        --ipset-files /tmp/rwpmaplookup-ipset-ip-v6-file1
/usr/bin/env SILK_ADDRESS_TYPES=address_types.pmap
rwpmaplookup --address-types --column-sep=,
        --no-files 10.10.10.10
/usr/bin/env SILK_COUNTRY_CODES=fake-cc-v6.pmap
rwpmaplookup --country-codes --no-title
        --no-files 2001:db8:a:a::a:a
/usr/bin/env SILK_COUNTRY_CODES=fake-cc.pmap
rwpmaplookup --country-codes --no-title --no-files 10.10.10.10
rwpmaplookup --map-file=ip-map.pmap --no-title
        --no-files 192.168.72.72
rwpmaplookup --map-file=ip-map-v6.pmap
        --no-files 2001:db8:ac:18::ba:d
```

```
rwpmaplookup --map-file=proto-port-map.pmap --no-title
        --no-files 17/67
/usr/bin/env SILK_COUNTRY_CODES=fake-cc.pmap
rwpmaplookup --country-codes --no-title
        --fields=start-block,end-block,value
        --no-files 10.10.10.10 10.200.200.200
rwpmaplookup --map-file=ip-map.pmap --no-title
        --fields=start-block,end-block,value
        --no-files 172.16.17.18 172.30.31.32
rwpmaplookup --map-file=ip-map-v6.pmap --no-title
        --fields=start-block,end-block,value
        --no-files 2001:db8:ac:10::11:12 2001:db8:ac:1e::1f:20
rwpmaplookup --map-file=proto-port-map.pmap --no-title
        --fields=start-block,end-block,value
        --no-files 17/0 6/0
rwcut --fields=sip --ipv6-policy=ignore --no-title
        --num-rec=1000 --delimited data.rwf
| /usr/bin/env SILK_ADDRESS_TYPES=address_types.pmap
  rwpmaplookup --address-types --no-final-delim
rwcut --fields=sip --no-title --start-rec=1000 --num-rec=1000
        --delimited data-v6.rwf
| rwpmaplookup --country-codes=fake-cc-v6.pmap
        --fields=value,input --delimited
rwcut --fields=sip --ipv6-policy=ignore --no-title
        --start-rec=1000 --num-rec=1000 --delimited data.rwf
| rwpmaplookup --country-codes=fake-cc.pmap
        --fields=value,input --delimited
echo 192.168.72.72
| rwpmaplookup --map-file=ip-map.pmap --ip-format=decimal
        --fields=key, value, input
rwcut --fields=sip --no-title --num-rec=200
        --delimited data-v6.rwf
| rwpmaplookup --map-file=ip-map-v6.pmap
        --ip-format=zero-padded --fields=key,value,input
rwpmaplookup --address-types=address_types.pmap
        --fields=value --no-title -delim --no-files 10.10.10.10
```

```
rwpmaplookup --country-codes=fake-cc-v6.pmap --fields=value
                  --no-title -delim --no-files 2001:db8:a:a::a:a
rwpmaplookup --country-codes=fake-cc.pmap --fields=value
                  --no-title -delim --no-files 10.10.10.10
rwpmaplookup --map-file=ip-map.pmap --fields=value
                                                                                                                                                     \
                  --no-title -delim --no-files 192.168.72.72
rwpmaplookup --map-file=ip-map-v6.pmap --fields=value
                   --no-title -delim --no-files 2001:db8:ac:18::ba:d
rwpmaplookup --map-file=proto-port-map.pmap --fields=value
                  --no-title -delim --no-files 17/67
rwrandomizeip --seed=38901 --consistent data.rwf -
| rwcat --compression-method=none --byte-order=little
                  --ipv4-output
rwrandomizeip --seed=38901 --save-table=stdout data.rwf
                  /dev/null
| rwrandomizeip --load-table=stdin data.rwf -
| rwcat --compression-method=none --byte-order=little
                  --ipv4-output
rwrandomizeip --seed=38901 data.rwf stdout
| rwcat --compression-method=none --byte-order=little
                  --ipv4-output
cat data.rwf
| rwrandomizeip --seed=38901 --consistent - -
| rwcat --compression-method=none --byte-order=little
                   --ipv4-output
rwrecgenerator --seed 987654321 --log-dest=none
                  --start-time=2011/01/01:00 --end-time=2011/01/01:01
                   --time-step=1000 --silk-output-path -
| rwcut --ipv6=ignore
                  -- fields = 1-7, 9-12, class, type, initial Flag, session Flag, attribute, application, icmp Type Code and the contraction of the contraction of
rwrecgenerator --seed 987654321 --log-dest=none
                  --start-time=2011/01/01:00 --end-time=2011/01/01:01
                  --time-step=1000 --text-output-path -
echo '0.0.0.0|0.0.0.0|'
| rwresolve --ip-fields=4,8 --column-width=20
```

```
echo '0.0.0.0|0.0.0.0|'
| rwresolve --column-width=20
echo '0.0.0.0,0.0.0.0'
| rwresolve --delimiter=, --column-width=20
echo '0.0.0.0|0.0.0.0'
| rwresolve --column-width=20
echo '0.0.0.0|0.0.0.0|'
| rwresolve --ip-fields=1 --column-width=20
echo '0.0.0.0|0.0.0.0|'
| rwresolve --ip-fields=1,4 --column-width=20
rwfilter --daddr=192.168.0.0/16 --pass=stdout data.rwf
| rwsort --fields=sip,proto,dip - scandata.rwf
| rwscan --scan-mode=2
rwfilter --daddr=192.168.0.0/16
        --pass=/tmp/rwscan-hybrid-in data.rwf
&& rwset --dip=/tmp/rwscan-hybrid-inset /tmp/rwscan-hybrid-in
&& rwsort --fields=sip,proto,dip
        /tmp/rwscan-hybrid-in scandata.rwf
| rwscan --trw-sip-set=/tmp/rwscan-hybrid-inset
rwfilter --daddr=192.168.0.0/16
        --pass=/tmp/rwscan-trw-only-in data.rwf
&& rwset --dip=/tmp/rwscan-trw-only-inset
        /tmp/rwscan-trw-only-in
&& rwsort --fields=sip,proto,dip
        /tmp/rwscan-trw-only-in scandata.rwf
| rwscan --scan-mode=1 --trw-sip-set=/tmp/rwscan-trw-only-inset
rwset --sip-file=/dev/null --copy-input=stdout data.rwf
| rwset --sip-file=-
| rwsetcat --print-ips
rwset --dip-file=stdout data-v6.rwf
| rwsetcat --cidr-blocks=0
rwset --dip-file=stdout data.rwf
| rwsetcat --cidr-blocks=0
rwset --sip-file=stdout empty.rwf data.rwf empty.rwf
| rwsetcat --cidr-blocks=0
```

```
rwset --nhip-file=stdout data-v6.rwf
                                                                 \
| rwsetcat
rwset --nhip-file=stdout data.rwf
| rwsetcat
rwset --sip=stdout --dip=/dev/null data-v6.rwf
| rwsetcat --cidr-blocks=0 --ip-format=hexadecimal
rwset --sip=stdout --dip=/dev/null data.rwf
| rwsetcat --cidr-blocks=0
rwset --sip=/dev/null --dip=stdout data-v6.rwf
| rwsetcat --cidr-blocks=0 --ip-format=decimal
rwset --sip=/dev/null --dip=stdout data.rwf
| rwsetcat --cidr-blocks=0
rwset --sip-file=stdout data-v6.rwf
| rwsetcat --cidr-blocks=0
rwset --sip-file=stdout data.rwf
| rwsetcat --cidr-blocks=0
cat data.rwf
| rwset --sip-file=stdout
| rwsetcat --cidr-blocks=0
rwsetcat --cidr set1-v4.set
| rwsetbuild
| rwsetcat --cidr
rwsetcat --cidr set1-v6.set
| rwsetbuild
| rwsetcat --cidr
rwsetcat --cidr set2-v4.set
| rwsetbuild
| rwsetcat --cidr
rwsetcat --cidr set2-v6.set
| rwsetbuild
| rwsetcat --cidr
```

```
rwset --sip-file=stdout data.rwf
| rwsetcat --cidr-blocks
| rwsetbuild
| rwsetcat
rwsetcat set1-v4.set
| rwsetbuild
| rwsetcat --cidr
rwsetcat set2-v4.set
| rwsetbuild
| rwsetcat --cidr
rwset --sip-file=stdout data-v6.rwf
| rwsetcat --cidr-blocks=0
| rwsetbuild stdin
| rwsetcat --cidr-blocks=0
rwset --sip-file=stdout data.rwf
| rwsetcat
| rwsetbuild stdin
| rwsetcat
rwsetcat --ip-ranges --delim=, set1-v4.set
l cut -d, -f2,3
| rwsetbuild --ip-ranges=,
| rwsetcat --cidr
rwsetcat --ip-ranges --delim=, set1-v6.set
| cut -d, -f2,3
| rwsetbuild --ip-ranges=,
| rwsetcat --cidr
rwsetcat --ip-ranges --delim=, set2-v4.set
l cut -d, -f2,3
| rwsetbuild --ip-ranges=,
| rwsetcat --cidr
rwsetcat --ip-ranges --delim=, set2-v6.set
| cut -d, -f2,3
| rwsetbuild --ip-ranges=,
| rwsetcat --cidr
rwset --sip-file=stdout data.rwf
| rwsetcat --ip-ranges --delim=,
| cut -d, -f2,3
| rwsetbuild --ip-ranges=, - -
| rwsetcat
```

```
rwsetcat --cidr-blocks set1-v4.set
rwsetcat --cidr-blocks set1-v6.set
rwsetcat --cidr-blocks set2-v4.set
rwsetcat --cidr-blocks set2-v6.set
rwset --sip-file=stdout data.rwf
                                                                \
| rwsetcat --cidr-blocks
rwsetcat --count-ips --print-filename=0 set1-v4.set set2-v4.set
rwsetcat --count-ips --print-filenames set1-v4.set
rwsetcat --count-ips set1-v4.set set2-v4.set
rwsetcat --count-ips set1-v4.set
rwsetcat --count-ips set1-v6.set
rwsetcat --count-ips set2-v4.set
rwsetcat --count-ips set2-v6.set
rwset --sip-file=stdout data.rwf
                                                                \
| rwsetcat --count-ips
rwset --sip-file=stdout data.rwf
| rwsetcat --ip-format=hexadecimal stdin
rwset --sip-file=stdout data.rwf
| rwsetcat --ip-format=decimal
rwsetcat --ip-ranges --print-filename=1 set1-v4.set
rwsetcat --ip-ranges set1-v4.set
rwsetcat --ip-ranges --ip-format=zero-padded set1-v6.set
rwsetcat --ip-ranges set2-v4.set
rwsetcat --ip-ranges --ip-format=zero-padded set2-v6.set
```

```
rwset --sip-file=stdout data.rwf
| rwsetcat --ip-ranges
echo 10.0.0.0/8
| rwsetbuild
| rwsetcat --net=v4:T,13,17,20/10,14,18
echo 10.0.0.0/8
| rwsetbuild
| rwsetcat --net=v4:ST,8,13,17,20/10,14,18,7
rwset --sip-file=stdout data.rwf
| rwsetcat --network-structure=12TS,12
rwsetcat --network-structure=18TS,18 set1-v4.set
rwsetcat --network-structure=20TS,20 set2-v4.set
echo 2001:db8::/32
| rwsetbuild
| rwsetcat --net=v6:ST,37,41,44,32/34,38,42,31
echo 2001:db8::/32
| rwsetbuild
| rwsetcat --net=v6:T,37,41,44/34,38,42
rwset --sip-file=stdout data-v6.rwf
| rwsetcat --network-structure=v6:48,T/48,64,123,112
rwset --sip-file=stdout data-v6.rwf
| rwsetcat --network-structure=v6:T60S
rwset --sip-file=stdout data.rwf
| rwsetcat --network-structure=ATS
rwsetcat --network-structure=v6:18TS,18/48,67,56,64 set1-v6.set
rwsetcat --network-structure set1-v4.set
rwsetcat --network-structure=v6: set1-v6.set
rwsetcat --network-structure=v6:60T,60/64,67,48,56 set2-v6.set
rwsetcat --network-structure set2-v4.set
```

```
rwsetcat --network-structure=v6: set2-v6.set
rwsettool --union set3-v4.set set3-v6.set
| rwsetcat --network-structure=v4:8TS
rwset --sip-file=stdout data-v6.rwf
| rwsetcat --network-structure=v6:
rwset --sip-file=stdout data.rwf
| rwsetcat --network-structure
rwsetcat --cidr-blocks=0 set2-v6.set
I head -n 257
rwset --sip-file=stdout data.rwf
| rwsetcat --ip-format=zero-padded stdin
rwsetmember --count 10.0.15.128/25 set1-v4.set set2-v4.set
| sed 's,.*/,,'
rwsetmember
        --count 2001:db8:0:x:x:x:x:x set1-v6.set set2-v6.set
| sed 's,.*/,,'
        --count 2001:db8:0:f:8000::/65 set1-v6.set set2-v6.set
| sed 's,.*/,,'
rwsetmember --count 10.x.x.x set1-v4.set set2-v4.set
                                                                \
| sed 's,.*/,,'
rwset --sip-file=stdout data-v6.rwf
| rwsetmember --count 2001:db8:c0:a8::x:x -
rwset --sip-file=stdout data.rwf
| rwsetmember --count 192.168.x.x -
rwset --sip-file=stdout data-v6.rwf
| rwsetmember 2001:db8:c0:a8::/64 stdin
rwset --sip-file=stdout data.rwf
| rwsetmember 192.168.0.0/16 stdin
rwsettool --difference setb.set seta.set
| rwsetcat --cidr=1
```

<pre>rwsettooldifference seta.set setb.set   rwsetcatcidr=1</pre>	\
<pre>rwsettooldifference setc.set seta.set   rwsetcatcidr=1</pre>	\
<pre>rwsettooldifference seta.set setc.set   rwsetcatcidr=1</pre>	\
<pre>rwsettooldifference set1-v4.set set2-v4.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set1-v6.set set2-v6.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set2-v4.set set1-v4.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set2-v6.set set1-v6.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set3-v4.set set4-v4.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set3-v6.set set4-v6.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set4-v4.set set3-v4.set   rwsetcatcidr</pre>	\
<pre>rwsettooldifference set4-v6.set set3-v6.set   rwsetcatcidr</pre>	\
rwsettoolintersect set1-v4.set set2-v4.set   rwsetcatcidr	\
<pre>rwsettoolintersect set1-v6.set set2-v6.set   rwsetcatcidr</pre>	\
<pre>rwsettoolintersect set2-v4.set set1-v4.set   rwsetcatcidr</pre>	\
<pre>rwsettoolintersect set2-v6.set set1-v6.set   rwsetcatcidr</pre>	١

rwsettoolintersect set3-v4.set set4-v4.set   rwsetcatcidr	\
rwsettoolintersect set3-v6.set set4-v6.set   rwsetcatcidr	\
rwsettoolintersect set4-v4.set set3-v4.set   rwsetcatcidr	\
rwsettoolintersect set4-v6.set set3-v6.set   rwsetcatcidr	\
rwsettoolmask=12 set1-v4.set   rwsetcat	\
<pre>rwsettoolmask=12 set2-v4.set   rwsetcat</pre>	\
<pre>rwsettoolmask=13 set1-v4.set   rwsetcat</pre>	\
<pre>rwsettoolmask=13 set2-v4.set   rwsetcat</pre>	\
rwsettoolmask=14 set1-v4.set   rwsetcat	\
rwsettoolmask=14 set2-v4.set   rwsetcat	\
rwsettoolmask=15 set1-v4.set   rwsetcat	\
rwsettoolmask=15 set2-v4.set   rwsetcat	\
rwsettoolmask=16 set1-v4.set   rwsetcat	\
rwsettoolmask=16 set2-v4.set   rwsetcat	\
rwsettoolmask=17 set1-v4.set   rwsetcat	\

rwsettoolmask=17   rwsetcat	set2-v4.set	\
rwsettoolmask=18   rwsetcat	set1-v4.set	\
rwsettoolmask=18   rwsetcat	set2-v4.set	\
rwsettoolmask=19   rwsetcat	set1-v4.set	١
rwsettoolmask=19   rwsetcat	set2-v4.set	\
rwsettoolmask=20   rwsetcat	set1-v4.set	\
rwsettoolmask=20   rwsetcat	set2-v4.set	\
rwsettoolmask=21   rwsetcat	set1-v4.set	\
rwsettoolmask=21   rwsetcat	set2-v4.set	\
rwsettoolmask=22   rwsetcat	set1-v4.set	\
rwsettoolmask=22   rwsetcat	set2-v4.set	\
rwsettoolmask=23   rwsetcat	set1-v4.set	\
rwsettoolmask=23   rwsetcat	set2-v4.set	\
rwsettoolmask=24   rwsetcat	set1-v4.set	\
rwsettoolmask=24   rwsetcat	set2-v4.set	\

rwsettoolmask=25   rwsetcat	set1-v4.set	\
rwsettoolmask=25   rwsetcat	set2-v4.set	\
rwsettoolmask=26   rwsetcat	set1-v4.set	\
rwsettoolmask=26   rwsetcat	set2-v4.set	\
rwsettoolmask=27   rwsetcat	set1-v4.set	\
rwsettoolmask=27   rwsetcat	set2-v4.set	\
rwsettoolmask=28   rwsetcat	set1-v4.set	\
rwsettoolmask=28   rwsetcat	set2-v4.set	\
rwsettoolmask=29   rwsetcat	set1-v4.set	\
rwsettoolmask=29   rwsetcat	set2-v4.set	\
rwsettoolmask=30   rwsetcat	set1-v4.set	\
rwsettoolmask=30   rwsetcat	set2-v4.set	\
rwsettoolmask=52   rwsetcat	set1-v6.set	\
rwsettoolmask=52   rwsetcat	set2-v6.set	\
rwsettoolmask=53   rwsetcat	set1-v6.set	\

rwsettoolmask=53   rwsetcat	set2-v6.set	\
rwsettoolmask=54   rwsetcat	set1-v6.set	\
rwsettoolmask=54   rwsetcat	set2-v6.set	\
rwsettoolmask=55   rwsetcat	set1-v6.set	\
rwsettoolmask=55   rwsetcat	set2-v6.set	\
rwsettoolmask=56   rwsetcat	set1-v6.set	\
rwsettoolmask=56   rwsetcat	set2-v6.set	\
rwsettoolmask=57   rwsetcat	set1-v6.set	\
rwsettoolmask=57   rwsetcat	set2-v6.set	\
rwsettoolmask=58   rwsetcat	set1-v6.set	\
rwsettoolmask=58   rwsetcat	set2-v6.set	\
rwsettoolmask=59   rwsetcat	set1-v6.set	\
rwsettoolmask=59   rwsetcat	set2-v6.set	\
rwsettoolmask=60   rwsetcat	set1-v6.set	\
rwsettoolmask=60   rwsetcat	set2-v6.set	\

rwsettoolmask=61   rwsetcat	set1-v6.set	\
rwsettoolmask=61   rwsetcat	set2-v6.set	\
rwsettoolmask=62   rwsetcat	set1-v6.set	\
rwsettoolmask=62   rwsetcat	set2-v6.set	١
rwsettoolmask=63   rwsetcat	set1-v6.set	\
rwsettoolmask=63   rwsetcat	set2-v6.set	\
rwsettoolmask=64   rwsetcat	set1-v6.set	\
rwsettoolmask=64   rwsetcat	set2-v6.set	\
rwsettoolmask=65   rwsetcat	set1-v6.set	\
rwsettoolmask=65   rwsetcat	set2-v6.set	\
rwsettoolmask=66   rwsetcat	set1-v6.set	\
rwsettoolmask=66   rwsetcat	set2-v6.set	\
rwsettoolmask=67   rwsetcat	set1-v6.set	\
rwsettoolmask=67   rwsetcat	set2-v6.set	\
rwsettoolmask=68   rwsetcat	set1-v6.set	\

```
rwsettool --mask=68 set2-v6.set
                                                                \
| rwsetcat
rwsettool --mask=69 set1-v6.set
| rwsetcat
rwsettool --mask=69 set2-v6.set
| rwsetcat
rwsettool --mask=70 set1-v6.set
| rwsetcat
rwsettool --mask=70 set2-v6.set
| rwsetcat
rwset --sip-file=stdout data.rwf
| rwsettool --union --output-path=stdout
| rwsetcat
rwset --sip-file=/tmp/sipset data-v6.rwf
rwsettool --sample --ratio=0.02 --seed=2749473 /tmp/sipset
        --output-path=/tmp/sampleset
rwsetcat /tmp/sampleset
rwsettool --intersect /tmp/sipset /tmp/sampleset
| rwsetcat
rwsetcat --count /tmp/sampleset
rwset --sip-file=/tmp/sipset data.rwf
rwsettool --sample --ratio=0.02 --seed=2749473 /tmp/sipset
        --compression=none --invocation-strip
        --output-path=/tmp/sampleset
cat /tmp/sampleset
rwsettool --intersect /tmp/sipset /tmp/sampleset
        --compression=none --invocation-strip
rwsetcat --count /tmp/sampleset
rwset --sip-file=/tmp/sipset data-v6.rwf
rwsettool --sample --size=2000 /tmp/sipset
        --output-path=/tmp/sampleset
rwsetcat /tmp/sampleset
rwsettool --intersect /tmp/sipset /tmp/sampleset
| rwsetcat
rwsettool --sample --size=3000 /tmp/sampleset
| rwsetcat
rwsetcat --count /tmp/sampleset
rwsettool --difference /tmp/sipset /tmp/sampleset
| rwsettool --sample --size=100 - /tmp/sampleset
| rwsetcat --count
```

```
rwset --sip-file=/tmp/sipset data.rwf
rwsettool --sample --size=2000 /tmp/sipset
        --compression=none --invocation-strip
        --output-path=/tmp/sampleset
cat /tmp/sampleset
rwsettool --intersect /tmp/sipset /tmp/sampleset
        --compression=none --invocation-strip
rwsettool --sample --size=3000 /tmp/sampleset
        --compression=none --invocation-strip
rwsetcat --count /tmp/sampleset
rwsettool --difference /tmp/sipset /tmp/sampleset
| rwsettool --sample --size=100 - /tmp/sampleset
| rwsetcat --count
rwsettool --symmetric-difference set1-v4.set set2-v4.set
| rwsetcat --cidr
rwsettool --intersect set1-v6.set set2-v6.set >
        /tmp/rwsettool-symmet-diff-s1-s2-v6-intersect
&& rwsettool --union set1-v6.set set2-v6.set
| rwsettool --difference -
        /tmp/rwsettool-symmet-diff-s1-s2-v6-intersect
| rwsetcat --cidr
rwsettool --intersect set2-v4.set set1-v4.set >
        /tmp/rwsettool-symmet-diff-s2-s1-v4-intersect
&& rwsettool --union set2-v4.set set1-v4.set
| rwsettool --difference -
        /tmp/rwsettool-symmet-diff-s2-s1-v4-intersect
| rwsetcat --cidr
rwsettool --symmetric-difference set2-v6.set set1-v6.set
| rwsetcat --cidr
rwsettool --intersect set3-v4.set set4-v4.set >
        /tmp/rwsettool-symmet-diff-s3-s4-v4-intersect
&& rwsettool --union set3-v4.set set4-v4.set
| rwsettool --difference -
        /tmp/rwsettool-symmet-diff-s3-s4-v4-intersect
| rwsetcat --cidr
rwsettool --symmetric-difference set3-v6.set set4-v6.set
| rwsetcat --cidr
rwsettool --symmetric-difference set4-v4.set set3-v4.set
| rwsetcat --cidr
```

```
rwsettool --intersect set4-v6.set set3-v6.set >
        /tmp/rwsettool-symmet-diff-s4-s3-v6-intersect
&& rwsettool --union set4-v6.set set3-v6.set
| rwsettool --difference -
        /tmp/rwsettool-symmet-diff-s4-s3-v6-intersect
| rwsetcat --cidr
rwsplit --basename=/tmp/v4 --flow-limit=5000 data.rwf
rwsetbuild /dev/null /tmp/v4.sip
rwsetbuild /dev/null /tmp/v4.dip
rwsetbuild /dev/null /tmp/v4.any
for i in /tmp/v4*.rwf; do
    rwset --sip=- $i
    | rwsettool --output=/tmp/v4.sip.union --union
            - /tmp/v4.sip ;
    rwsettool --difference /tmp/v4.sip /tmp/v4.sip.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v4.sip ;
    rwsettool --intersect /tmp/v4.sip.union /tmp/v4.sip
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v4.sip /tmp/v4.sip.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v4.sip.union /tmp/v4.sip ;
    rwset --dip=- $i
    | rwsettool --output=/tmp/v4.dip.union --union
            - /tmp/v4.dip ;
    rwsettool --difference /tmp/v4.dip /tmp/v4.dip.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v4.dip ;
    rwsettool --intersect /tmp/v4.dip.union /tmp/v4.dip
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v4.dip /tmp/v4.dip.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v4.dip.union /tmp/v4.dip ;
    rwset --any=- $i
    | rwsettool --output=/tmp/v4.any.union --union
            - /tmp/v4.any;
    rwsettool --difference /tmp/v4.any /tmp/v4.any.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v4.any ;
    rwsettool --intersect /tmp/v4.any.union /tmp/v4.any
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v4.any /tmp/v4.any.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v4.any.union /tmp/v4.any ;
done
rwsort --fields=sip,dur,proto,sport,dport
    data.rwf data-v6.rwf
| rwsplit --basename=/tmp/v4v6 --flow-limit=10000
```

```
rwsetbuild /dev/null /tmp/v4v6.sip
rwsetbuild /dev/null /tmp/v4v6.dip
rwsetbuild /dev/null /tmp/v4v6.any
for i in /tmp/v4v6*.rwf ; do
    rwset --sip=- $i
    | rwsettool --output=/tmp/v4v6.sip.union --union
           - /tmp/v4v6.sip ;
    rwsettool --difference /tmp/v4v6.sip /tmp/v4v6.sip.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v4v6.sip ;
    rwsettool --intersect /tmp/v4v6.sip.union /tmp/v4v6.sip
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v4v6.sip /tmp/v4v6.sip.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v4v6.sip.union /tmp/v4v6.sip ;
    rwset --dip=- $i
    | rwsettool --output=/tmp/v4v6.dip.union --union
           - /tmp/v4v6.dip ;
    rwsettool --difference /tmp/v4v6.dip /tmp/v4v6.dip.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v4v6.dip ;
    rwsettool --intersect /tmp/v4v6.dip.union /tmp/v4v6.dip
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v4v6.dip /tmp/v4v6.dip.union
    | rwsetcat --cidr-blocks=1;
    mv /tmp/v4v6.dip.union /tmp/v4v6.dip ;
    rwset --any=- $i
    | rwsettool --output=/tmp/v4v6.any.union --union
            - /tmp/v4v6.any;
    rwsettool --difference /tmp/v4v6.any /tmp/v4v6.any.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v4v6.any ;
    rwsettool --intersect /tmp/v4v6.any.union /tmp/v4v6.any
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v4v6.any /tmp/v4v6.any.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v4v6.any.union /tmp/v4v6.any ;
done
rwsplit --basename=/tmp/v6 --flow-limit=5000 data-v6.rwf
rwsetbuild /dev/null /tmp/v6.sip
rwsetbuild /dev/null /tmp/v6.dip
rwsetbuild /dev/null /tmp/v6.any
for i in /tmp/v6*.rwf; do
    rwset --sip=- $i
    | rwsettool --output=/tmp/v6.sip.union --union
           - /tmp/v6.sip ;
    rwsettool --difference /tmp/v6.sip /tmp/v6.sip.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v6.sip ;
```

```
rwsettool --intersect /tmp/v6.sip.union /tmp/v6.sip
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v6.sip /tmp/v6.sip.union
    | rwsetcat --cidr-blocks=1;
    mv /tmp/v6.sip.union /tmp/v6.sip ;
    rwset --dip=- $i
    | rwsettool --output=/tmp/v6.dip.union --union
           - /tmp/v6.dip ;
    rwsettool --difference /tmp/v6.dip /tmp/v6.dip.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v6.dip ;
    rwsettool --intersect /tmp/v6.dip.union /tmp/v6.dip
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v6.dip /tmp/v6.dip.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v6.dip.union /tmp/v6.dip ;
    rwset --any=- $i
    | rwsettool --output=/tmp/v6.any.union --union
            - /tmp/v6.any;
    rwsettool --difference /tmp/v6.any /tmp/v6.any.union
    | rwsetcat --count ;
    rwsetcat --cidr-blocks=1 /tmp/v6.any ;
    rwsettool --intersect /tmp/v6.any.union /tmp/v6.any
    | rwsetcat --cidr-blocks=1 ;
    rwsettool --intersect /tmp/v6.any /tmp/v6.any.union
    | rwsetcat --cidr-blocks=1 ;
    mv /tmp/v6.any.union /tmp/v6.any ;
done
rwsettool --union set1-v4.set set2-v4.set
| rwsetcat --cidr
rwsettool --union set1-v6.set set2-v6.set
| rwsetcat --cidr
rwsettool --union set2-v4.set set1-v4.set
| rwsetcat --cidr
rwsettool --union set2-v6.set set1-v6.set
| rwsetcat --cidr
rwsettool --union set3-v4.set set4-v4.set
| rwsetcat --cidr
rwsettool --union set3-v6.set set4-v6.set
| rwsetcat --cidr
rwsettool --union set4-v4.set set3-v4.set
| rwsetcat --cidr
```

```
rwsettool --union set4-v6.set set3-v6.set
| rwsetcat --cidr
mapsid S9 8 S11 10 S7
mapsid --print-classes
rwsiteinfo --fields=class,type,flowtype,id-flowtype,sensor,id-sensor,describe-sensor,default-class,default-type,mark-default
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=class,default-class
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=class,default-type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=class,sensor
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=class,type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=class
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=default-class,type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=default-class
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=default-type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --delimited='+' --fields=class,type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=flowtype
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=sensor,class --classes=@,bar-class
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=flowtype,class
        --flowtypes=all/type1,bar-class/all,foo-class/type5
        --site-config-file tests/rwsiteinfo-site.conf
```

```
rwsiteinfo --fields=sensor,class --sensors=3-5,17,S
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=class,type --types=type1,0
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --no-title --no-final-delimiter --no-columns
        --fields=class,type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=sensor:list,class:list,type:list
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=sensor,class
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=sensor
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --column-separator='+' --list-delimiter=';'
        --fields=class,type:list
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=type,default-type
        --site-config-file tests/rwsiteinfo-site.conf
rwsiteinfo --fields=type
        --site-config-file tests/rwsiteinfo-site.conf
cat data.rwf
| rwcombine --buffer-size=1m --max-idle-time=0.002
        --output-path=/dev/null --print-statistics=stdout
cat data.rwf
| rwcombine --buffer-size=1m --max-idle-time=0.002
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwcombine --buffer-size=2m --max-idle-time=0.002
        --output-path=/dev/null --print-statistics=stdout
        data-v6.rwf
```

```
rwcombine --buffer-size=2m --max-idle-time=0.002
        data-v6.rwf
| rwuniq --fields=1-5 --ipv6-policy=force
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwcombine data.rwf empty.rwf --max-idle-time=0.002
        --output-path=/dev/null --print-statistics=stdout
rwcombine data.rwf --max-idle-time=0.002
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwcombine empty.rwf data.rwf --output-path=/dev/null
        --print-statistics=stdout
rwcombine data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwdedupe --buffer-size=10m data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwdedupe --ignore-fields=stime data-v6.rwf empty.rwf
| rwuniq --fields=1-5 --ipv6-policy=force
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwdedupe --ignore-fields=stime data.rwf empty.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
echo '2001:db8::5
::1
10.0.0.2
2001:db8::6
```

```
::ffff:10.0.0.2' > tmp/ips
; rwtuc --fields=sip tmp/ips
| rwdedupe --ignore=sport
| rwcut --fields=sip --no-title --delimited
echo '2001:db8::5
::1
10.0.0.2
2001:db8::6
::ffff:10.0.0.2' > tmp/ips
; rwtuc --fields=sip tmp/ips
| rwdedupe
| rwcut --fields=sip --no-title --delimited
rwdedupe data-v6.rwf
| rwuniq --fields=1-5 --ipv6-policy=force
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwdedupe empty.rwf data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwcat data-v6.rwf data-v6.rwf
| rwdedupe
| rwuniq --fields=1-5 --ipv6-policy=force
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwsort --fields=sip,sensor,type,stime data.rwf data.rwf
| rwdedupe
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwsort --fields=dtype data.rwf
 \  \  \, | \  \, rwuniq \  \, \hbox{--fields=dtype --values=dip-distinct --delimited} \\
        --ipv6=ignore --presorted-input
rwsort --fields=stype data.rwf
| rwuniq --fields=stype --values=sip-distinct --delimited
        --ipv6=ignore --presorted-input
```

```
rwsort --fields=bytes data.rwf empty.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=class,type,sensor data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=dcc data-v6.rwf
| rwuniq --fields=dcc --values=distinct:dip --presorted-input
rwsort --fields=dcc data.rwf
| rwuniq --fields=dcc --values=dip-distinct --ipv6=ignore
        --presorted-input
rwsort --fields=scc data-v6.rwf
| rwuniq --fields=scc --values=distinct:sip --presorted-input
rwsort --fields=scc data.rwf
| rwuniq --fields=scc --values=sip-distinct --ipv6=ignore
        --presorted-input
rwsort --fields=dip data-v6.rwf
 \  \  \, | \  \, \text{rwcat --compression-method=none --byte-order=little} \\
rwsort --fields=dip data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=10 data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --plugin=flowrate.so --fields=bytes/sec data.rwf
| rwuniq --plugin=flowrate.so --fields=bytes/sec
        --values=bytes --presorted-input
rwsort --plugin=flowrate.so --fields=payload-bytes data.rwf
                                                                  \
| rwuniq --plugin=flowrate.so --fields=payload-bytes
        --values=bytes,packets,records --presorted-input
rwsort --plugin=flowrate.so --fields=pckts/sec data.rwf
| rwuniq --plugin=flowrate.so --fields=pckts/sec
        --values=packets --presorted-input
```

```
cat data.rwf
| rwsort --field=9,1 --input-pipe=stdin
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwsort --plugin=int-ext-fields.so
        --fields=ext-ip,ext-port data.rwf
| /usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwuniq --plugin=int-ext-fields.so --delimited
        --fields=ext-ip,ext-port --presorted-input
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwsort --plugin=int-ext-fields.so
        --fields=int-ip,int-port data-v6.rwf
| /usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwuniq --plugin=int-ext-fields.so --delimited
        --fields=int-ip,int-port --presorted-input
rwsort --fields=5,1,3,2,4 data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        --timestamp-format=epoch
        --values=bytes, packets, records, stime, etime
        --sort-output --delimited --no-title
rwfilter --sport=20000-25000 --pass=- data.rwf
| rwsplit --basename=/tmp/rwsort-many-presorted-onerec
        --flow-limit=1
&& find 'dirname /tmp/rwsort-many-presorted-onerec' -type f
        -name 'basename /tmp/rwsort-many-presorted-onerec',*'
        -print
| rwsort --fields=sport --presorted-input --xargs=-
| rwcut --fields=sport
rwsort --field=9,1 data.rwf data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --field=9,1 --output-path=stdout data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwsort --pmap-file=servhost:ip-map-v6.pmap
```

```
--fields=dst-servhost
| rwuniq --pmap-file=servhost:ip-map-v6.pmap
        --fields=dst-servhost --presorted-input
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=servhost:ip-map.pmap
        --fields=dst-servhost
| rwuniq --pmap-file=servhost:ip-map.pmap
        --fields=dst-servhost --presorted-input
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwsort --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --fields=src-service-host,src-service-port
| rwuniq --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --fields=src-service-host,src-service-port
        --presorted-input
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map.pmap
        --fields=src-service-host,src-service-port
| rwuniq --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map.pmap
        --fields=src-service-host,src-service-port
        --presorted-input
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=proto-port-map.pmap --fields=sval
| rwuniq --pmap-file=proto-port-map.pmap --fields=sval
        --presorted-input
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwsort --pmap-file=ip-map-v6.pmap --fields=src-service-host
| rwuniq --pmap-file=ip-map-v6.pmap --fields=src-service-host
        --presorted-input
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --pmap-file=ip-map.pmap --fields=src-service-host
| rwuniq --pmap-file=ip-map.pmap --fields=src-service-host
        --presorted-input
rwfilter --proto=6 --pass=-
        --fail=/tmp/rwsort-presorted-data1b data.rwf
| rwsort --field=9,1
        --output-path=/tmp/rwsort-presorted-data1
&& rwfilter --proto=17 --pass=-
```

```
--fail=/tmp/rwsort-presorted-data2b
        /tmp/rwsort-presorted-data1b
| rwsort --field=9,1
        --output-path=/tmp/rwsort-presorted-data2
&& rwsort --field=9,1
        --output-path=/tmp/rwsort-presorted-data3
        /tmp/rwsort-presorted-data2b
&& rwsort --field=9,1 --presorted
        /tmp/rwsort-presorted-data1 empty.rwf
        /tmp/rwsort-presorted-data2 empty.rwf
        /tmp/rwsort-presorted-data3
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=5,3-4 data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwsort --fields=5,3-4 data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --python-file=pysilk-plugin.py
        --fields=lower_port data.rwf
| rwuniq --python-file=pysilk-plugin.py --fields=lower_port
        --values=bytes --presorted-input
rwsort --python-file=pysilk-plugin.py
        --fields=proto_name data.rwf
| rwuniq --python-file=pysilk-plugin.py --fields=proto_name
        --values=bytes --presorted-input
rwsort --python-file=pysilk-plugin.py
        --fields=lower_port_simple data.rwf
| rwuniq --python-file=pysilk-plugin.py
        --fields=lower_port_simple --values=bytes
        --presorted-input
rwsort --python-file=pysilk-plugin.py
        --fields=server_ip data.rwf
| rwuniq --python-file=pysilk-plugin.py --fields=server_ip
        --values=bytes --presorted-input
rwsort --python-file=pysilk-plugin.py
        --fields=server_ipv6 data-v6.rwf
| rwuniq --python-file=pysilk-plugin.py --fields=server_ipv6
        --values=bytes --presorted-input
```

```
rwsort --fields=6 --reverse data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
cat data.rwf
| rwuniq --fields=1-5 --ipv6-policy=ignore
        \verb|--timestamp-format=epoch|\\
        --values=bytes,packets,records,stime,etime
        --sort-output --delimited --no-title
rwsort --fields=1 data-v6.rwf
| rwcat --compression-method=none --byte-order=little
rwsort --fields=1 data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --plugin=skplugin-test.so --fields=copy-bytes data.rwf
| rwuniq --plugin=skplugin-test.so --ipv6-policy=ignore
        --fields=copy-bytes --values=bytes,packets,records
        --presorted-input
rwsort --field=9,1 --sort-buffer-size=10M data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
cat data.rwf
| rwsort --field=9,1
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsort --fields=9,1 empty.rwf data.rwf
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwsplit --basename=$temp --byte-limit=10000000 --seed=737292
        --file-ratio=800 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
rwsplit --basename=$temp --byte-limit=10000000
        --max-outputs=4 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
rwsplit --basename=$temp --byte-limit=10000000 --seed=737292
        --sample-ratio=1000 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
```

```
rwsplit --basename=$temp --flow-limit=10000 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
rwsplit --basename=$temp --ip-limit=5000 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
rwsplit --basename=$temp --packet-limit=10000000 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
rwsplit --basename=$temp --packet-limit=50000 --seed=737292
        --sample-ratio=20 --file-ratio=10 data.rwf
&& rwcat --compression-method=none --byte-order=little
        --ipv4-output $temp*
rwstats --fields=dtype --values=dip-distinct --delimited
        --ipv6=ignore --count=2 --no-percent data.rwf
rwstats --fields=stype --values=sip-distinct --delimited
        --ipv6=ignore --count=2 --no-percent data.rwf
rwstats --fields=etime --bin-time=3600 --values=bytes
        --count=100 data.rwf
rwstats --fields=stime,etime,dur --bin-time=3600
        --values=bytes,packets,flows
        --count=500 data.rwf
rwstats --fields=stime,etime --bin-time=3600
        --values=bytes,packets,flows --count=500 data.rwf
rwstats --fields=stime --bin-time=3600 --values=packets
        --count=100 data.rwf
rwstats --count=10 --fields=dip --column-sep=/ --top
        --ipv6-policy=ignore data.rwf
rwstats --fields=sip --top --count=10 --output-path=/dev/null
        --copy-input=stdout data.rwf
| rwstats --fields=dip --count=10 --ipv6-policy=ignore
rwstats --fields=dcc --values=dip-distinct --count=10
        --no-percent data-v6.rwf
```

```
rwstats --fields=dcc --values=dip-distinct --ipv6=ignore
        --count=10 --no-percent data.rwf
rwstats --fields=scc --values=sip-distinct --count=10
        --no-percent data-v6.rwf
rwstats --fields=scc --values=sip-distinct --ipv6=ignore
        --count=10 --no-percent data.rwf
rwstats --fields=dip --count=10 --delimited=, --top
        --ipv6-policy=ignore data.rwf
rwstats --fields=dip --values=records --percentage=4
        --ipv6-policy=ignore data.rwf
rwstats --fields=dip --values=packets --threshold=25000
        --top data-v6.rwf
rwstats --fields=dip --values=packets --threshold=25000 --top
        --ipv6-policy=ignore data.rwf
rwstats --dip=16 --values=bytes --count=10 --bottom
                                                                ١
        --ipv6-policy=ignore data.rwf
rwfilter --dport=0-66,69-1023,8080 --pass=- data.rwf
| rwstats --fields=dport --bottom --values=bytes --count=20
rwstats --fields=dport --values=dip-distinct,records
        --threshold=5000 --no-percent data.rwf
rwfilter --dport=68 --fail=- data.rwf
| rwstats --fields=proto,dport,iType,iCode --count=16
rwstats --fields=dport --threshold=8000 --top data.rwf
rwstats --plugin=flowrate.so --fields=bytes/sec
                                                                \
        --values=bytes --count=10 data.rwf
rwstats --plugin=flowrate.so --fields=payload-bytes
        --values=bytes,packets,records --count=10 data.rwf
rwstats --plugin=flowrate.so --fields=pckts/sec
                                                                \
        --values=packets --count=10 data.rwf
```

```
rwfilter --dport=68 --fail=- data.rwf
| rwstats --fields=proto,iType,iCode,dport --count=16
rwfilter --proto=1 --pass=- data.rwf
| rwstats --icmp --byte --percentage=5
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwstats --plugin=int-ext-fields.so
        --fields=ext-ip,ext-port --values=packets,records
        --count=35 --delimited data.rwf
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwstats --plugin=int-ext-fields.so
        --fields=int-ip,int-port --values=packets,records
        --count=65 data-v6.rwf
rwstats --fields=sip,dip --values=bytes --count=8 --top
        --ip-format=decimal --ipv6-policy=ignore data.rwf
rwfilter --sport=20000-25000 --pass=- data.rwf
| rwsplit --basename=/tmp/rwstats-many-presorted-onerec
        --flow-limit=1
&& find 'dirname /tmp/rwstats-many-presorted-onerec' -type f
        -name 'basename /tmp/rwstats-many-presorted-onerec',*'
        -print
| rwstats --fields=sport --count=70 --presorted-input
        --values=packets, distinct: sip, flows --xargs=-
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --fields=3-5
        --output-path=/tmp/rwstats-multi-inputs-3-5-pre-in
&& rwfilter --type=in,inweb --fail=stdout data.rwf
| rwsort --fields=3-5
        --output-path=/tmp/rwstats-multi-inputs-3-5-pre-out
&& rwstats --fields=3-5 --values=bytes,packets
        --threshold=30000000 --no-percents --presorted-input
        /tmp/rwstats-multi-inputs-3-5-pre-in
        /tmp/rwstats-multi-inputs-3-5-pre-out
rwfilter --type=in,inweb
        --pass=/tmp/rwstats-multi-inputs-3-5-in
        --fail=/tmp/rwstats-multi-inputs-3-5-out data.rwf
&& rwstats --fields=3-5 --values=bytes,packets
        --threshold=30000000 --no-percents
        /tmp/rwstats-multi-inputs-3-5-in
        /tmp/rwstats-multi-inputs-3-5-out
```

```
rwstats --fields=dport --values=bytes --count=20
        --top empty.rwf data-v6.rwf empty.rwf data.rwf
rwstats --fields=dport --values=bytes --count=20
        --top data-v6.rwf data-v6.rwf empty.rwf
rwstats --fields=dport --values=bytes --count=20
        --top data.rwf empty.rwf data.rwf
rwstats --fields=dip --count=10 --top --no-column
        --column-sep=, --ipv6-policy=ignore data.rwf
rwstats --fields=dip --count=10 --top --no-titles
        --ipv6-policy=ignore data.rwf
rwstats --overall-stats data.rwf
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwstats --pmap-file=servhost:ip-map-v6.pmap
        --fields=dst-servhost --count=10
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwstats --pmap-file=servhost:ip-map.pmap
        --fields=dst-servhost --count=10
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwstats --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --fields=src-service-host,src-service-port --count=10
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwstats --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map.pmap
        --fields=src-service-host,src-service-port --count=10
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwstats --pmap-file=proto-port-map.pmap --fields=sval
        --bottom --count=10
rwfilter --type=in,inweb --pass=stdout data-v6.rwf
| rwstats --pmap-file=ip-map-v6.pmap
        --fields=src-service-host --count=10
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwstats --pmap-file=ip-map.pmap --fields=src-service-host
        --count=10
```

```
rwstats --fields=protocol --values=packets --count=15
        --bottom data.rwf
rwstats --fields=proto --values=distinct:sip,distinct:dip
        --count=5 --no-percent data-v6.rwf
rwstats --fields=proto --values=sip-distinct,dip-distinct
        --count=5 --ipv6=ignore --no-percent data.rwf
rwstats --detail-proto-stats=1 data.rwf
rwstats --fields=protocol --values=packets --count=15 data.rwf
rwstats --python-file=pysilk-plugin.py --fields=lower_port
        --values=max_bytes --count=10 --no-percent data.rwf
rwstats --python-file=pysilk-plugin.py --fields=lower_port
        --value=bytes --count=10 --no-percent data.rwf
rwstats --python-file=pysilk-plugin.py
        --fields=lower_port_simple
        --values=large_packet_flows,largest_packets,smallest_packets \
        --count=5 --no-percent data.rwf
rwstats --python-file=pysilk-plugin.py --fields=sip
        --values=max_bytes --ipv6=ignore --count=10
        --no-percent data.rwf
rwstats --fields=sip,dip --values=bytes --count=8
        --top data-v6.rwf
rwstats --fields=sip,dip --values=bytes --count=8 --top
        --ipv6-policy=ignore data.rwf
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwstats --fields=sport,dport --count=5
rwstats --fields=sip --values=bytes --count=100 --top
        --ipv6-policy=ignore data.rwf
rwstats --fields=sip --percentage=4 --top data-v6.rwf
rwstats --fields=sip --percentage=4 --top
        --ipv6-policy=ignore data.rwf
```

```
rwstats --sip=24 --values=packets --percentage=1 --top
        --ipv6-policy=ignore data.rwf
rwstats --sip=24 --values=packets --percentage=2 --top
        --ipv6-policy=ignore data.rwf
rwstats --plugin=skplugin-test.so --fields=copy-bytes
                                                                 \
        --values=bytes,packets,records --count=10 data.rwf
rwfilter --sport=0-66,69-1023,8080 --pass=- data.rwf
| rwstats --fields=sport --values=records --bottom --count=4
rwstats --fields=sport --values=sip-distinct --threshold=5000
        --no-percent data.rwf
rwstats --fields=sport,sip --values=packets,bytes --count=10
        --ipv6-policy=ignore data.rwf
rwstats --fields=sport --percentage=5 data.rwf
cat data.rwf
                                                                \
| rwstats --fields=dip --top --count=10 --ipv6-policy=ignore
rwswapbytes --big-endian data-v6.rwf stdout
                                                                \
| rwcut --fields=1-15,26-29 --timestamp-format=epoch
rwswapbytes --big-endian data.rwf stdout
| rwcut --fields=1-15,26-29 --ip-format=decimal
        --timestamp-format=epoch --ipv6-policy=ignore
rwswapbytes --little-endian data-v6.rwf -
| rwcut --fields=1-15,26-29 --timestamp-format=epoch
rwswapbytes --little-endian data.rwf -
| rwcut --fields=1-15,26-29 --ip-format=decimal
        --timestamp-format=epoch --ipv6-policy=ignore
cat data.rwf
| rwswapbytes --big - -
| rwcut --fields=1-15,26-29 --ip-format=decimal
        --timestamp-format=epoch --ipv6-policy=ignore
rwswapbytes --swap-endian data.rwf stdout
| rwcut --fields=1-15,26-29 --ip-format=decimal
        --timestamp-format=epoch --ipv6-policy=ignore
```

```
rwtotal --bytes --skip-zero data.rwf
rwtotal --sport --output-path=/dev/null
        --copy-input=stdout data.rwf
| rwtotal --sport --skip-zero
rwtotal --sport --delimited --skip-zero data.rwf
rwtotal --dip-first-16 --skip-zero data.rwf
rwtotal --dip-first-24 --skip-zero data.rwf
rwtotal --dip-first-8 data.rwf
rwtotal --dip-last-16 --skip-zero data.rwf
rwtotal --dip-last-8 data.rwf
rwtotal --dport data.rwf
rwtotal --duration --skip-zero data.rwf
rwfilter --proto=1 --pass=- data.rwf
                                                                \
| rwtotal --icmp-code
rwtotal --sport
        --skip-zero empty.rwf data.rwf data-v6.rwf empty.rwf
rwtotal --sport --skip-zero data-v6.rwf empty.rwf data-v6.rwf
rwtotal --sport --skip-zero data.rwf empty.rwf data.rwf
rwtotal --sport --no-column --column-sep=, data.rwf
rwtotal --sport --no-titles data.rwf
rwtotal --packets --skip-zero data.rwf
rwtotal --proto data.rwf
rwtotal --sip-first-16 --skip-zero data.rwf
rwtotal --sip-first-24 --skip-zero data.rwf
```

```
rwtotal --sip-first-8 data.rwf
rwtotal --sip-last-16 --skip-zero data.rwf
rwtotal --sip-last-8 data.rwf
rwtotal --sport --min-byte=2000 data.rwf
rwtotal --sport --min-packet=20 data.rwf
rwtotal --sport --min-record=10 data.rwf
rwtotal --sport --max-byte=2000 --skip-zero data.rwf
rwtotal --sport --max-packet=20 --skip-zero data.rwf
rwtotal --sport --max-record=10 --skip-zero data.rwf
cat data.rwf
                                                                \
| rwtotal --sport --skip-zero
rwtotal --sport --summation --skip-zero data.rwf
rwcut --fields=sip,dip,sport,dport,proto,packets,bytes,stime,dur,sensor,class,type,in,out,application,initialflags,sessionfl
| rwcat --compression-method=none --byte-order=little
rwcut --fields=sip,dip,sport,dport,proto,packets,bytes,stime,dur,sensor,class,type,in,out,application,initialflags,sessionfl
| rwtuc
| rwcat --compression-method=none --byte-order=little
        --ipv4-output
rwuniq --fields=stype,proto --values=packets
        --sort-output data.rwf
rwuniq --fields=dtype --values=dip-distinct --delimited
        --ipv6=ignore --sort-output data.rwf
rwuniq --fields=stype --values=sip-distinct --delimited
        --ipv6=ignore --sort-output data.rwf
rwuniq --fields=etime --bin-time=3600 --values=bytes
        --sort-output data.rwf
```

```
rwuniq --fields=stime,etime,dur --bin-time=3600
        --values=bytes, packets, flows
        --sort-output data.rwf
rwuniq --fields=stime,etime --bin-time=3600
        --values=bytes,packets,flows --sort-output data.rwf
rwuniq --fields=stime --bin-time=3600 --sort-output data.rwf
rwuniq --fields=sensor,class,type --sort-output data.rwf
rwuniq --fields=sport --output-path=/dev/null
        --copy-input=stdout data.rwf
| rwuniq --fields=sport --sort-output
rwuniq --fields=dcc --values=distinct:scc
        --sort-output data-v6.rwf
rwuniq --fields=dcc --values=dip-distinct --ipv6=ignore
        --sort-output data.rwf
rwuniq --fields=scc --values=distinct:dcc
        --sort-output data-v6.rwf
rwuniq --fields=scc --values=sip-distinct --ipv6=ignore
        --sort-output data.rwf
rwuniq --fields=sport --delimited --sort-output data.rwf
rwuniq --fields=2 --ipv6-policy=ignore --ip-format=decimal
                                                                \
        --bytes --sort-output data.rwf
rwuniq --fields=2 --values=packets --ipv6-policy=force
        --sort-output data-v6.rwf
rwuniq --fields=dip --ipv6-policy=ignore
        --ip-format=zero-padded --packets
        --sort-output data.rwf
rwuniq --fields=dport --all-counts --sort-output data.rwf
rwuniq --fields=dport,iType,iCode,proto --sort-output data.rwf
rwuniq --fields=dur --bytes --sort-output data.rwf
```

```
rwuniq --fields=etime --timestamp-format=epoch
                                                                \
        --sort-output data.rwf
rwuniq --plugin=flowrate.so --fields=bytes/sec --values=bytes
        --sort-output data.rwf
rwuniq --plugin=flowrate.so --fields=payload-bytes
                                                                \
        --values=bytes,packets,records --sort-output data.rwf
rwuniq --plugin=flowrate.so --fields=pckts/sec
                                                                ١
        --values=packets --sort-output data.rwf
rwuniq --fields=iType,iCode,dport,proto --sort-output data.rwf
rwfilter --proto=1 --pass=- data.rwf
                                                                \
| rwuniq --fields=icmpTypeCode --sort-output
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwuniq --plugin=int-ext-fields.so
        --fields=ext-ip,ext-port --sort-output
        data.rwf
/usr/bin/env INCOMING_FLOWTYPES=all/in,all/inweb
        OUTGOING_FLOWTYPES=all/out,all/outweb
rwuniq --plugin=int-ext-fields.so
        --fields=int-ip,int-port --sort-output
        data-v6.rwf
rwuniq --fields=9,11 --timestamp-format=default
        --sort-output data.rwf
rwuniq --fields=9,11 --timestamp-format=m/d/y
        --sort-output data.rwf
rwfilter --sport=20000-25000 --pass=- data.rwf
| rwsplit --basename=/tmp/rwuniq-many-presorted-onerec
        --flow-limit=1
&& find 'dirname /tmp/rwuniq-many-presorted-onerec' -type f
        -name 'basename /tmp/rwuniq-many-presorted-onerec',*,
| rwuniq --fields=sport --values=packets,flows,distinct:sip
        --presorted-input --xargs=-
rwuniq --fields=sport
        --sort-output empty.rwf data.rwf empty.rwf
```

```
rwuniq --fields=sport --no-column --column-sep=,
                                                                \
        --sort-output data.rwf
rwuniq --fields=sport --no-titles --sort-output data.rwf
rwuniq --pmap-file=servhost:ip-map-v6.pmap
        --fields=dst-servhost --sort-output data-v6.rwf
rwuniq --pmap-file=servhost:ip-map.pmap --fields=dst-servhost
        --sort-output data.rwf
rwuniq --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map-v6.pmap
        --fields=src-service-host,src-service-port
        --sort-output data-v6.rwf
rwuniq --pmap-file=service-port:proto-port-map.pmap
        --pmap-file=ip-map.pmap
        --fields=src-service-host,src-service-port
        --sort-output data.rwf
rwuniq --pmap-file=proto-port-map.pmap --fields=sval
                                                                ١
        --sort-output data.rwf
rwuniq --pmap-file=ip-map-v6.pmap --fields=src-service-host
        --sort-output data-v6.rwf
rwuniq --pmap-file=ip-map.pmap --fields=src-service-host
        --sort-output data.rwf
rwfilter --type=in,inweb --pass=stdout data.rwf
| rwsort --fields=3-5
        --output-path=/tmp/rwuniq-ports-proto-multi-pre-in
&& rwfilter --type=in,inweb --fail=stdout data.rwf
| rwsort --fields=3-5
        --output-path=/tmp/rwuniq-ports-proto-multi-pre-out
&& rwuniq --fields=3-5 --presorted-input --no-title
        /tmp/rwuniq-ports-proto-multi-pre-in
        /tmp/rwuniq-ports-proto-multi-pre-out
rwsort --fields=3-5 data.rwf
| rwuniq --fields=3-5 --presorted-input --no-title
rwuniq --fields=sport,dport,proto --no-title
        --sort-output data-v6.rwf
```

```
rwuniq --fields=sport,dport,proto --no-title
        --sort-output data.rwf
rwuniq --fields=3-5 --no-title data.rwf
| sort
rwuniq --fields=proto --sort-output data.rwf
rwuniq --python-file=pysilk-plugin.py --fields=lower_port
        --values=max_bytes --sort-output data.rwf
rwuniq --python-file=pysilk-plugin.py --fields=lower_port
        --value=bytes --sort-output data.rwf
rwuniq --python-file=pysilk-plugin.py
        --fields=lower_port_simple
        --values=large_packet_flows,largest_packets,smallest_packets \
        --sort-output data.rwf
rwuniq --python-file=pysilk-plugin.py --fields=sip
        --values=max_bytes --ipv6=ignore --sort-output data.rwf
rwuniq --fields=sip --values=bytes --sort-output data-v6.rwf
rwuniq --fields=sip --bytes --ipv6-policy=ignore
                                                                \
        --sort-output data.rwf
rwuniq --plugin=skplugin-test.so --ipv6-policy=ignore
        --no-column --fields=protocol
        --values=bytes,sum-bytes,min-bytes,max-bytes,weird-bytes \
        --sort-output data.rwf
rwsort --fields=sport data.rwf
| rwuniq --fields=sport --sip-distinct --dip-distinct
        --presorted-input --ipv6-policy=ignore
rwuniq --fields=sport --sip-distinct --dip-distinct
        --sort-output data-v6.rwf
rwuniq --fields=sport --sip-distinct --dip-distinct
        --ipv6-policy=ignore --sort-output data.rwf
rwsort --fields=sport data-v6.rwf
| rwuniq --fields=sport --sip-distinct --presorted-input
```

```
rwsort --fields=sport data.rwf
| rwuniq --fields=sport --sip-distinct --presorted-input
        --ipv6-policy=ignore
rwuniq --fields=sport --sip-distinct --sort-output data-v6.rwf
rwuniq --fields=sport --sip-distinct --sort-output data.rwf
rwuniq --fields=sport --values=distinct:sip,distinct:dip
                                                                ١
        --sort-output data-v6.rwf
rwuniq --fields=sport --bytes=2000 --sort-output data.rwf
rwuniq --fields=sport --packets=20 --sort-output data.rwf
rwuniq --fields=sport --flows=10 --sort-output data.rwf
rwuniq --fields=sport --bytes=0-2000 --sort-output data.rwf
rwuniq --fields=sport --packets=0-20 --sort-output data.rwf
rwuniq --fields=sport --flows=0-10 --sort-output data.rwf
cat data.rwf
                                                                ١
| rwuniq --fields=sport --sort-output
rwuniq --fields=stime --packets --flows --sort-output data.rwf
rwuniq --fields=stime,proto --bin-time=86400
                                                                \
        --sort-output data.rwf
```

## 9.9 Perform a checksum of the output-failure

The following tests perform a variety of checks for error conditions. The output of the command is gathered and compared to a known checksum (MD5). In all cases, the application should exit with a non-zero exit status.

```
rwcut --fields=sport,dport --start-rec-num=300
        --tail-recs=100 data.rwf 2>&1
rwcut --fields=sport,dport --end-rec-num=300
        --tail-recs=100 data.rwf 2>&1
rwcut --fields=sport,dport --tail-recs=0 data.rwf 2>&1
rwcut --fields=sport,dport --start-rec-num=0 data.rwf 2>&1
rwflowpack ----sensor-conf=sk-teststmp-sensor.conf
        --verify-sensor 2>&1"
rwpackchecker --value max-tcp-bpp=5000
        --allowable-count max-tcp-bpp=1 data.rwf
rwpackchecker --value match-sport=123
        --value match-dport=123 data.rwf
echo 172.16-31.x.x
| rwsetbuild - -
| rwpackchecker --value match-sip=- data.rwf
rwgroup --id-field=3 --delta-value=10 empty.rwf 2>&1
rwgroup --delta-field=9 empty.rwf 2>&1
rwgroup --id-fields=3 data.rwf empty.rwf 2>&1
rwset --sip=- empty.rwf
| rwipaimport --catalog=my-cat --description=my-description
        --start-time=2009/02/12:00:00 - 2>&1
rwset --sip=- empty.rwf
| rwipaimport --catalog=my-cat --description=my-description
        --end-time=2009/02/14:23:59:59 - 2>&1
rwmatch --relate=1,2 data.rwf 2>&1
rwmatch --relate=1,2 data.rwf data.rwf 2>&1
rwscan empty.rwf 2>&1
```

## 9.10 Comparing checksums

The following tests perform a variety of checks. Multiple commands are run and the output of those commands are gathered. The checksum (MD5) of the outputs are compared to ensure the outputs are identical.

```
rwcat --byte-order=little empty.rwf
| rwfileinfo --fields=byte-order --no-title -
rwfileinfo --fields=3 --no-title empty.rwf

rwcat --compression-method=none empty.rwf
| rwfileinfo --fields=compression --no-title -
rwfileinfo --fields=4 --no-title empty.rwf
```