CERT | Software Engineering Institute Carnegie Mellon

SiLK Tool Suite Quick Reference

October 2011 for SiLK v 3.0

Black tools produce flow binary.
Green tools produce bag binary.
Blue tools produce pcap binary.
Purple tools produce IP set binary.
Orange tools produce other binary formats.

SiLK Flow Record Fields

#	Name	Description	
1	sip	Source IP address	Five Tuple (key for flow)
2	dip	Destination IP address	
3	sport	Source port	
4	dport	Destination port	
5	proto	Protocol	
6	packets	Packets	
7	bytes	Bytes	
8	flags	TCP flags in all packets	
9	stime	Start time	
10	dur	Duration	
11	etime	End time	
12	sensor	Sensor number	
13	in	(Unused)	
14	out	(Unused)	
15	nhip	(Used for marking)	
16	stype	pmap index for source IP address	
17	dtype	pmap index for destination IP address	
18	scc	Country code of source IP address	
19	dcc	Country code of destination IP address	
20	class	(Set by configuration, by default, `all`)	
21	type	Flow category (`in`, `out`, `inweb`, `outweb`, etc.)	
22	stime+msec	(Same as stime)	
23	dur+msec	(Same as dur)	
24	etime+msec	(Same as etime)	
25	icmptypecode	ICMP type & code	
26	initialflags	TCP flags for first packet	
27	sessionflags	TCP flags for later packets	
28	attributes	Termination conditions	
29	application	Service recognition	
	src-mapname	Label for source IP from mapname	
	dst-mapname	Label for destination IP from mapname	

This material is based upon work supported by the United States Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

SiLK Parameter Formats

Parameter order is up to the user except that parameters created via pmaps and plugins must be defined before they are referenced.

General Parameter Formats

--name=argument

Where name may be shortened to the minimum prefix not shared with another parameter

filename

Where name follows Linux path formats, or may be `stdin` or `stdout` (as appropriate), or named pipe

Argument Formats

Attr-mask	*High/Care,* where both *High* and *Care* are a series of `FTCS` F = additional packets after FIN, T = active timeout, C = continued flow, S = equal size packets
Cc-list	Comma-separated list of top-level country code abbreviations
Cidr-list	Comma-separated list of IP addresses (in dotted-decimal notation) or CIDR blocks
Date	YYYY/MM/DD:HH or YYYY/MM/DD
Decimal	Any non-negative decimal number
Dec-range	*Decimal-Decimal* or *Decimal-*
Dirname	Local or full path naming directory
Fieldlist	Comma-separated list of field names or *Int-range*
Flag-mask	TCP flags as *High/Care* or comma-separated list of *High/Care* Where both *High* and *Care* are a series (no separator) of `SFARPECU`
Integer	Any positive whole number, range specified by context
Int-range	*Integer-Integer* or *Integer-*
Int-list	Comma-separated list of *Int-range* or *Integer*
Ip-addr	A series of exactly four dot-separated *Int-list* or `x`, or a CIDR block
Sensors	Comma-separated list of sensor names or *Int-range*
String	Sequence of characters between quotes
Time	YYYY/MM/DD:HH or YYYY/MM/DD:HH:MM or YYYY/MM/DD:HH:MM:SS or YYYY/MM/DD:HH:MM:SS.mmm
Time-range	*Time-Time* (both must be same format)

Compression Options (Comp. Opt.)

none	No compression
zlib	Best compression, slower performance
lzo1x	Lesser compression, better performance (default)
best	Implementation defined (currently *lzo1x*)

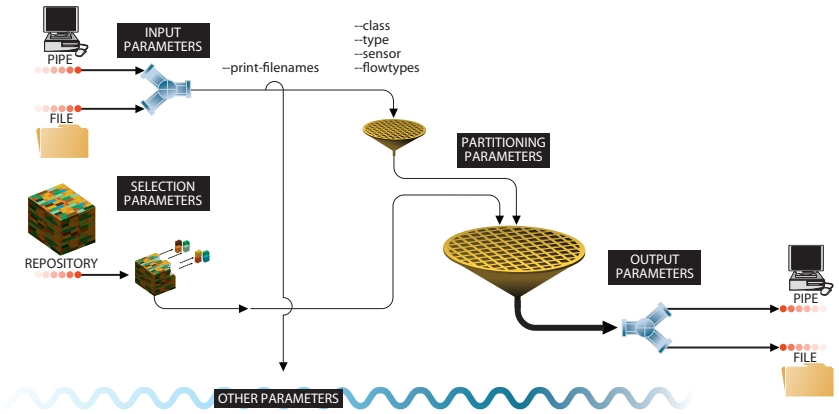
rwfilter

Retrieve flow records from pipe, file, or repository; select records of interest; and store to pipe or file.

Syntax summary: (input or selection [not both], partitioning, and output are required)

```
rwfilter input selection partitioning output other
```

Functional flow diagram:



Examples:

Address block for eight hours:

```
rwfilter --start=2011/04/15:00 --end=2011/04/15:07 \
  --sensor=SEN1 --type=out --daddress=10.5.x.x \
  --pass=10-5.raw
```

All inbound traffic for 15 minutes:

```
rwfilter --start=2011/04/15:00 --sensor=SEN1 \
  --type=in,inweb \
  --stime=2011/04/15:00:00-2011/04/15:00:15 \
  --pass=first-quarter.raw
```

Isolate complete TCP from a file:

```
rwfilter all-outbound.raw --proto=6 \
  --flags-all=SAF/SARF,SAR/SARF \
  --packets=4- --bytes-per=65- --pass=comp-tcp.raw
```

`rwfilter` Parameters

Min-name	Description	Arguments
	Input Parameters	
data	Root directory of repository	dirname
	Flow file to filter (no parameter prefix)	
input-p	Read SiLK flow records from a pipe	filename
site-con	Location of the site configuration file	filename
xarg	Read input file names from file or pipe	filename (opt.)
	Selection Parameters	
class	Class of data to process	`all`
end	Final hour of data to process	hour
flowtype	Class/type pairs to process	`all`/type
sensor	Sensors to process	sensors
start	First hour of data to process	hour
type	Types of data to process	type
	Output Parameters	
all	Destination for all records	filename
fail	Destination for records that fail	filename
pass	Destination for records that pass	filename
print-m	Print the names of missing files	none
print-s	Print a count of total flows	filename (opt.)
print-v	Print count of flows/packets/bytes	filename (opt.)
	Other Parameters	
compress	Set compression for output	comp. opt.
dry-run	Report command line errors	none
help	Print command info	none
max-f	Write <=Arg records to fail	integer (0=all)
max-p	Write <=Arg records to pass	integer (0=all)
note-a	Put arg in file header	string
note-f	Put content of arg in file header	filename
plug	Use plugin to filter records	filename
print-f	Print names of input files	none
thread	Set filter threads	integer
version	Print version	none

`rwfilter` Parameters (continued)

Min-name	Description	Arguments
	Partitioning Parameters	
	also class, flowtype, type, sensor from selection	
active	Flow active during this time window	time-range
any-a	Src or dest address matches IP (and not-)	ip-addr
any-c	Src or dest address in this list (and not-)	cidr-list
anyset	Src or dest address is in this IP set (and not-)	filename
aport	Source or destination port in this list	int-list
app	Packet signature in list	int-list
attrib	Attribute field matches list	attr-mask
bytes	Byte count within this range	int-range
bytes-p	Byte-per-packet count within this range	dec-range
daddr	Destination address matches IP (and not-)	ip-addr
dcc	Destination address maps country code in list	cc-list
dcidr	Destination address in this list (and not-)	cidr-list
dipset	Destination address is in this IP set (and not-)	filename
dport	Destination port in this list	int-list
dtype	Destination address matches classification	integer
dur	Duration in seconds within this range	dec-range
etime	Ending time within this time window	time-range
flags-a	TCP flags match list	flags-mask
flags-i	Initial TCP flags match list	flags-mask
flags-s	Session TCP flags match list	flags-mask
icmp-c	ICMP or ICMPv6 code is in this list	int-list
icmp-t	ICMP or ICMPv6 type is in this list	int-list
ip-v	IP version in list	int-list
ipa-a	Src or dest address matches expression	string
ipa-d	Destination address matches expression	string
ipa-s	Source address matches expression	string
next	nhIP field matches IP (and not-)	ip-addr
nhcidr	nhIP field in this list (and not-)	cidr-list

`rwfilter` Parameters (continued)

Min-name	Description	Arguments
	Partitioning Parameters	
nhipset	nhIP field is in this IP set (and not-)	filename
packet	Packet count within this range	int-range
pmap-dst-MAPNAME	Destination address matches expression	string
pmap-f	Prefix map file to read	filename
pmap-src-MAPNAME	Source address matches list	string
proto	Protocol in this list	int-list
python-e	Run expression	string
python-f	Use Python code to extend processing	filename
saddr	Source address matches IP (and not-)	ip-addr
scc	Source address maps country code in list	cc-list
scidr	Source address in this list (and not-)	cidr-list
sipset	Source address is in this IP set (and not-)	filename
sport	Source port in this list	int-list
stime	Start time within this time window	time-range
stype	Source address matches classification	integer
tcp-flag	TCP flags are in the list	`FSRPAUEC`
tuple-del	Character separating the fields	character
tuple-dir	Specify IP-port mapping `forward` – as given `reverse` – flip source and destination `both` – do either matching	
tuple-field	Fields in tuple file for match	fieldlist
tuple-file	Record five-tuple matches line in file	filename

rwcut

Display network flow records as columnar or delimited text.

Syntax summary: (all parameters are optional)

```
rwcut formatting-parameters range-parameters
      output-parameters filename
```

Examples:

Quick overview of records in file:

```
rwcut --fields=1-6,stime flows.raw --pager=less
```

Output full records from file in csv format (sed command adds space after each comma):

```
rwcut --all-fields --delim=',' flows.raw | \
  sed -e 's/,/, /g' >flows.csv
```

Output data with integer IP addresses (rather than dotted-quad) for sorting, plotting, etc.:

```
rwcut --integer-ip --fields=sip,dip flows.raw \
  >int-sip-dip.txt
```

Changing order of columnar display:

```
rwcut --fields=protocol,sip,sport,dip,dport \
  flows.raw >flows.txt
```

Labeling source addresses using a pmap:

```
rwcut --pmap-file=mal:malware.pmap --pmap-col=10 \
  --fields=src-mal,1-7,stime\
  flows.raw >mal-flows.txt
```

Parameters:

Min-Name	Description	Arguments
	Output Parameters	
copy	Copy all input SiLK Flows to given pipe or file	filename
dry-run	Parse options and print column titles only	none
help	Print usage summary	none
output	Send output to given file path	filename
pager	Program to invoke to process output	filename
print-f	Print names of input files as they are opened	none
site-con	Specify location of the site configuration file	filename
version	Print this program's version	none

`rwcut` Parameters (continued)

Min-Name	Description	Arguments
	Range Parameters	
all-fields	Print all known fields to the output	none
end-rec	Specify ending record number	integer
fields	Specify fields to print	fieldlist
ipv6-policy	Specify how to handle v4/v6 `ignore` – drop IPv6 records `asv4` – convert v6 to v4 else ignore `mix` – allow both `force` – convert v6 to v4 `only` – drop IPv4	
num	Specify number of records to print	integer
pmap-file	Prefix map file to read	map:filename
start-rec	Specify starting record number	integer
xarg	Read input file names from file or pipe	filename (opt.)
	Formatting Parameters	
col	Specify separation character between columns	character
delim	Shortcut for no-columns no-final-del column-sep	character (opt.)
icmp	Print ICMP type & code in sPort and dPort fields	none
integer-ips	Print IP numbers as integers	none
integer-sen	Print sensor as an integer	none
integer-tcp	Print TCP flags as an integer	none
no-col	Disable fixed-width columnar output	none
no-final	Suppress column delimiter after last	none
no-titles	Do not print column headers	none
plugin	Load given plugin to add fields	filename
pmap-col	Maximum column width to use for pmap value output	integer
python-f	Use Python code to extend processing	filename
timestamp	Time format options `default` – yyyy/mm/ddThh:mm:ss.sss `iso` – yyyy-mm-dd hh:mm:ss.sss `m/d/y` – mm/dd/yyyy hh:mm:ss.sss `epoch` – seconds since UNIX epoch; ignores timezone `utc` – use UTC timezone `local` – use local timezone `no-msec` – truncate milliseconds	
zero-pad	Print IP numbers in zero-padded dotted-decimal	none

rwfileinfo

Print summary information about SiLK binary format files (flow, set, bag, etc.)

Syntax summary: (all non-file parameters are optional)

```
rwfileinfo parameters files
```

Examples:

Show all summary information on two files:

```
rwfileinfo flows.raw internal-ip.set
```

Show how generated and any comments:

```
rwfileinfo --fields=command-lines,annotations \
   flows.raw
```

Output info for loading into spreadsheet (without headings):

```
rwfileinfo --no-titles flows.raw
```

Parameters:

Min-Name	Description	Arguments
fields	List of fields to print	fieldlist
	1 – format(id); 2 – version; 3 – byte-order	
	4 – compression(id); 5 – header-length; 6 – record-length	
	7 – count-records; 8 – file-size; 9 – command-lines	
	10 – record-version; 11 – silk-version; 12 – packed-file-info	
	13 – probe-name; 14 – annotations; 15 – prefix-map	
	16 – IP set; 17 – bag	
help	Print usage summary	none
no-titles	Suppress file names and field names	none
site-con	Specify location of the site configuration file	filename
summary	Print total files; file sizes; records	none
version	Print this program's version	none

rwsiteinfo

Displays information about site collection configuration, including sensor names and numbers. Replaces `mapsid` command from prior versions of SiLK.

Syntax summary: (`fields` parameter is required)
`rwsiteinfo` *`parameters`* `--fields=`*`site-fields`*

Examples:

Print list of all sensor names and numbers:
```
rwsiteinfo --fields=sensor,id-sensor
```
Print sensor name for two sensor numbers:
```
rwsiteinfo --fields=sensor --sensor=0,1
```
Print description of a sensor:
```
rwsiteinfo --fields=describe-sensor --sensor=SEN0
```

Parameters:

Min-Name	Description	Arguments
classes	Display listed classes	`all`
col	Specify separation character between columns	character
data	Root of directory containing repository	filename
delim	Shortcut for no-columns no-final-del column-sep	character (opt.)
fields	List of fields to print	site-fields
flowtypes	Display listed class/type pairs	`all`/type
help	Print usage summary	none
list-delim	Use specified character in fields list	character
no-col	Disable fixed-width columnar output	none
no-final	Suppress column delimiter after last	none
no-titles	Do not print column headers	none
pager	Program to invoke to process output	filename
sensors	Display listed sensors	int-list or name list
site-con	Specify location of the site configuration file	filename
type	Display listed types	type list
version	Print this program's version	none

Site fields:

`class` – role of sensor as configured*	`flowtype` – class/type pair*
`default-class` – default sensor role*	`id-flowtype` – integer class/type pair*
`mark-defaults` – indicate use of defaults	`id-sensor` – integer sensor ID*
`default-type` – default flow category*	`sensor` – name of sensor*
`describe-sensor` – text description of sensor	`type` – flow category*

** These fields also have a :list form (e.g. class:list) that formats the entry as a comma-separated list instead of across multiple lines.*

rwset

Read binary flow records and generate one or more IP sets.

Syntax summary: (option parameters and source are optional)

```
rwset option-parameters field-parameters source
```

Examples:

Generate set from source IP addresses or records in file:

```
rwset --sip-file=src.set flows.raw
```

Generate sets with refiltering:

```
rwfilter --start=2011/04/15:00 --end=2011/04/15:07 \
  --type=out --proto=6 --pass=stdout | \
rwset --sip=tcp-src.set --copy=stdout | \
rwfilter --input-pipe=stdin --sport=0-1023 \
  --pass=stdout | \
rwset --sip=tcp-rsvd-src.set
```

Parameters:

Min-Name	Description	Arguments
	Option Parameters	
compress	Select compression	comp. opt.
copy	Copy all input SiLK Flows to given pipe or file	filename
help	Print usage summary	none
invocation-strip	Remove command history from file header	none
note-a	Put arg in file header	string
note-f	Put contents of arg in file header	filename
note-s	Remove note entries from file header	none
print-f	Print names of input files as they are opened	none
record-v	SiLK version compatibility	2 or 3
site-con	Specify location of configuration file	filename
version	Print this program's version	none
xarg	Read input file names from file or pipe	filename (opt.)
	Field Parameters (at least one needed)	
any-file	Store IP set of both source and desitination addresses	filename
dip-file	Store IP set of destination addresses	filename
nhip-file	Store set of flow markings	filename
sip-file	Store IP set of source addresses	filename

rwsetbuild

Read text list of IP addresses and produce binary IP set.

Syntax summary: (can use `stdin` for input and `stdout` for output, otherwise filenames)

```
rwsetbuild parameters input output
```

Examples:

Generate IP set from one-address-per-line file:

```
rwsetbuild list.txt list.set
```

Generate IP set from file with address ranges (colon-separated):

```
rwsetbuild --ip-range=':' ranges.txt ranges.set
```

Produce sorted list of unique IP addresses in file:

```
rwsetbuild input.txt stdout | rwsetcat
```

Parameters: (all optional)

Min-Name	Description	Arguments
compress	Select compression	comp. opt.
help	Print usage summary	none
invocation-strip	Remove command history from file header	none
ip-ranges	Allow input of address ranges in IP or integer format (no wildcards)	character (opt.)
note-a	Put arg in file header	string
note-f	Put contents of arg in file header	filename
record-v	SiLK version compatibility	2 or 3
version	Print this program's version	none

rwsettool

Perform operations on set files to produce new set files.

Syntax summary: (operation and arg-sets are required, parameters are optional)

```
rwsettool operation arg-sets parameters
```

where *arg-sets* is a blank-delimited list of IP set file names

Examples:

Merging two sets:

```
rwsettool --union day1.set day2.set \
  --output=either.set
```

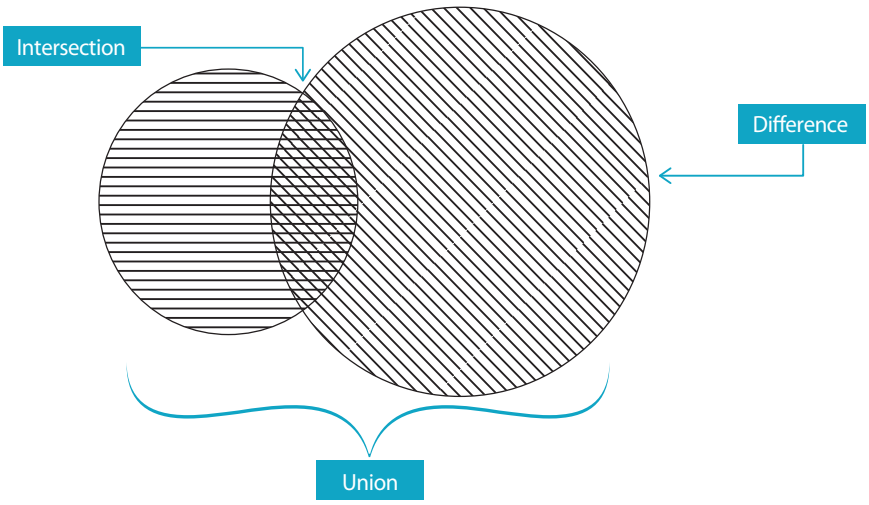
Finding common elements:

```
rwsettool --intersect day1.set day2.set \
  --output=both.set
```

Finding non-common elements:

```
rwsettool --diff day1.set day2.set --output=only1.set
rwsettool --diff day2.set day1.set | \
rwsettool --union stdin only1.set \
  --output=not-comm.set
```

Set operations:



`rwsettool` Parameters

Min-Name	Description	Arguments
	Operations	
diff	Gathers IPs from first input not in other input sets	none
intersect	Gathers IPs that exist in ALL the input sets	none
mask	Only one IP per block	integer
sample	Only random samples from input; need size or ratio	none
union	Gathers IPs that exist in ANY input set	none
	Option Parameters	
compress	Compression for output	comp. opt.
fill	Use complete blocks of given prefix length	integer
help	Print usage summary	none
invocation-strip	Remove command history from file header	none
note-a	Use arg as annotation	string
note-f	Use arg contents as annotation	filename
note-s	Do not copy notes from the input to the output	none
output	Specify output location	filename
ratio	The probability that an individual IP will be sampled	0.0-1.0
record-v	SiLK version compatibility	`2` or `3`
seed	Random number seed for `--sample`	decimal
size	The sample size for `--sample` per input	integer
version	Print this program's version	none

rwsetcat

Read binary IP set and produce text.

Syntax summary: (all optional)

```
rwsetcat  parameters  set-filenames
```

Examples:

List IP addresses in set into text file:

```
rwsetcat list.set >list.txt
```

Count IP addresses from standard input:

```
rwsetcat --count
```

Summarize CIDR/16 blocks (class B subnets) in set:

```
rwsetcat --net=B list.set
```

List IP addresses in set as integers (for plotting):

```
rwsetcat --integer-ip list.set
```

Parameters: (also formatting as rwcut takes – no plugin, pmap, or python)

Min-Name	Description	Arguments
cidr	Print IPs in CIDR block notation	none
count	Print the number of IPs	none
help	Print usage summary	none
ip-range	Print IPs as ranges of count|low|high|	none
net	Summarize CIDR blocks in set	blocks/sub
	blocks is comma-separated list of CIDR sizes or letters	
	T=0; A=8; B=16; C=24; X=27; H=32; S=count sub blocks	
	sub is comma-separated list of CIDR sizes or letters	
print-ips	Also print IPs if count or statistics parameter present	none
print-s	Print set statistics	none
version	Print this program's version	none

rwsort

Sort binary flow records, merging files if required.

Syntax summary: (parameters and flow-files are optional, `fields` is required)

```
rwsort --fields=key-fields parameters flow-files
```

Examples:

Ordering flow file by start time:

```
rwsort --fields=stime flows.raw >sorted.raw
```

Ordering flow file by source IP address, then by time:

```
rwsort --fields=sip,stime flows.raw \
  --output=src-time.raw
```

Merging two flow files and ordering by start time:

```
rwsort --fields=stime one.raw two.raw >time-order.raw
```

Parameters:

Min-Name	Description	Arguments
input-pipe	Get input byte stream from pipe	filename
help	Print usage summary	none
note-a	Store arg as an annotation	text
note-f	Store contents of arg as an annotation	filename
output	Output destination	filename
plugin	Load given plugin(s) to add fields	filename
pmap-f	Prefix map file to read. Use before `fields`	map:filename
presort	Merge only (do not sort)	none
print-f	Print names of input files	none
python	Use Python code to extend processing	filename
reverse	Reverse the sort order	none
site-con	Site configuration file	filename
sort-buff	Memory allocation for sort buffer	integer[k,M,G]
temp	Store temporary files here	dirname
version	Print this program's version	none
xarg	Read input file names from file or pipe	filename (opt.)

rwcount

Summarize binary flow records across time.

Syntax summary: (all parameters optional)

```
rwcount parameters flow-files
```

Examples:

Generate 30-second counts of records from standard input, with data proportional to time:

```
rwcount >30-sec.txt
```

Generate five-minute counts from file, with data proportional to time:

```
rwcount --bin-size=300 flows.raw >five-min.txt
```

Generate hourly counts in csv format, with data only in start time block, from file (including sed command to add space after comma):

```
rwcount --bin-size=3600 --delim=',' \
  --load-scheme=1  flows.raw  | \
  sed -e 's/,/, /g'  >hr.csv
```

Common bin-size values:

Interval	bin-size Value
5 min	300
10 min	600
15 min	900
30 min	1800
Hour	3600
Day	86400
Week	604800

`rwcount` Parameters

(Also formatting as rwcut takes – no plugin, pmap, or python)

Min-Name	Description	Arguments
bin-size	Size of bins in seconds (default 30.000)	decimal
bin-slots	Print bin labels using the internal bin index	none
copy	Copy all input SiLK Flows to given pipe or file	filename
end	Print bins until this time	time
epoch	Print bin labels using epoch time	none
help	Print usage summary	none
load	Specifies handling of flows that span bins	Integer
	0 – split volume EVENLY across the bins	
	1 – fill FIRST appropriate bin with complete volume	
	2 – fill LAST appropriate bin with complete volume	
	3 – fill CENTERMOST bin with complete volume	
	4 – split volume into bins proportional to time ACTIVE	
	5 – assign MAXIMUM possible volume for each bin	
	6 – assign MINIMUM possible volume for each bin	
	(for 5 and 6, sum of all bin values may not match total volume)	
output	Send output to given file path (default stdout)	filename
pager	Program to invoke to process output	filename
print-f	Print names of input files as they are opened	none
site-con	Specify location of the site configuration file	filename
skip-zero	Don't print bins that have no flows	none
start	Print bins from this time forward	time
version	Print this program's version	none
xarg	Read input file names from file or pipe	filename (opt.)

rwstats

Generate top N, bottom N, or descriptive statistics from file.

Syntax summary: (two alternative forms)

Generate descriptive statistics: (overall or by protocol, filename is optional)

```
rwstats --overall filename
rwstats --detail=protocol-list filename
```

Generate top/bottom N lists:

```
rwstats --fields=fieldlist --values=vallist --top bound
  filename options
rwstats --fields=fieldlist --values=vallist --bottom bound
  filename options
```

where *vallist* is a comma-separated list of bytes, packets, records, sip-distinct, or dip-distinct

Examples:

Find 10 highest-volume IP pairs:

```
rwstats --fields=sip,dip --values=bytes --top \
  --count=10 flows.raw
```

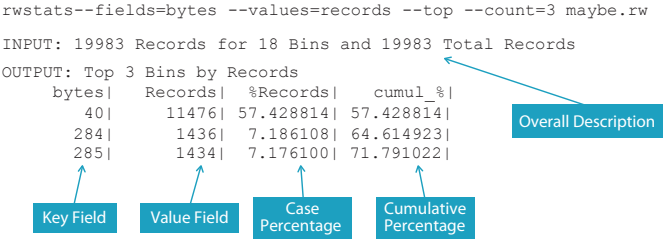
Find all destination ports that get more than ten percent of the traffic by frequency:

```
rwstats --fields=dport --values=records --top \
  --percent=10 flows.raw
```

Print descriptive statistics on traffic volumes:

```
rwstats --overall flows.raw
```

Output sample:



`rwstats` Parameters

(Also all formatting parameters from rwcut)

Min-Name	Description	Arguments
	Bounds	
count	Specify N	integer
percent	Specify percent for bound. Only for Bytes, Packets, or Flows	decimal
threshold	Specify value for bound (not from plugins)	integer
	Options	
bin-time	Specify bin size for time keys	integer
copy	Copy all input SiLK Flows to given pipe or file	filename
help	Print usage summary	none
ipv6-policy	Specify how to handle v4/v6	(see `rwcut` description)
legacy-h	Print help including legacy switches	none
no-per	Don't print the percentage columns	none
output	Send output to given file path	filename
pager	Program to invoke to process output	filename
presort	Assume input has been presorted for fields	none
print-f	Print names of input files	none
site-con	Specify location of the site configuration file	filename
temp	Store temporary files in this directory	dirname
version	Print this program's version	none
xarg	Read input file names from file or pipe	filename (opt.)

rwuniq

Summarize traffic volumes based on unique combinations of flow record fields.

Syntax summary: (options and filename are optional; values may be replaced by counting parameters)

```
rwuniq --fields=fieldlist --values=vallist  options file-
name
```

Examples:

Generate byte count totals of protocols grouped by hour from a file:

```
rwuniq --fields=proto,stime --bin-time=3600 \
  --values=bytes flows.raw
```

Generate flow and byte count totals of high-volume destination ports from a file:

```
rwuniq --fields=dport --values=bytes,dist:sip \
  maybe.rw
```

Generate contrasting views of traffic by size and by source port:

```
rwuniq --fields=bytes --values=records,distinct:dip \
  --output=bytes.txt --copy=stdout flows.raw | \
rwuniq --fields=sport --values=records, distinct:dip \
  --output=sport.txt
```

Count source ports per source address:

```
rwuniq --fields=sip --values=distinct:sport flows.rw
```

Output sample:

```
rwuniq --fields=dport --values=bytes,distinct:sip maybe.rw

dPort|               Bytes|sIP-Distin|
   22|             2492768|         1|
 7051|              636478|         1|
 7052|              635862|         1|
```

`rwuniq` Parameters

(Also all formatting parameters from rwcut)

Min-Name	Description	Arguments
	Option Parameters	
bin-time	Specify bin size for time keys	Integer
copy	Copy all input SiLK Flows to given pipe or file	filename
fields	Field combination for bins	fieldlist
help	Print usage summary	none
ipv6-policy	Specify how to handle v4/v6	(see `rwcut` description)
output	Send output to given file path	filename
pager	Program to invoke to process output	filename
presort	Assume input has been presorted with fields	none
print-f	Print names of input files as they are opened	none
site-con	Specify location of the site configuration file	filename
sort-out	Present the output in sorted order	none
temp	Store temporary files here	dirname
version	Print this program's version	none
	Counting Parameters	
all	Bytes, packets, flows, stime, and etime	none
bytes	Sum bytes in each bin	int-range
dip-dist	Count distinct dIPs in each bin	int-range
etime	Print latest time flow was seen in each bin	none
flows	Count flow records in each bin	int-range
packets	Sum packets in each bin	int-range
sip-dist	Count distinct sIPs in each bin	int-range
stime	Print earliest time flow was seen in each bin	none
values	Value(s) to compute: `bytes`, `packets`, `records`, `distinct:`*`KEYFIELD`*`, stime-latest, etime-earliest`	valuelist
xarg	Read input file names from file or pipe	filename (opt.)

Notes

IP Protocols

Num	Name	Description	Header bytes
0	HOPOPT	IPv6 Hop-by-Hop	28
1	ICMP	Internet Control Messages	24
2	IGMP	Internet Group Management	28
3	GGP	Gateway-to-Gateway	20
4	IPv4	v4 Encapsulation	40
6	TCP	Transmission Control	40
8	EGP	Exterior Gateway	34
9	IGP	Interior Gateway	20
17	UDP	User Datagram	28
27	RDP	Reliable Data	20
28	IRTP	Internet Reliable Transaction	24
41	IPv6	IPv6 Encapsulation	40
43	IPv6-Route	IPv6 Routing Header	36
44	IPv6-Frag	IPv6 Fragment Header	44
46	RSVP	Reservation Protocol	28
47	GRE	Generic Route Encapsulation	19
50	ESP	Encap Security Payload	28
51	AH	Authentication Header	32
53	SWIPE	IP with Encryption	28
58	ICMP	ICMP for IPv6	28
59	NoNxt	No Next Header for IPv6	--
60	IPv6-Opts	Destination Options for IPv6	28
88	EIGRP	Enhanced Interior Gateway Routing	20
98	ENCAP	Encapsulation Header	24
99		Private Encryption	20
132	SCTP	Stream Control Transmission	32
143-252		Unassigned	--
253-254		Experimental	--
255		Reserved	--

SiLK Commands (continued)

Text Output SiLK Tools

page	Tool summary
	rwbagcat – display and characterize bag content
	rwcompare – determine if two flow files are identical
18	rwcount – time-series counts
8	rwcut – text from flows
	rwfglob – list repository files from rwfilter selection parameters
10	rwfileinfo – describe file contents
	rwpcut – display packet fields of PCAP data
	rwpmapcat – display pmap content
	rwpmaplookup – display pmap label for IP addresses
	rwresolve – perform DNS lookup from IP address text
	rwscan – apply scan detection models to flows
16	rwsetcat – display IP set content
	rwsetmember – determine which IP sets have this address
11	rwsiteinfo – display repository information as configured
20	rwstats – generate top N/bottom N counts
22	rwuniq – generate aggregate counts

For More Information

http://tools.netsa.cert.org/silk/docs.html

Analysts' Handbook: Using SiLK for Network Traffic Analysis - tutorial on the SiLK tools and on using them for analyzing network traffic

PySiLK: SiLK in Python - reference guide for manipulating SiLK Flow data from within Python

The SiLK Reference Guide - every SiLK manual page in a single document

SiLK Installation Handbook - instructions on configuring, building, and installing SiLK at your site