Network Analysis with SiLK
Day 1

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Notices

This material is based upon work funded and supported by the Department of State under Contract No. FA8702-15-D-0002 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.

DM18-1124

Intended Audience

Attendees should have familiarity with network technology with emphasis on TCP/IP and the basic services in use on the Internet.

To get the most from the hands-on portion of the class you will need:

- A laptop computer
- Familiarity with the Linux command line
- Familiarity with a text editor in Linux (vi, emacs, nano…)

Overview:
Network Analysis with SiLK

Day 1

- Part 1
  - What is Network Flow?
  - Interpreting Flow Records
  - Issuing SiLK Commands
- Part 2
  - Analysis Tools and Categorization
  - Tuple Files
  - Pmaps

Day 2

- Part 3
  - IP Sets
  - Bags
  - Advanced Partitioning

Part I Lessons: Network Flow

1. What is Network Flow?
2. Interpreting Flow Records
3. Issuing SiLK Commands

What is network flow?

A log of all network activity; not a recording of all packets

A record of metadata from related packets

- similar to a phone bill (call detail record)

Content of messages is *not* recorded

- much, much more compact
  - longer retention
  - less processing
- increased privacy
- less impact from encryption

Call Detail Records

LATITUDE	LONGITUDE	DATE	TIME	NUMBER	NAME	DURATION
44.50880 N	73.18223 W	1/28/2008	0917	802-555-1234	Chittenden Bank	0:10:17
44.50880 N	73.18223 W	1/28/2008	0942	802-555-8673	Poopsie LaRue	0:01:03
44.50880 N	73.18223 W	1/28/2008	0945	802-555-9201	Hanley Strappman	0:05:32
44.27834 N	73.21263 W	1/29/2008	2205	802-555-7758	Verizon Voice Mail	0:01.13
44.27834 N	73.21263 W	1/29/2008	1532	802-555-4492	Widgets LLC	0:03:47
44.27834 N	73.21263 W	1/29/2008	2209	802-555-7758	Verizon Voice Mail	0:00.36
44.50880 N	73.18223 W	1/30/2008	0830	202-555-1818	British Embassy	0:18:12
44.27834 N	73.21263 W	1/30/2008	2208	802-555-7758	Verizon Voice Mail	0:00.53
44.27834 N	73.21263 W	1/30/2008	2211	802-555-8673	Poopsie LaRue	0:06:18
44.50880 N	73.18223 W	1/31/2008	0903	202-555-1843	British Embassy	0:03:21
44.50880 N	73.18223 W	1/31/2008	0908	416-555-9834	British Embassy	0:22:04
44.4143 N	73.03561 W	1/31/2008	1047	802-555-9201	Hanley Strappman	0:01:02
44.4143 N	73.03561 W	1/31/2008	1050	213-555-2761	M. Fendell	0:09:06
44.25295 N	72.58229 W	1/31/2008	1127	802-555-9201	Hanley Strappman	0:05:38

What SiLK does

Investigation analysis

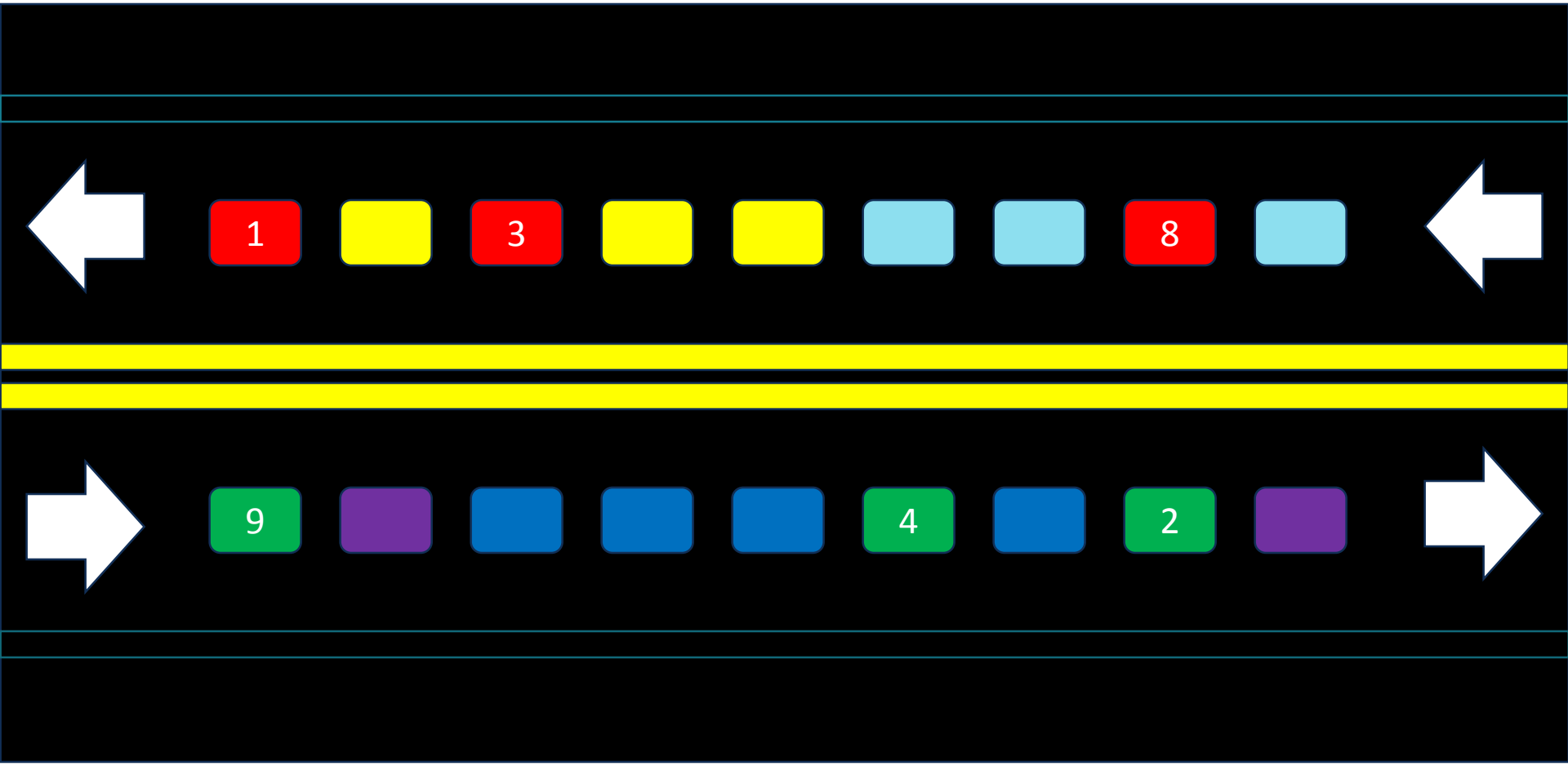
- most useful for analysing past network events
- may feed an automated report generator
- good for forensics (what happened ***before*** the incident?)

Descriptive analysis – profiling/categorizing

Directed analysis (hunt) – looking for specific malicious behavior

Exploratory analysis – looking for the unusual

Unidirectional Flows (Uniflows)



Terminology Description

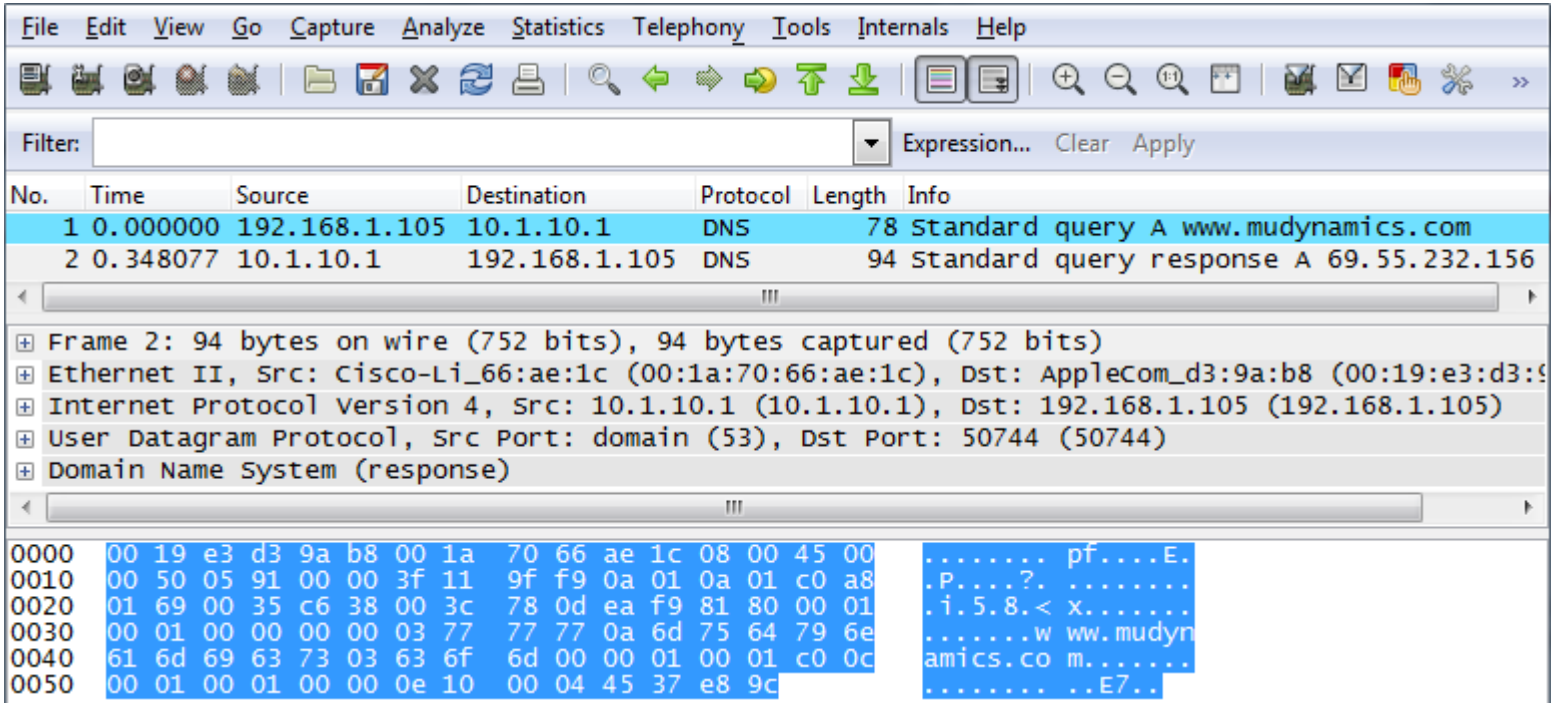
Term	Description
Network Flow	A generic term for the summarization of packets related to the same flow or connection into a single record
NetFlow	A term that Cisco coined to describe a set of format specifications for storing network flow information in a digital record. Most common are versions 5 and 9. Often used interchangeably with Network Flow. Also written as netflow.
IPFIX	A format specification from the IETF for flow records, an extension of Cisco NetFlow v9 See IETF 7011 https://tools.ietf.org/html/rfc7011
SiLK	The System for Internet Level Knowledge Another set of format specifications for flow records and other related data, plus the tool suite to process that data

What's in a Record?

Fields found to be useful in analysis:

- source address, destination address
- source port, destination port (Internet Control Message Protocol [ICMP] type/code)
- IP [transport] protocol
- bytes, packets in flow
- accumulated TCP flags (all packets, first packet)
- start time, duration (milliseconds)
- end time (derived)
- sensor identity
- flow termination conditions
- application-layer protocol

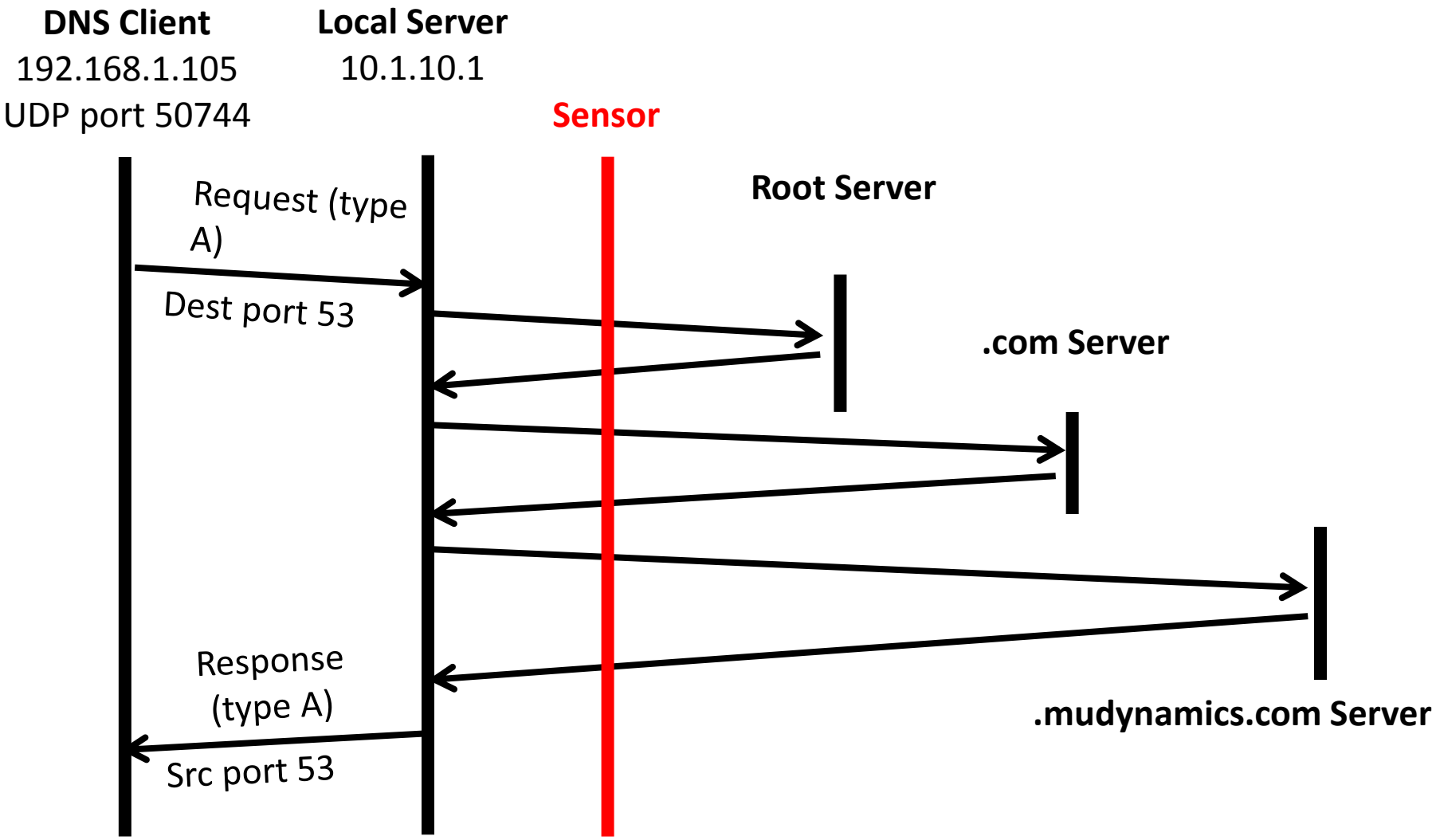
DNS packets viewed in Wireshark



SiLK tool (**`rwcut`**) output

```
          sIP|          dIP|sPort|dPort|pro|packets|bytes|sensor|type|
192.168.1.105|    10.1.10.1|50744|   53| 17|      1|   64|    S1| out|
    10.1.10.1|192.168.1.105|   53|50744| 17|      1|   80|    S1|  in|
```

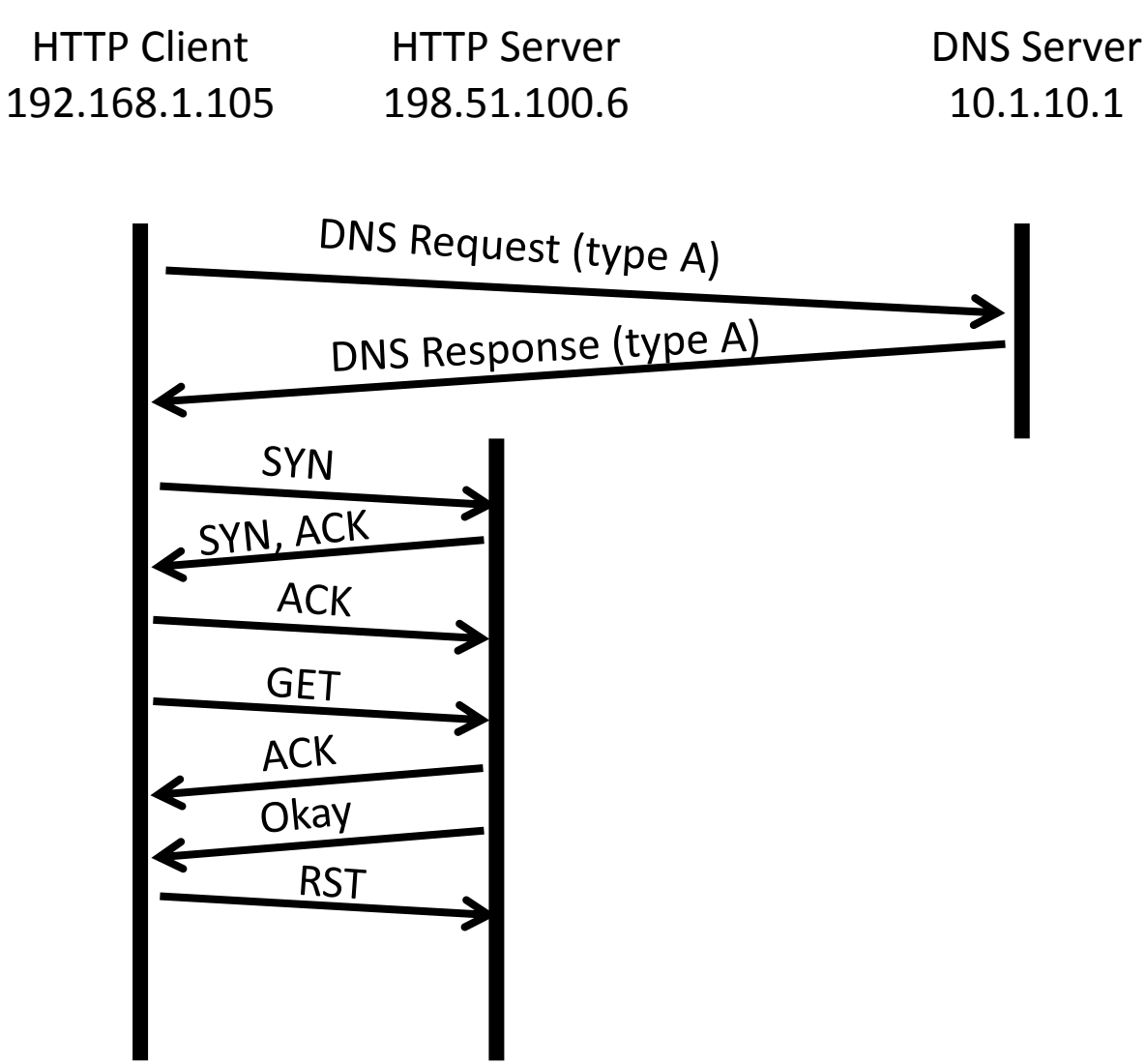
Realistic Sequence Diagram



What is this? -1

```
          sIP|          dIP|sPort|dPort|pro|packets|flags|initF|  type|
192.168.1.105|    10.1.10.1|50744|   53| 17|      1|     |     |   out|
    10.1.10.1|192.168.1.105|   53|50744| 17|      1|     |     |    in|
192.168.1.105| 198.51.100.6|49152|   80|  6|      4| SRPA| S   |outweb|
 198.51.100.6|192.168.1.105|   80|49152|  6|      3| S PA| S  A| inweb|
```

HTTP Sequence Diagram



What is this? -2

```
          sIP|           dIP|sPort|dPort|pro|packets|  bytes|flags|
30.22.105.250| 71.55.40.253|52415|   25|  6|     22| 14045|FSRPA|
 71.55.40.253|30.22.105.250|   25|52415|  6|     19|  1283|FS PA|
30.22.105.250| 71.55.40.253|52415|   25|  6|      1|    40|  R  |
```

What is this? -3

sIP	dIP	pro	packets	bytes	sTime
99.217.139.155	177.252.24.89	1	2	122	2010/12/08T00:04:30.172
99.217.139.155	177.252.149.249	1	2	122	2010/12/08T00:04:37.302
99.217.139.155	177.252.24.52	1	2	122	2010/12/08T00:04:37.312
99.217.139.155	177.252.24.127	1	2	122	2010/12/08T00:04:58.363
99.217.139.155	177.252.24.196	1	2	122	2010/12/08T00:05:04.327
99.217.139.155	177.252.149.30	1	2	122	2010/12/08T00:05:09.242
99.217.139.155	177.252.149.173	1	2	122	2010/12/08T00:05:12.174
99.217.139.155	177.252.24.13	1	2	122	2010/12/08T00:05:14.114
99.217.139.155	177.252.24.56	1	2	122	2010/12/08T00:05:15.383
99.217.139.155	177.252.24.114	1	2	122	2010/12/08T00:05:18.228
99.217.139.155	177.252.202.92	1	2	122	2010/12/08T00:05:22.466
99.217.139.155	177.252.202.68	1	2	122	2010/12/08T00:05:23.497
99.217.139.155	177.252.24.161	1	2	122	2010/12/08T00:05:30.256
99.217.139.155	177.252.202.238	1	2	122	2010/12/08T00:05:33.088

What is this? -4

```
         sIP|          dIP|sPort|dPort|pkts| bytes|flags|              sTime|
88.187.13.78|71.55.40.204|40936|   80|  83|  3512|FS PA|2010/12/08T11:00:01|
71.55.40.204|88.187.13.78|   80|40936|  84|104630|FS PA|2010/12/08T11:00:01|
88.187.13.78|71.55.40.204|40938|   80| 120|  4973|FS PA|2010/12/08T11:00:04|
71.55.40.204|88.187.13.78|   80|40938| 123|155795|FS PA|2010/12/08T11:00:05|
88.187.13.78|71.55.40.204|56172|   80|  84|  3553|FS PA|2010/12/08T12:00:02|
71.55.40.204|88.187.13.78|   80|56172|  83|103309|FS PA|2010/12/08T12:00:02|
88.187.13.78|71.55.40.204|56177|   80| 123|  5093|FS PA|2010/12/08T12:00:05|
71.55.40.204|88.187.13.78|   80|56177| 124|157116|FS PA|2010/12/08T12:00:05|
```

Lesson I.1 Summary

Flow records constitute a log of network activity.

Flow analysis can answer many questions without storing content.

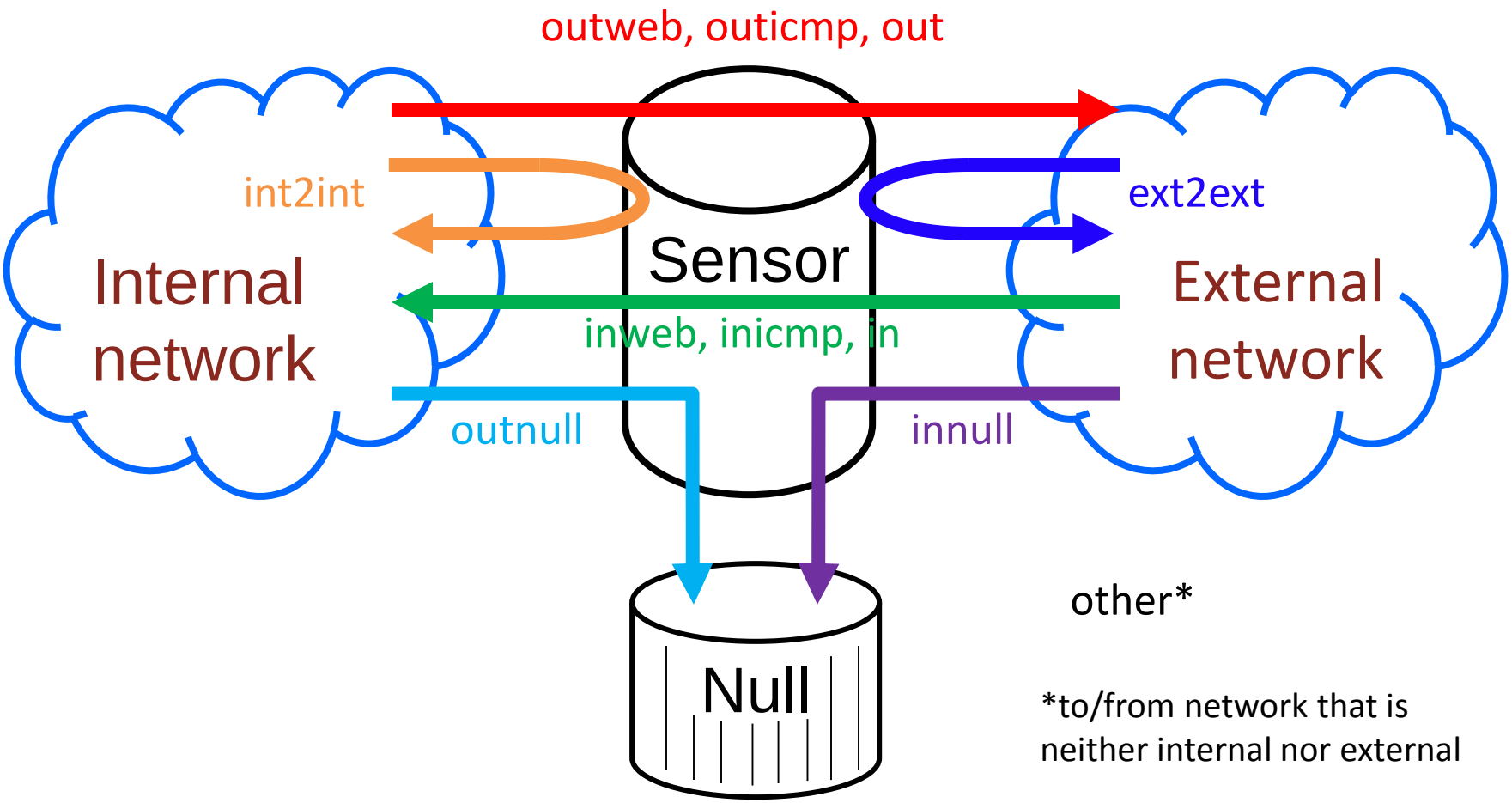
Flow records are extremely compact. Benefits are

- long retention
- faster processing
- reduced privacy concerns
- encryption is not an obstacle

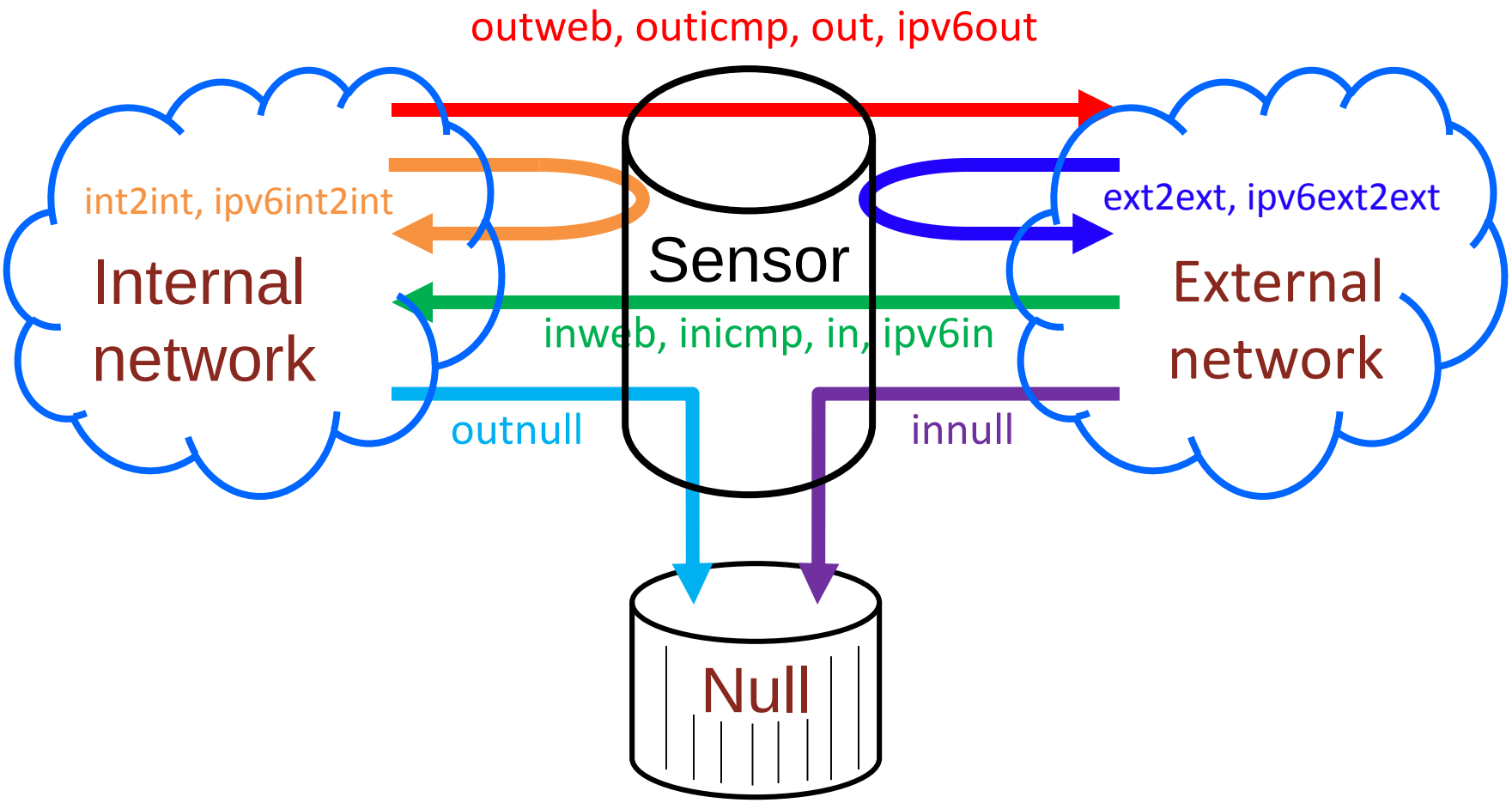
SiLK uses unidirectional flows—uniflows.

Lesson I.2
Interpreting Network Flows

Standard SiLK Types

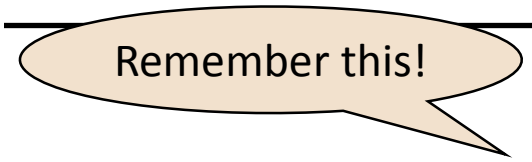


More complete SiLK Types



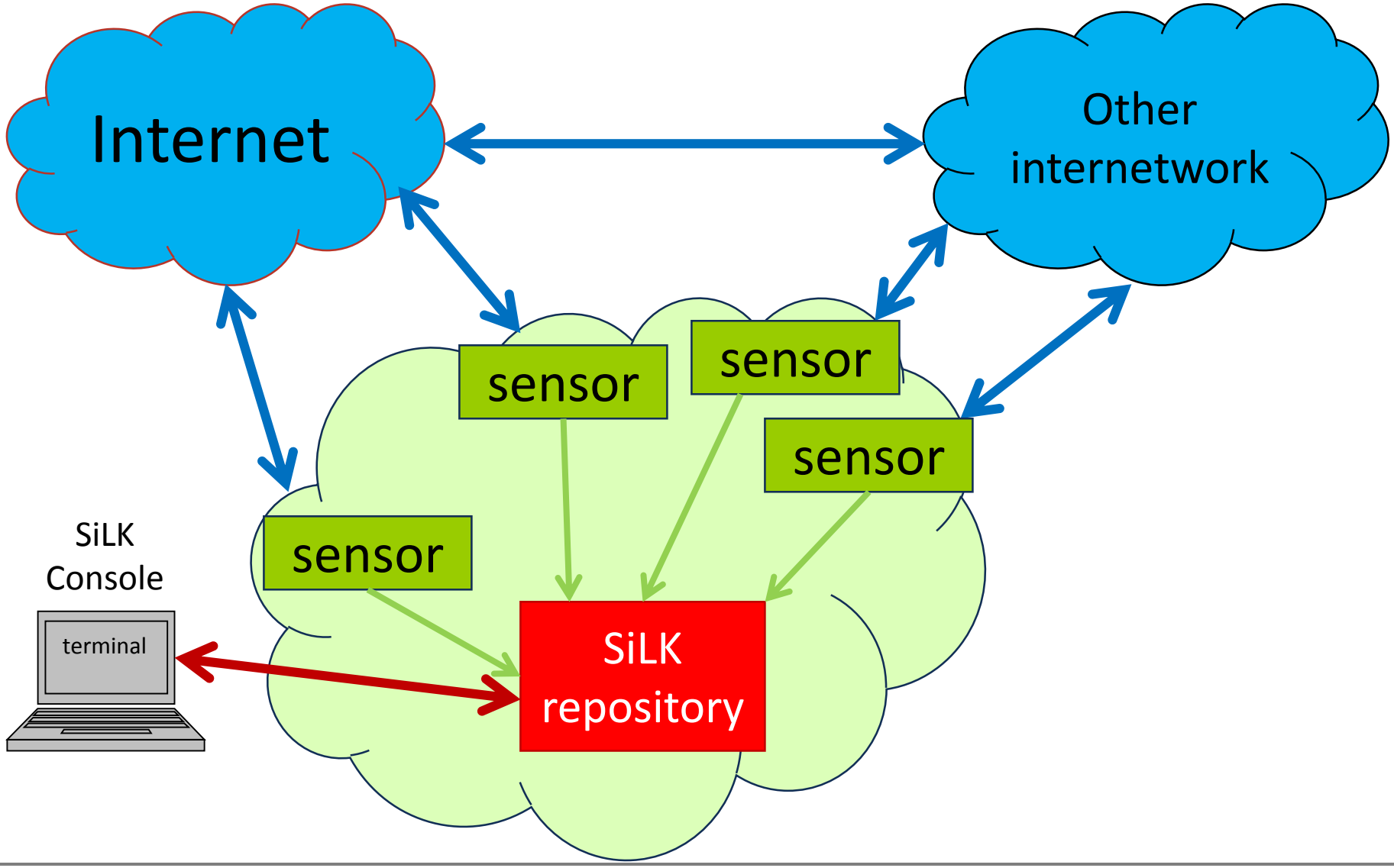
Common Types in SiLK

Type	Description
inweb, outweb	Inbound/outbound TCP ports 80, 443, 8080
innull, outnull	Inbound/outbound filtered traffic
inicmp, outicmp	Inbound/outbound IP protocol 1
in, out	Inbound/outbound not in above categories
int2int, ext2ext	Internal to internal, external to external
other	Source not internal or external, or destination not internal, external, or null

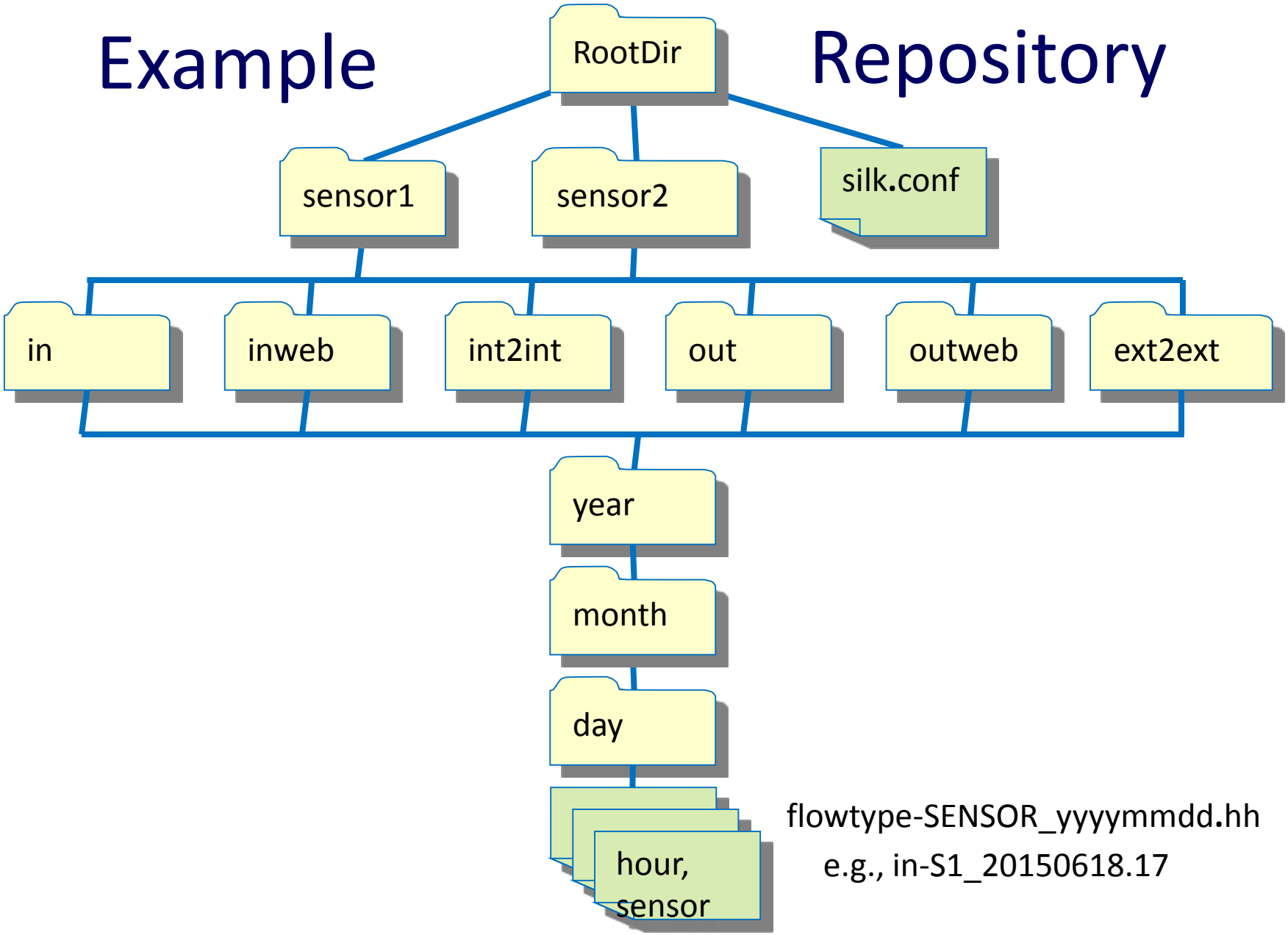


Names in **bold** are often default types, confirm with: `rwsiteinfo --fields=default-type`

Network Monitoring



Example Repository



flowtype-SENSOR_yyyymmdd.hh

e.g., in-S1_20150618.17

What does the repository contain?

The SiLK repository is a collection of files containing the netflow records in SiLK binary format.

There is a configuration file that describes the repository.

The best way to learn about the local repository is with the SiLK **`rwsiteinfo`** command.

By now you have probably figured out that most SiLK commands start with **`rw`**.

Synopsis for the rwsiteinfo command

```
rwsiteinfo --fields=FIELD[,FIELD...]
        { [--classes=CLASS[,CLASS...]] [--
types=TYPE[,TYPE...]]
          | [--flowtypes=CLASS/TYPE[,CLASS/TYPE...]] }
        [--sensors=SENSOR[,SENSOR...]]
        [--data-rootdir=ROOT_DIRECTORY] [--site-config-
file=FILENAME]
        [--timestamp-format=FORMAT] [--no-titles]
        [--no-columns] [--column-separator=CHAR]
        [--no-final-delimiter] [{--delimited | --
delimited=CHAR}]
        [--list-delimiter=CHAR] [--output-path=PATH]
        [--pager=PAGER_PROG]

  rwsiteinfo --help
  rwsiteinfo --help-fields
  rwsiteinfo --version
```

Using the **`rwsiteinfo`** command

As you just saw, the **`rwsiteinfo`** command has several options and parameters.

You cannot just run it by itself, you have to tell it what you want.

Now let's find out what sensors are used to collect netflow records.

- For this we will use the **`--fields`** option

Run the **`rwsiteinfo`** command to find out about sensors

```
$ rwsiteinfo --fields=id-sensor,sensor,describe-sensor
Sensor-ID|Sensor|Sensor-Description|
        0|    S0|           Div0Ext|
        1|    S1|           Div1Ext|
        2|    S2|           Div0Int|
        3|    S3|          Div1Int1|
        4|    S4|          Div1Int2|
        5|    S5|          Div1log1|
        6|    S6|          Div1log2|
        7|    S7|          Div1log3|
        8|    S8|          Div1log4|
        9|    S9|          Div1ops1|
       10|   S10|          Div1ops2|
       11|   S11|          Div1ops3|
       12|   S12|           Div1svc|
       13|   S13|           Div1dhq|
       14|   S14|           Div1dmz|
       15|   S15|           Div1mar|
       16|   S16|           Div1med|
       17|   S17|          Div1nusr|
       18|   S18|           Div1mgt|
       19|   S19|        Div1intel1|
       20|   S20|        Div1intel2|
       21|   S21|        Div1intel3|
$
```

Exercise 1
Use **`rwsiteinfo`** to learn about types

Modify the **`rwsiteinfo`** command you just saw to display the SiLK types in use in the local repository.

- Use the man page to find out what to use with the **`--fields=`** option.

  `- man rwsiteinfo`

- Then run the command.

Lesson I.2 Summary

Sensor placement affects what is seen or not seen in flow records.

We learned to interpret network activity from flow records.

A class of SiLK sensors uses a particular set of record types.

You know how to find out more about the repository.

Lesson 1.3
Issuing SiLK Commands

Collection, Packing, and Analysis

Collection of flow data

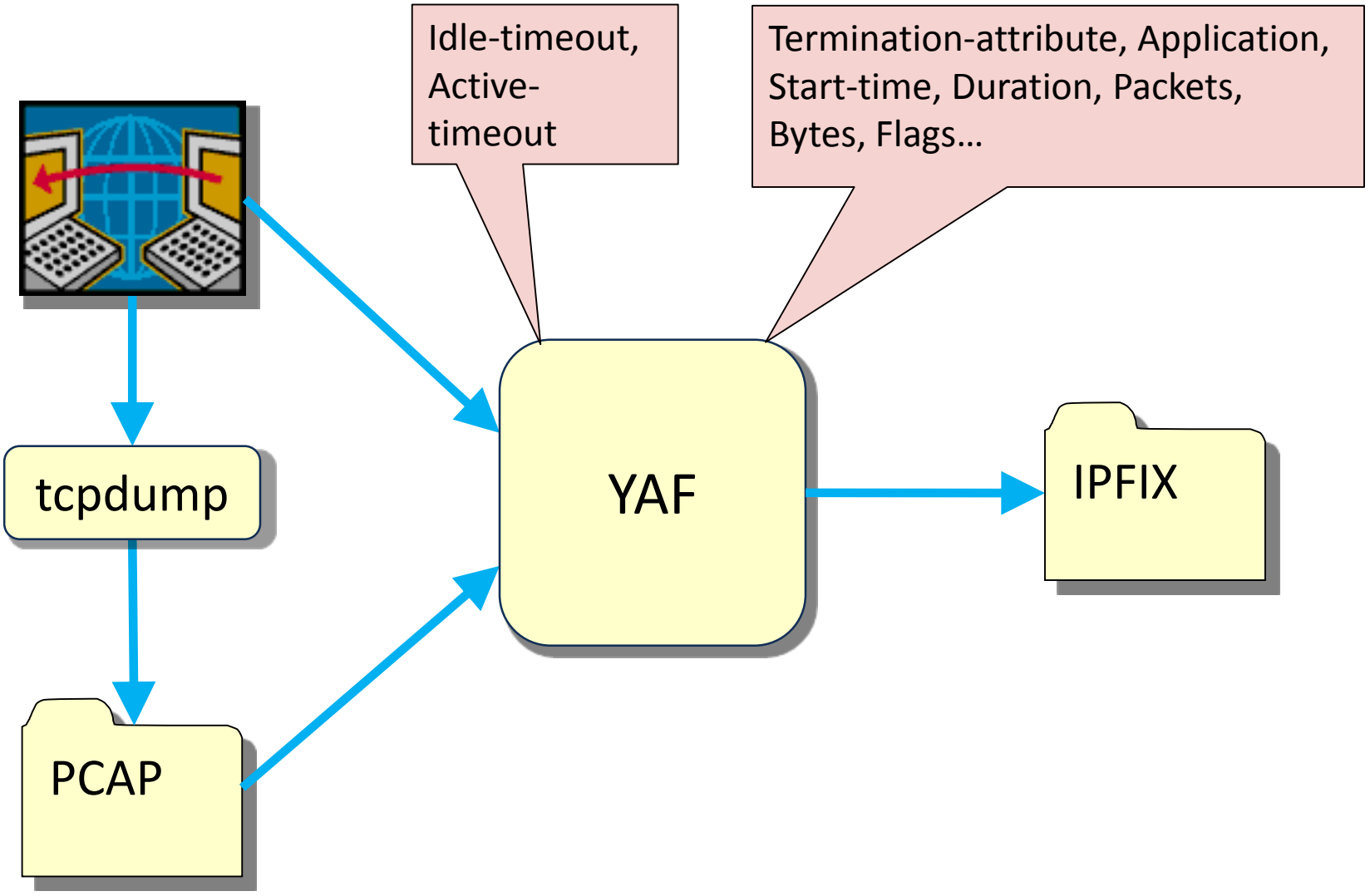
- Examines packets and summarizes into standard flow records
- Timeout and payload-size values are established during collection

Packing stores flow records in a scheme optimized for space and ease of analysis.

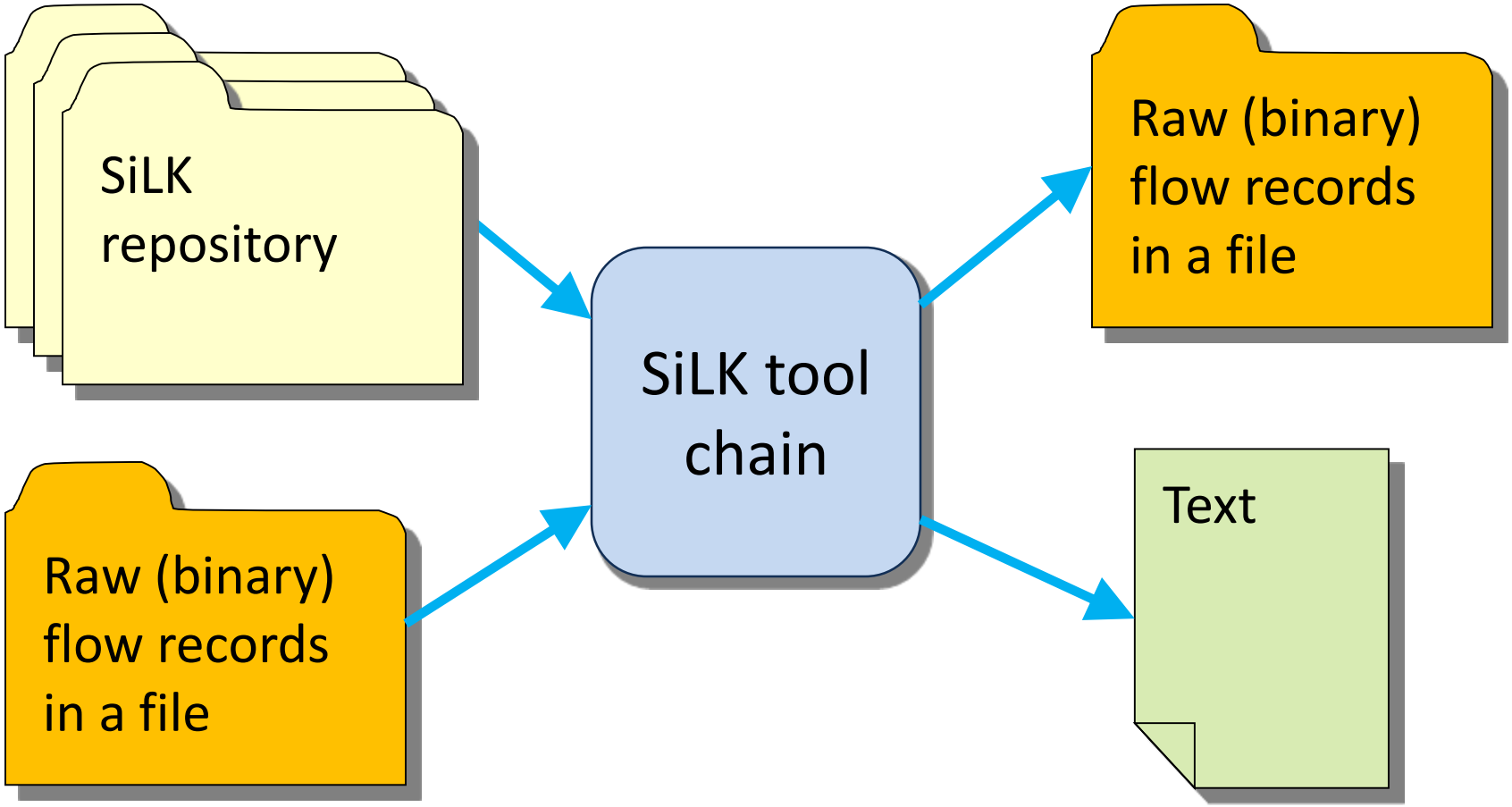
Analysis of flow data

- Investigation of flow records using SiLK tools

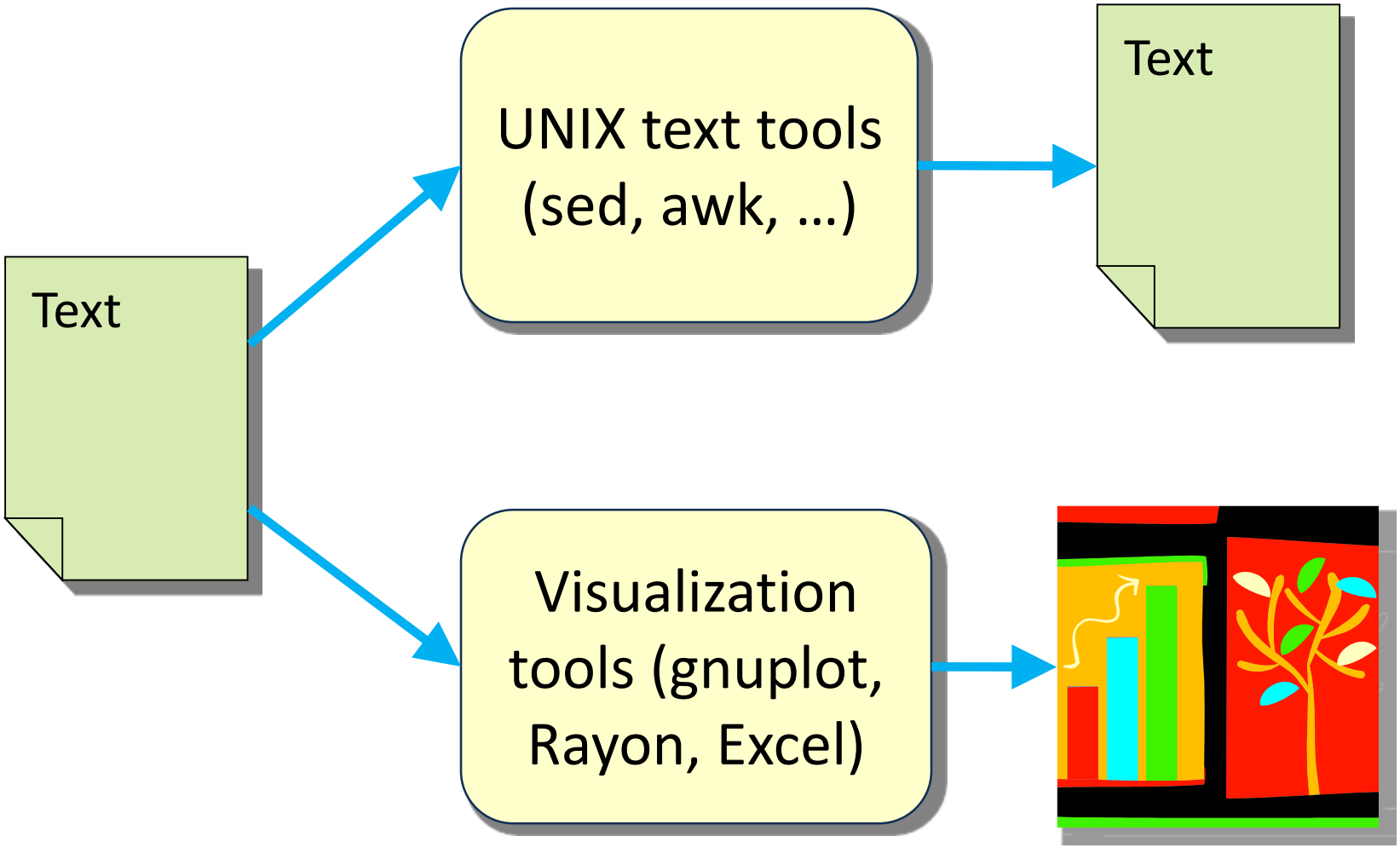
Collection



Analysis



Reporting



Rayon is a trademark of Carnegie Mellon University
Excel is a registered trademark of Microsoft Corporation

Exercise 2: Which sensors are defined?

```
rwsiteinfo --fields=id-sensor,sensor

rwsiteinfo --fields=id-sensor,sensor | less

rwsiteinfo --fields=id-sensor,sensor,describe-sensor \
 | less
```

http://www.rainbird.com/landscape/products/central/flowsensors.htm

Which record-types are defined?

```
$ rwsiteinfo --fields=type,mark-defaults
   Type|Defaults|
     in|      +*|
    out|      + |
  inweb|      +*|
 outweb|      + |
 innull|      + |
outnull|      + |
int2int|      + |
ext2ext|      + |
 inicmp|      +*|
outicmp|      + |
  other|      + |
$
```

Where can I get more information?

We can't discuss all parameters for every tool.

Resources

- Analyst's Handbook
- SiLK Reference Guide (collected man pages)
- **`--help`** option
- **`man`** command
- **`http://tools.netsa.cert.org`**

Additional resources at the end

http://retirementincomejournal.com/issue/august-24-2011/article/a-reference-book-built-for-you

Where can I get more information? <silk command> --help

http://winefolly.com/review/types-dessert-wine/

```
$ rwsiteinfo --help
rwsiteinfo --fields=<FIELDS> [SWITCHES]
        Print selected information about the classes, types, flowtypes
        and sensors defined in the SiLK site configuration file.  By
        default, the selected information is printed for every class,
        type, and sensor defined in the file; to restrict the output,
        specify one or more of --classes, --types, --flowtypes, or
        --sensors.

SWITCHES:
--help No Arg. Print this usage output and exit. Def. No
--version No Arg. Print this program's version and exit. Def. No
--help-fields No Arg. Describe each field and exit. Def. no
--fields Req Arg. Print the fields named in this comma-separated list. Choices:
        (Semicolon separates unique items. Comma separates item aliases.
         Names are case-insensitive and may be abbreviated to the shortest
         unique prefix.)
        class; type; flowtype; id-flowtype; sensor; id-sensor;
        describe-sensor; default-class; default-type; mark-defaults;
        class:list; type:list; flowtype:list; id-flowtype:list; sensor:list;
        id-sensor:list; default-class:list; default-type:list;
        repo-start-date; repo-end-date; repo-file-count;
…
```

Where can I get more information?
Man page

```
rwsiteinfo(1)                    SiLK Tool Suite                    rwsiteinfo(1)

NAME
     rwsiteinfo - Print information from the silk.conf site configuration file

SYNOPSIS
          rwsiteinfo --fields=FIELD[,FIELD...]
               { [--classes=CLASS[,CLASS...]] [--types=TYPE[,TYPE...]]
                 | [--flowtypes=CLASS/TYPE[,CLASS/TYPE...]] }
               [--sensors=SENSOR[,SENSOR...]]
               [--data-rootdir=ROOT_DIRECTORY] [--site-config-file=FILENAME]
               [--timestamp-format=FORMAT] [--no-titles]
               [--no-columns] [--column-separator=CHAR]
               [--no-final-delimiter] [{--delimited | --delimited=CHAR}]
               [--list-delimiter=CHAR] [--output-path=PATH]
               [--pager=PAGER_PROG]
         rwsiteinfo --help
         rwsiteinfo --help-fields
         rwsiteinfo --version

DESCRIPTION
       rwsiteinfo is a utility to print selected information about the classes, types,
       flowtypes, and sensors that are defined in the silk.conf(5) site configuration file.
       The --fields switch is required, and its argument is a comma-separated list of field


...
```

http://winefolly.com/review/types-dessert-wine/

Lesson I.3 Summary

We learned the parts of Linux commands.

Data should be kept in binary form as long as possible.

We learned where to get more information about commands.

We learned to obtain information about the SiLK repository using the rwsiteinfo command.

Part II: Basic SiLK Tools

Part II Lessons: Network Flow

1. Analysis Tools and Categorization
2. Tuple Files
3. Pmaps

Basic SiLK Tools: `rwcut`

http://www.ifrick.ch/2011/09/apple-senkt-preise-in-der-schweiz-teilweise-massiv/price-cut/

But I can't read binary...

`rwcut` provides a way to display binary records as human-readable ASCII:

- useful for printing flows to the screen
- useful for input to text-processing tools
- Usually you'll only need the **`--fields`** option.

sip	packets	type	flags
dip	bytes	in	initialflags
sport	sensor	out	sessionflags
dport	*scc*	dur	application
protocol	*dcc*	stime	attributes
class	nhip	*etime*	*itype & icode*

Field names in italics are *derived* fields

`rwcut` Default Display

By default

- sIP (1), sPort (3)
- dIP (2), dPort (4)
- Protocol (5)
- packets, bytes
- flags
- sTime, eTime, duration
- sensor

```
--all-fields  # way too much info
--fields=1-5,sTime  # just right
```

Create the ex3records.rw file with the ex2.sh script

```
# Single Bash script command with pipes

rm ex3records.rw

rwfilter --type=all                \
         --sensor=S0               \
         --start=2015/06/18T17     \
         --proto=0-                \
         --pass=stdout             \
| rwsort --fields=stime            \
| rwfilter --input-pipe=stdin      \
         --max-pass=30             \
         --proto=0-                \
         --pass=ex3records.rw
```

Exercise 2
Create Binary File for Next Exercises

In the VM confirm that the script is there by typing

`ls –l ex02.sh`

(ell ess hyphen ell)

Run the script **`ex02.sh`**

do this by typing:

`./ex02.sh`

Confirm the results

`ls –l ex3records.rw`

Exercise 3: What do the data look like?

```
rwcut ex3records.rw --fields=1-5,packets
```

Try other values for **`--fields`**.

Try omitting the **`--fields`** option.

Exercise 3: What do the data look like?

```
rwcut --fields=1-5,packets ex3records.rw
```

```
            sIP|            dIP|sPort|dPort|pro|   packets|
     10.0.40.27|    155.6.2.254|    0| 2048|  1|         5|
    155.6.2.254|      155.6.2.1|    0|    0|  1|         5|
 192.168.200.10|     10.0.20.59|54671|   80|  6|         4|
     10.0.20.59| 192.168.200.10|   80|54671|  6|         4|
     10.0.20.58|     67.215.0.5|29020|   53| 17|         2|
     10.0.20.58|     67.215.0.5|18727|   53| 17|         2|
     67.215.0.5|      155.6.5.1|   53| 2737| 17|         1|
     10.0.20.58|    128.63.2.53|12367|   53| 17|         2|
     10.0.40.20|  192.168.40.20|60958|   53| 17|         2|
     10.0.20.58|   193.0.14.129|55147|   53| 17|         2|
     10.0.20.58|    128.8.10.90|63201|   53| 17|         2|
     10.0.20.58|    128.8.10.90|50785|   53| 17|         2|
     10.0.20.58|  192.58.128.30| 7119|   53| 17|         2|
     10.0.20.58| 192.228.79.201|35031|   53| 17|         2|
     10.0.40.20|  192.168.40.20|60418|   53| 17|         2|
     10.0.40.20| 192.168.161.69|  137|  137| 17|         3|
     10.0.20.58|     67.215.0.5|48803|   53| 17|         2|
     67.215.0.5|      155.6.5.1|   53| 5856| 17|         1|
     10.0.20.58|    128.63.2.53|19107|   53| 17|         2|
     10.0.20.58|    128.63.2.53|21024|   53| 17|         2|
     10.0.20.58|   202.12.27.33|64137|   53| 17|         2|
     10.0.40.20|  192.168.20.58|49326|   53| 17|         1|
     10.0.40.20|  192.168.40.20|65535|   53| 17|         2|
     67.215.0.5|      155.6.5.1|   53| 2741| 17|         1|
     10.0.20.58|    128.63.2.53|55584|   53| 17|         2|
     10.0.20.58|    128.63.2.53| 9554|   53| 17|         2|
     10.0.20.58|    128.63.2.53|51147|   53| 17|         2|
     10.0.20.58|    128.63.2.53|47997|   53| 17|         2|
     10.0.20.58|    128.63.2.53|35387|   53| 17|         2|
     10.0.20.58|  192.58.128.30|27414|   53| 17|         2|
```

Exercise 3: What do the data look like?

```
rwcut ex3records.rw
```

```
          sIP|           dIP|sPort|dPort|pro|   packets|     bytes|   flags|                 sTime| duration|                 eTime|sen|
    10.0.40.27|    155.6.2.254|    0| 2048|  1|         5|       420|        |2015/06/18T17:00:00.071|    4.026|2015/06/18T17:00:04.097| S0|
   155.6.2.254|      155.6.2.1|    0|    0|  1|         5|       420|        |2015/06/18T17:00:00.073|    4.024|2015/06/18T17:00:04.097| S0|
192.168.200.10|     10.0.20.59|54671|   80|  6|         4|       216|FS  A   |2015/06/18T17:00:00.211|    0.002|2015/06/18T17:00:00.213| S0|
    10.0.20.59| 192.168.200.10|   80|54671|  6|         4|       200| SR A   |2015/06/18T17:00:00.211|    0.004|2015/06/18T17:00:00.215| S0|
    10.0.20.58|     67.215.0.5|29020|   53| 17|         2|       168|        |2015/06/18T17:00:00.223|    0.000|2015/06/18T17:00:00.223| S0|
    10.0.20.58|     67.215.0.5|18727|   53| 17|         2|       170|        |2015/06/18T17:00:00.223|    0.000|2015/06/18T17:00:00.223| S0|
    67.215.0.5|      155.6.5.1|   53| 2737| 17|         1|        85|        |2015/06/18T17:00:00.277|    0.000|2015/06/18T17:00:00.277| S0|
    10.0.20.58|    128.63.2.53|12367|   53| 17|         2|       170|        |2015/06/18T17:00:00.281|    0.000|2015/06/18T17:00:00.281| S0|
    10.0.40.20|  192.168.40.20|60958|   53| 17|         2|       146|        |2015/06/18T17:00:00.287|    4.010|2015/06/18T17:00:04.297| S0|
    10.0.20.58|   193.0.14.129|55147|   53| 17|         2|       168|        |2015/06/18T17:00:00.293|    0.000|2015/06/18T17:00:00.293| S0|
    10.0.20.58|    128.8.10.90|63201|   53| 17|         2|       138|        |2015/06/18T17:00:00.305|    0.000|2015/06/18T17:00:00.305| S0|
    10.0.20.58|    128.8.10.90|50785|   53| 17|         2|       148|        |2015/06/18T17:00:00.305|    0.000|2015/06/18T17:00:00.305| S0|
    10.0.20.58|  192.58.128.30| 7119|   53| 17|         2|       148|        |2015/06/18T17:00:00.317|    0.000|2015/06/18T17:00:00.317| S0|
    10.0.20.58| 192.228.79.201|35031|   53| 17|         2|       148|        |2015/06/18T17:00:00.331|    0.000|2015/06/18T17:00:00.331| S0|
    10.0.40.20|  192.168.40.20|60418|   53| 17|         2|       148|        |2015/06/18T17:00:01.082|    4.008|2015/06/18T17:00:05.090| S0|
    10.0.40.20| 192.168.161.69|  137|  137| 17|         3|       234|        |2015/06/18T17:00:01.211|    3.022|2015/06/18T17:00:04.233| S0|
    10.0.20.58|     67.215.0.5|48803|   53| 17|         2|       170|        |2015/06/18T17:00:01.234|    0.000|2015/06/18T17:00:01.234| S0|
    67.215.0.5|      155.6.5.1|   53| 5856| 17|         1|        85|        |2015/06/18T17:00:01.276|    0.000|2015/06/18T17:00:01.276| S0|
    10.0.20.58|    128.63.2.53|19107|   53| 17|         2|       170|        |2015/06/18T17:00:01.278|    0.000|2015/06/18T17:00:01.278| S0|
    10.0.20.58|    128.63.2.53|21024|   53| 17|         2|       166|        |2015/06/18T17:00:01.302|    0.000|2015/06/18T17:00:01.302| S0|
    10.0.20.58|   202.12.27.33|64137|   53| 17|         2|       146|        |2015/06/18T17:00:01.314|    0.000|2015/06/18T17:00:01.314| S0|
    10.0.40.20|  192.168.20.58|49326|   53| 17|         1|        74|        |2015/06/18T17:00:01.508|    0.000|2015/06/18T17:00:01.508| S0|
    10.0.40.20|  192.168.40.20|65535|   53| 17|         2|       148|        |2015/06/18T17:00:01.862|    4.008|2015/06/18T17:00:05.870| S0|
    67.215.0.5|      155.6.5.1|   53| 2741| 17|         1|        84|        |2015/06/18T17:00:02.275|    0.000|2015/06/18T17:00:02.275| S0|
    10.0.20.58|    128.63.2.53|55584|   53| 17|         2|       168|        |2015/06/18T17:00:02.279|    0.000|2015/06/18T17:00:02.279| S0|
    10.0.20.58|    128.63.2.53| 9554|   53| 17|         2|       168|        |2015/06/18T17:00:02.289|    0.002|2015/06/18T17:00:02.291| S0|
    10.0.20.58|    128.63.2.53|51147|   53| 17|         2|       160|        |2015/06/18T17:00:02.289|    0.000|2015/06/18T17:00:02.289| S0|
    10.0.20.58|    128.63.2.53|47997|   53| 17|         2|       168|        |2015/06/18T17:00:02.291|    0.000|2015/06/18T17:00:02.291| S0|
    10.0.20.58|    128.63.2.53|35387|   53| 17|         2|       170|        |2015/06/18T17:00:02.297|    0.000|2015/06/18T17:00:02.297| S0|
    10.0.20.58|  192.58.128.30|27414|   53| 17|         2|       148|        |2015/06/18T17:00:02.315|    0.000|2015/06/18T17:00:02.315| S0|
```

Exercise 3: What do the data look like?

```
rwcut --fields=1-5,packets,flags,stime ex3records.rw
```

```
            sIP|            dIP|sPort|dPort|pro|   packets|   flags|                  sTime|
    10.0.40.27|    155.6.2.254|    0| 2048|  1|         5|        |2015/06/18T17:00:00.071|
   155.6.2.254|      155.6.2.1|    0|    0|  1|         5|        |2015/06/18T17:00:00.073|
192.168.200.10|     10.0.20.59|54671|   80|  6|         4|FS  A   |2015/06/18T17:00:00.211|
    10.0.20.59| 192.168.200.10|   80|54671|  6|         4| SR A   |2015/06/18T17:00:00.211|
    10.0.20.58|     67.215.0.5|29020|   53| 17|         2|        |2015/06/18T17:00:00.223|
    10.0.20.58|     67.215.0.5|18727|   53| 17|         2|        |2015/06/18T17:00:00.223|
    67.215.0.5|      155.6.5.1|   53| 2737| 17|         1|        |2015/06/18T17:00:00.277|
    10.0.20.58|    128.63.2.53|12367|   53| 17|         2|        |2015/06/18T17:00:00.281|
    10.0.40.20|  192.168.40.20|60958|   53| 17|         2|        |2015/06/18T17:00:00.287|
    10.0.20.58|   193.0.14.129|55147|   53| 17|         2|        |2015/06/18T17:00:00.293|
    10.0.20.58|    128.8.10.90|63201|   53| 17|         2|        |2015/06/18T17:00:00.305|
    10.0.20.58|    128.8.10.90|50785|   53| 17|         2|        |2015/06/18T17:00:00.305|
    10.0.20.58|  192.58.128.30| 7119|   53| 17|         2|        |2015/06/18T17:00:00.317|
    10.0.20.58| 192.228.79.201|35031|   53| 17|         2|        |2015/06/18T17:00:00.331|
    10.0.40.20|  192.168.40.20|60418|   53| 17|         2|        |2015/06/18T17:00:01.082|
    10.0.40.20| 192.168.161.69|  137|  137| 17|         3|        |2015/06/18T17:00:01.211|
    10.0.20.58|     67.215.0.5|48803|   53| 17|         2|        |2015/06/18T17:00:01.234|
    67.215.0.5|      155.6.5.1|   53| 5856| 17|         1|        |2015/06/18T17:00:01.276|
    10.0.20.58|    128.63.2.53|19107|   53| 17|         2|        |2015/06/18T17:00:01.278|
    10.0.20.58|    128.63.2.53|21024|   53| 17|         2|        |2015/06/18T17:00:01.302|
    10.0.20.58|   202.12.27.33|64137|   53| 17|         2|        |2015/06/18T17:00:01.314|
    10.0.40.20|  192.168.20.58|49326|   53| 17|         1|        |2015/06/18T17:00:01.508|
    10.0.40.20|  192.168.40.20|65535|   53| 17|         2|        |2015/06/18T17:00:01.862|
    67.215.0.5|      155.6.5.1|   53| 2741| 17|         1|        |2015/06/18T17:00:02.275|
    10.0.20.58|    128.63.2.53|55584|   53| 17|         2|        |2015/06/18T17:00:02.279|
    10.0.20.58|    128.63.2.53| 9554|   53| 17|         2|        |2015/06/18T17:00:02.289|
    10.0.20.58|    128.63.2.53|51147|   53| 17|         2|        |2015/06/18T17:00:02.289|
    10.0.20.58|    128.63.2.53|47997|   53| 17|         2|        |2015/06/18T17:00:02.291|
    10.0.20.58|    128.63.2.53|35387|   53| 17|         2|        |2015/06/18T17:00:02.297|
    10.0.20.58|  192.58.128.30|27414|   53| 17|         2|        |2015/06/18T17:00:02.315|
```

Exercise 3: What do the data look like?

```
rwcut --fields=1-5,iType,iCode,packets,flags,stime ex3records.rw
```

```
           sIP|            dIP|sPort|dPort|pro|iTy|iCo|   packets|   flags|                  sTime|
    10.0.40.27|    155.6.2.254|    0| 2048|  1|  8|  0|         5|        |2015/06/18T17:00:00.071|
   155.6.2.254|      155.6.2.1|    0|    0|  1|  0|  0|         5|        |2015/06/18T17:00:00.073|
192.168.200.10|     10.0.20.59|54671|   80|  6|   |   |         4|FS  A   |2015/06/18T17:00:00.211|
    10.0.20.59| 192.168.200.10|   80|54671|  6|   |   |         4| SR A   |2015/06/18T17:00:00.211|
    10.0.20.58|     67.215.0.5|29020|   53| 17|   |   |         2|        |2015/06/18T17:00:00.223|
    10.0.20.58|     67.215.0.5|18727|   53| 17|   |   |         2|        |2015/06/18T17:00:00.223|
    67.215.0.5|      155.6.5.1|   53| 2737| 17|   |   |         1|        |2015/06/18T17:00:00.277|
    10.0.20.58|    128.63.2.53|12367|   53| 17|   |   |         2|        |2015/06/18T17:00:00.281|
    10.0.40.20|  192.168.40.20|60958|   53| 17|   |   |         2|        |2015/06/18T17:00:00.287|
    10.0.20.58|   193.0.14.129|55147|   53| 17|   |   |         2|        |2015/06/18T17:00:00.293|
    10.0.20.58|    128.8.10.90|63201|   53| 17|   |   |         2|        |2015/06/18T17:00:00.305|
    10.0.20.58|    128.8.10.90|50785|   53| 17|   |   |         2|        |2015/06/18T17:00:00.305|
    10.0.20.58|  192.58.128.30| 7119|   53| 17|   |   |         2|        |2015/06/18T17:00:00.317|
    10.0.20.58| 192.228.79.201|35031|   53| 17|   |   |         2|        |2015/06/18T17:00:00.331|
    10.0.40.20|  192.168.40.20|60418|   53| 17|   |   |         2|        |2015/06/18T17:00:01.082|
    10.0.40.20| 192.168.161.69|  137|  137| 17|   |   |         3|        |2015/06/18T17:00:01.211|
    10.0.20.58|     67.215.0.5|48803|   53| 17|   |   |         2|        |2015/06/18T17:00:01.234|
    67.215.0.5|      155.6.5.1|   53| 5856| 17|   |   |         1|        |2015/06/18T17:00:01.276|
    10.0.20.58|    128.63.2.53|19107|   53| 17|   |   |         2|        |2015/06/18T17:00:01.278|
    10.0.20.58|    128.63.2.53|21024|   53| 17|   |   |         2|        |2015/06/18T17:00:01.302|
    10.0.20.58|   202.12.27.33|64137|   53| 17|   |   |         2|        |2015/06/18T17:00:01.314|
    10.0.40.20|  192.168.20.58|49326|   53| 17|   |   |         1|        |2015/06/18T17:00:01.508|
    10.0.40.20|  192.168.40.20|65535|   53| 17|   |   |         2|        |2015/06/18T17:00:01.862|
    67.215.0.5|      155.6.5.1|   53| 2741| 17|   |   |         1|        |2015/06/18T17:00:02.275|
    10.0.20.58|    128.63.2.53|55584|   53| 17|   |   |         2|        |2015/06/18T17:00:02.279|
    10.0.20.58|    128.63.2.53| 9554|   53| 17|   |   |         2|        |2015/06/18T17:00:02.289|
    10.0.20.58|    128.63.2.53|51147|   53| 17|   |   |         2|        |2015/06/18T17:00:02.289|
    10.0.20.58|    128.63.2.53|47997|   53| 17|   |   |         2|        |2015/06/18T17:00:02.291|
    10.0.20.58|    128.63.2.53|35387|   53| 17|   |   |         2|        |2015/06/18T17:00:02.297|
    10.0.20.58|  192.58.128.30|27414|   53| 17|   |   |         2|        |2015/06/18T17:00:02.315|
```

Exercise 3: What do the data look like?

```
rwcut --fields=1-5,packets,flags,initialFlags,sessionFlags,stime ex3records.rw
```

```
          sIP|            dIP|sPort|dPort|pro|   packets|   flags|initialF|sessionF|                  sTime|
  10.1.60.203|   10.1.60.187|50398|   80|  6|         5|FS PA   | S      |F  PA   |2009/04/20T11:35:19.439|
  10.1.60.187|   10.1.60.203|   80|50398|  6|         5|FS PA   | S  A   |F  PA   |2009/04/20T11:35:19.440|
  10.1.60.203|    10.1.60.73|50189| 5222|  6|         6|FS PA   | S      |F  PA   |2009/04/20T11:35:19.446|
   10.1.60.73|   10.1.60.203| 5222|50189|  6|         5|FS PA   | S  A   |F  PA   |2009/04/20T11:35:19.447|
  10.1.60.203|   10.1.60.187|49592|  443|  6|        10|FS PA   | S      |F  PA   |2009/04/20T11:35:19.463|
  10.1.60.187|   10.1.60.203|  443|49592|  6|        10|FS PA   | S  A   |F  PA   |2009/04/20T11:35:19.464|
  10.1.60.203|   10.1.60.187|    0| 2048|  1|       276|        |        |        |2009/04/20T11:35:19.490|
  10.1.60.187|   10.1.60.203|    0|    0|  1|       276|        |        |        |2009/04/20T11:35:19.491|
  10.1.60.203|     10.1.60.5|    0| 2048|  1|       279|        |        |        |2009/04/20T11:35:19.494|
    10.1.60.5|   10.1.60.203|    0|    0|  1|       279|        |        |        |2009/04/20T11:35:19.495|
  10.1.60.203|     10.1.60.5|56515|   53| 17|         1|        |        |        |2009/04/20T11:35:19.500|
    10.1.60.5|   10.1.60.203|   53|56515| 17|         1|        |        |        |2009/04/20T11:35:19.502|
  10.1.60.203|    10.1.60.73|    0| 2048|  1|       276|        |        |        |2009/04/20T11:35:19.514|
   10.1.60.73|   10.1.60.203|    0|    0|  1|       276|        |        |        |2009/04/20T11:35:19.516|
  10.1.60.203|    10.1.60.25|    0| 2048|  1|       356|        |        |        |2009/04/20T11:35:19.528|
   10.1.60.25|   10.1.60.203|    0|    0|  1|       356|        |        |        |2009/04/20T11:35:19.529|
  10.1.60.203|    10.1.60.25|60515|   25|  6|         4| S      | S      | S      |2009/04/20T11:35:19.529|
    10.1.10.5|     10.1.60.5| 1031|   53| 17|        32|        |        |        |2009/04/20T11:35:23.415|
    10.1.60.5|     10.1.10.5|   53| 1031| 17|        32|        |        |        |2009/04/20T11:35:23.417|
    10.1.10.5|     10.1.60.5| 3507|   53|  6|         1| S      | S      |        |2009/04/20T11:35:23.443|
    10.1.60.5|     10.1.10.5|   53| 3507|  6|         1|  R A   |  R A   |        |2009/04/20T11:35:23.444|
    10.1.10.5|     10.1.60.5| 3508|   53|  6|         1| S      | S      |        |2009/04/20T11:35:23.445|
    10.1.60.5|     10.1.10.5|   53| 3508|  6|         1|  R A   |  R A   |        |2009/04/20T11:35:23.446|
    10.1.10.5|     10.1.60.5| 3507|   53|  6|         1| S      | S      |        |2009/04/20T11:35:24.084|
    10.1.10.5|     10.1.60.5| 3508|   53|  6|         1| S      | S      |        |2009/04/20T11:35:24.085|
    10.1.60.5|     10.1.10.5|   53| 3507|  6|         1|  R A   |  R A   |        |2009/04/20T11:35:24.085|
    10.1.60.5|     10.1.10.5|   53| 3508|  6|         1|  R A   |  R A   |        |2009/04/20T11:35:24.086|
    10.1.10.5|     10.1.60.5| 3507|   53|  6|         1| S      | S      |        |2009/04/20T11:35:24.632|
    10.1.10.5|     10.1.60.5| 3508|   53|  6|         1| S      | S      |        |2009/04/20T11:35:24.632|
    10.1.60.5|     10.1.10.5|   53| 3508|  6|         1|  R A   |  R A   |        |2009/04/20T11:35:24.633|
```

Exercise 3a: Make a Small File to Examine

```
# remove file; ok if it is not there yet
rm -f ex3arecords.rw
# get one record from the repository
rwfilter --type=in \
  --start-date=2015/6/17:00 --protocol=0- \
  --compress=none  \
  --max-pass=1     \
  --pass=ex3arecords.rw
# look at directory listing
ls -l ex3arecords.rw
```

Exercise 3a: What does a raw file look like?

```
# Basic information about the file
rwfileinfo ex3arecords.rw
# look at some of the fields in the record
rwcut --fields=1-5,packets ex3arecords.rw
# look at all of the fields in the record
rwcut --all-fields ex3arecords.rw
# look at the record at the "byte level"
# Is there any readable text?
hexdump -C ex3arecords.rw
```

Exercise 3a Output -1

```
$
$ ls -l ex3arecords.rw
-rw-r--r--. 1 pnk pnk 264 Sep 25 16:41 ex3arecords.rw

$ rwcut --fields=1-5,packets ex3arecords.rw
            sIP|            dIP|sPort|dPort|pro|   packets|
192.168.111.109|     10.0.40.53|50211| 5723|  6|        12|


$
$ rwcut --all-fields ex3arecords.rw
            sIP|            dIP|sPort|dPort|pro|   packets|     bytes|   flags|
sTime| duration|                  eTime|sen|   in|  out|
nhIP|initialF|sessionF|attribut|appli|cla|   type|             sTime+msec|
eTime+msec| dur+msec|iTy|iCo|
192.168.111.109|     10.0.40.53|50211| 5723|  6|        12|      4236|FS PA
|2015/06/17T00:00:00.148|    0.008|2015/06/17T00:00:00.156| S0|    0|    0|        0.0.0.0|
|        |        |    0|all|     in|2015/06/17T00:00:00.148|2015/06/17T00:00:00.156|
0.008|   |   |

$
```

Exercise 3a Output -2

```
rwcut --fields=1-5,packets ex3arecords.rw
            sIP|            dIP|sPort|dPort|pro|   packets|
 192.168.200.10|     10.0.40.20|55517|   53| 17|         1|


rwcut --all-fields ex3arecords.rw
            sIP|            dIP|sPort|dPort|pro|   packets|     bytes|   flags|                  sTime|
duration|                 eTime|sen|   in|  out|           nhIP|initialF|sessionF|attribut|appli|cla|
type|             sTime+msec|             eTime+msec| dur+msec|iTy|iCo|
 192.168.200.10|     10.0.40.20|55517|   53| 17|         1|        62|        |2015/06/18T17:00:08.971|
0.000|2015/06/18T17:00:08.971| S0|    0|    0|        0.0.0.0|        |        |        |    0|all|
  in|2015/06/18T17:00:08.971|2015/06/18T17:00:08.971|    0.000|   |   |
```

Exercise 3a Output -3A
Can you find the binary record?

```
hexdump -C ex3arecords.rw
00000000  de ad be ef 00 0c 10 00  00 2e 05 41 00 58 00 01  |...........A.X..|
00000010  00 00 00 02 00 00 00 74  72 77 66 69 6c 74 65 72  |.......trwfilter|
00000020  20 2d 2d 74 79 70 65 3d  69 6e 20 2d 2d 73 74 61  | --type=in --sta|
00000030  72 74 2d 64 61 74 65 3d  32 30 31 35 2f 30 36 2f  |rt-date=2015/06/|
00000040  31 37 20 2d 2d 70 72 6f  74 6f 63 6f 6c 3d 30 2d  |17 --protocol=0-|
00000050  20 2d 2d 63 6f 6d 70 72  65 73 73 3d 6e 6f 6e 65  | --compress=none|
00000060  20 2d 2d 6d 61 78 2d 70  61 73 73 3d 31 20 2d 2d  | --max-pass=1 --|
00000070  70 61 73 73 3d 65 78 33  61 72 65 63 6f 72 64 73  |pass=ex3arecords|
00000080  2e 72 77 00 00 00 00 00  00 00 00 2c 00 00 00 00  |.rw........,....|
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000000b0  94 b4 d0 fe 4d 01 00 00  08 00 00 00 23 c4 5b 16  |....M.......#.[.|
000000c0  06 00 00 00 1b 00 00 00  00 00 00 00 00 00 00 00  |................|
000000d0  0c 00 00 00 8c 10 00 00  00 00 00 00 00 00 00 00  |................|
000000e0  00 00 ff ff c0 a8 6f 6d  00 00 00 00 00 00 00 00  |......om........|
000000f0  00 00 ff ff 0a 00 28 35  00 00 00 00 00 00 00 00  |......(5........|
00000100  00 00 ff ff 00 00 00 00                           |........|
00000108
```

Exercise 3a Output -3B
Can you find it?

Decimal	Hex
192.168.200.010	c0 a8 c8 0a
010.000.040.020	0a 00 28 14
17	11
00053	0035
05517	d8dd

```
readable text?
                                          00 2e 05 41 00 58 00 01  |...........A.X..|
                                          72 77 66 69 6c 74 65 72  |.......|rwfilter|
                                          3d 32 30 31 35 2f 30 36  | --start=2015/06|
                                          2d 74 79 70 65 3d 61 6c  |/18T17 --type=al|
                                          6f 72 3d 53 30 20 2d 2d  |l --sensor=S0 --|
                                          20 2d 2d 63 6f 6d 70 72  |proto=0- --compr|
                                          20 2d 2d 6d 61 78 2d 70  |ess=none --max-p|
                                          70 61 73 73 3d 65 78 33  |ass=1 --pass=ex3|
00000080  61 72 65 63 6f 72 64 73  2e 72 77 00 00 00 00 00  |arecords.rw.....|
00000090  00 00 00 24 00 00 00 00  00 00 00 00 00 00 00 00  |...$............|
000000a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000b0  8b 09 9d 07 4e 01 00 00  00 00 00 00 dd d8 35 00  |....N.........5.|
000000c0  11 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000d0  01 00 00 00 3e 00 00 00  00 00 00 00 00 00 00 00  |....>...........|
000000e0  00 00 ff ff c0 a8 c8 0a  00 00 00 00 00 00 00 00  |................|
000000f0  00 00 ff ff 0a 00 28 14  00 00 00 00 00 00 00 00  |......(.........|
00000100  00 00 ff ff 00 00 00 00                           |........|
00000108
```

Exercise 3a Output -3C
Can you find it?

	Decimal	Hex
192.168.200.010		c0 a8 c8 0a
010.000.040.020		0a 00 28 14
	17	11
	00053	0035
	05517	d8dd

```
readable text?
  00 2e 05 41 00 58 00 01  |..........A.X..|
  72 77 66 69 6c 74 65 72  |.......|rwfilter|
  3d 32 30 31 35 2f 30 36  | --start=2015/06|
  2d 74 79 70 65 3d 61 6c  |/18T17 --type=al|
  6f 72 3d 53 30 20 2d 2d  |l --sensor=S0 --|
  20 2d 2d 63 6f 6d 70 72  |proto=0- --compr|
  20 2d 2d 6d 61 78 2d 70  |ess=none --max-p|
  70 61 73 73 3d 65 78 33  |ass=1 --pass=ex3|
  2e 72 77 00 00 00 00 00  |arecords.rw.....|
  00 00 00 00 00 00 00 00  |...$............|
  00 00 00 00 00 00 00 00  |................|
  00 00 00 00 dd d8 35 00  |....N.........5.|
  00 00 00 00 00 00 00 00  |................|
  00 00 00 00 00 00 00 00  |....>...........|
  00 00 00 00 00 00 00 00  |................|
  00 00 00 00 00 00 00 00  |......(.........|
                           |........|
00000080  61 72 65 63 6f 72 64 73
00000090  00 00 00 24 00 00 00 00
000000a0  00 00 00 00 00 00 00 00
000000b0  8b 09 9d 07 4e 01 00 00
000000c0  11 00 00 00 00 00 00 00
000000d0  01 00 00 00 3e 00 00 00
000000e0  00 00 ff ff c0 a8 c8 0a
000000f0  00 00 ff ff 0a 00 28 14
00000100  00 00 ff ff 00 00 00 00
00000108
```

Basic SiLK Tools: `rwfilter`

Swiss Army knife logo is a registered trademark of Victorinox AG

`rwfilter` Syntax

General form

```
rwfilter {INPUT | SELECTION}
  PARTITION OUTPUT [OTHER]
```

Example call

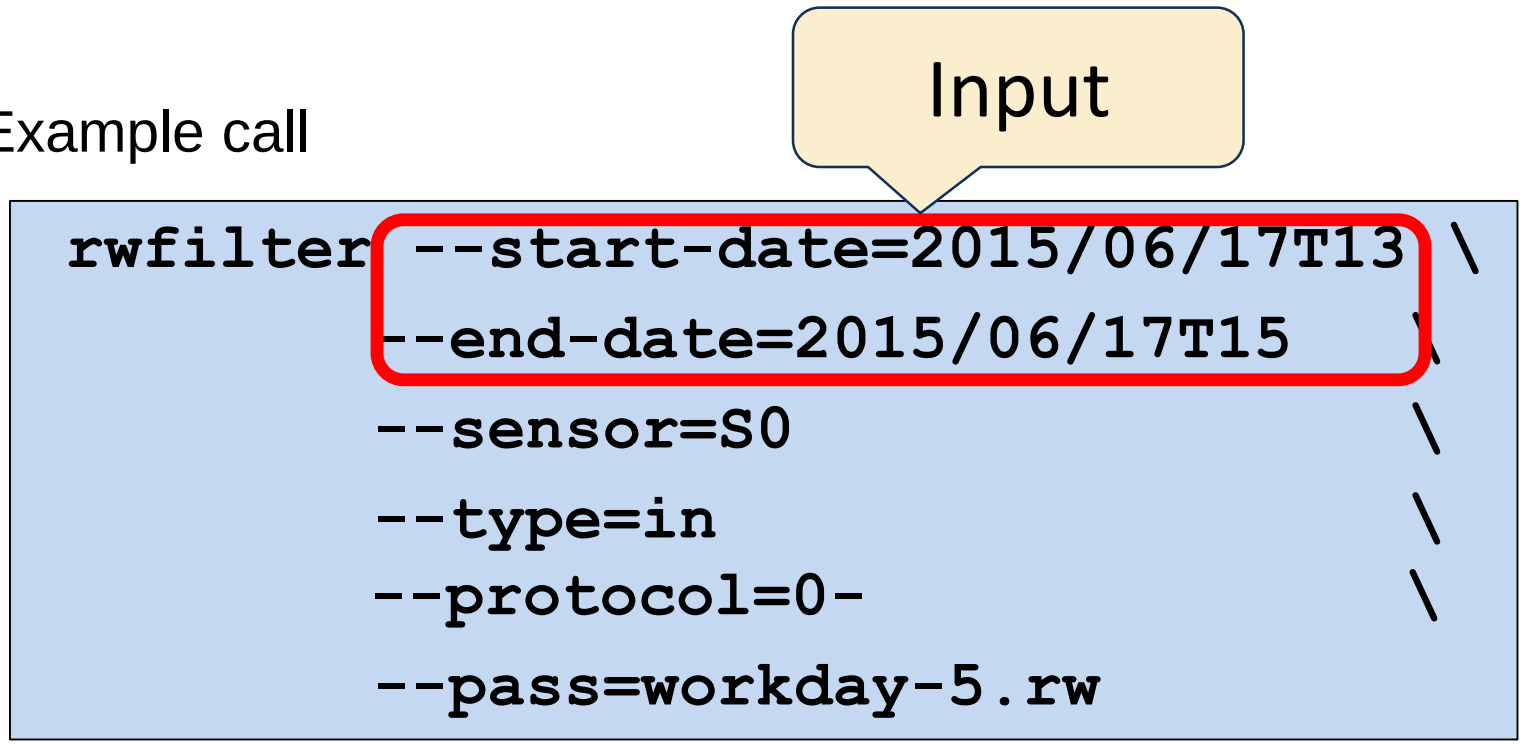
```
rwfilter --start-date=2015/06/17T13 \
         --end-date=2015/06/17T15   \
         --sensor=S0                \
         --type=in                  \
         --protocol=0-              \
         --pass=workday-5.rw
```

`rwfilter` Syntax

General form

```
rwfilter {INPUT | SELECTION}
  PARTITION OUTPUT [OTHER]
```

Example call



`rwfilter` Syntax

General form

```
rwfilter {INPUT | SELECTION}
  PARTITION OUTPUT [OTHER]
```

Selection

Example call

```
rwfilter --start-date=2015/06/17T13 \
         --end-date=2015/06/17T15   \
         --sensor=S0                \
         --type=in                  \
         --protocol=0-              \
         --pass=workday-5.rw
```

`rwfilter` Syntax

General form

```
rwfilter {INPUT | SELECTION}
  PARTITION OUTPUT [OTHER]
```

Example call

```
rwfilter --start-date=2015/06/17T13 \
         --end-date=201                \
         --sensor=S0                   \
         --type=in                     \
         --protocol=0-                 \
         --pass=workday-5.rw
```

`rwfilter` Syntax

General form

```
rwfilter {INPUT | SELECTION}
  PARTITION OUTPUT [OTHER]
```

Example call

```
rwfilter --start-date=2015/06/17T13 \
         --end-date=2015/06/17T15   \
         --sensor=S0                \
         --type=in                  \
         --protocol=0-              \
         --pass=workday-5.rw
```

Output

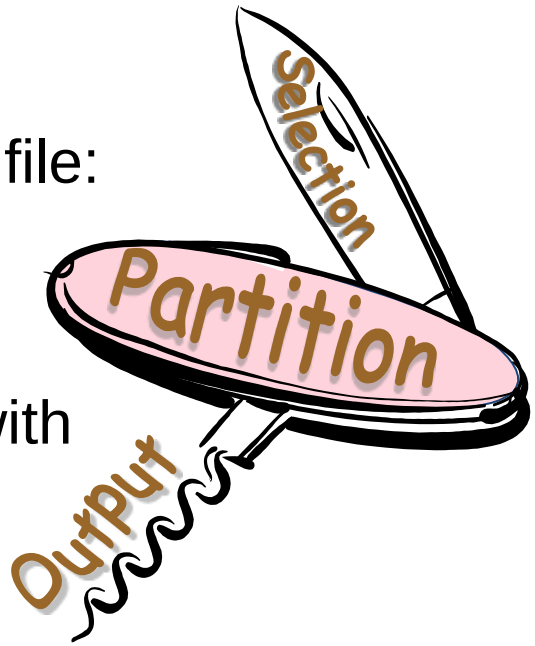
Selection and Input Criteria

Selection options control access to repository files:

- **`--start-date=2015/06/16`**
- **`--end-date=2015/6/17T03`**
- **`--sensor=S0`**
- **`--type=in,inweb`**

Alternatively, use input criteria for a pipe or a file:

- **`myfile.rw`**
- **`stdin`**
- useful for chaining filters through a pipe with stdin/stdout



Basic Partitioning Options

Simple numeric fields: ports, protocol, ICMP Type

Specified IP addresses, CIDR blocks

Sets of IP addresses

Combinations of key fields – Tuple files

Simple Numeric Key Fields

```
--protocol=
--sport= --dport= --aport=                  # source, dest, any

--protocol=6,17                             # TCP or UDP
--protocol=0-5,7-16,18-                     # not TCP or UDP
--protocol=0-                               # all protocols
--dport=80,443                              # HTTP or HTTPS
--sport=6000-6063,9100-9107                 # X11 or JetDirect
--aport=20,21                               # FTP
--sport=0-1023                              # Well-Known Ports
--itype=08                                  # ICMP type value
```

Specified IP address or CIDR block

--**s**address= --**d**address= --any-address=

--not-**s**address= --not-**d**address= --not-any-address=

May specify a single:

IP address 192.0.2.1

CIDR block 192.0.2.0/24

Specified IP addresses or CIDR blocks

--**s**cidr=	--**d**cidr=	--any-cidr=
--not-**s**cidr=	--not-**d**cidr=	--not-any-cidr=

May specify multiple:

IP addresses	192.0.2.1,198.51.100.3
CIDR blocks	192.0.2.0/24,198.51.100.0/24
mixture	192.0.2.1,192.0.2.8/29

Sets of arbitrary addresses

--**s**ipset= --**d**ipset= --anyset=
--not-**s**ipset= --not-**d**ipset= --not-anyset=

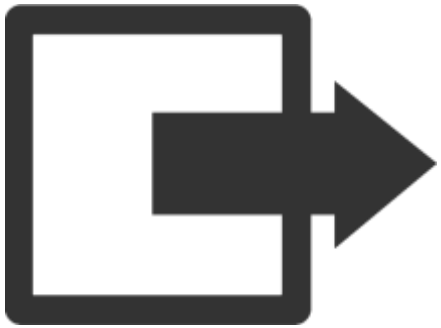
Specifies the name of a file storing the IP set:

--**s**ipset=internalservers.set
--**d**ipset=RussianBizNtwk.set
--anyset=TorNodes.set
--not-**d**ipset=whitelist.set

http://www.ikea.com/us/en/catalog/products/50149560/

rwfilter output options

```
--pass-destination=          # file to get records that pass
--fail-destination=          # file to get records that fail
--all-destination=           # file to get all records

--print-statistics           # report recs read/pass/fail
--print-volume-statistics    # report how many
                             # recs/pkts/bytes pass/fail
```



https://gotomydevices.com/

What is going on here? -5

```
rwfilter --start=2015/06/17T00  \
         --end=2015/06/18T07    \
         --type=in              \
         --sensor=S0            \
         --daddress=10.0.40.20  \
         --print-volume-stat
```

```
     |   Recs|  Packets|      Bytes|     Files|
Total| 848153|  5728597| 1870030015|        32|
 Pass| 684018|  2444635|  311976115|          |
 Fail| 164135|  3283962| 1558053900|          |
```

Exercise 4: `rwfilter`

1) Find all traffic going outbound to external HTTPS servers on June 17, 2015. Save these flows in file https0617.rw. Only pull records captured by sensor S0.

2) How many flow records matched the criteria?

Exercise 4: `rwfilter`

1) Find all traffic going outbound to external HTTPS servers on June 17, 2015. Save these flows in file https0617.rw. Only pull records captured by sensor S0.

2) How many flow records matched the criteria?

Hint

HTTPS normally uses port 443

Exercise 4: `rwfilter` solution

```
rwfilter --start=2015/06/17 --sensor=S0 \
      --type=outweb --dport=443         \
      --pass=https0617.rw               \
      --print-volume-statistics
```

```
     |    Recs|   Packets|       Bytes|   Files|
Total|  250623|   1843420|   330464744|      24|
 Pass|    1299|      4718|      480858|        |
 Fail|  249324|   1838702|   329983886|        |
```

```
rwfileinfo https0617.rw --fields=count
```

```
https0617.rw:
  count-records        1299
```

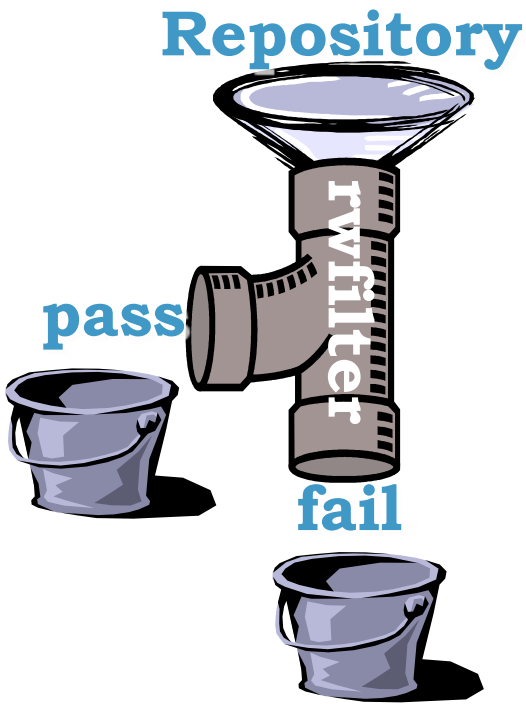
Output Criteria

rwfilter leaves the flows in binary (compact) form.

- **`--pass`**, **`--fail`**: direct the flows to a file or a pipe
- **`--all`**: destination for everything pulled from the repository
- One output is required but more than one can be used (screen not allowed for non-text data).

Other useful output

- **`--print-statistics`** `or` **`--print-volume-statistics`**
- **`--print-filenames`**, **`--print-missing-files`**



What is this? -8

```
rwfilter                                \
      --start-date=2015/06/17 \
      --type=outweb                     \
      --bytes=100000-                   \
      --pass=stdout                     \
|  rwfilter \
      stdin \
      --duration=60-          \
      --pass=long-http.rw \
      --fail=short-http.rw
```

One day's outgoing web, but only if 100,000 or more bytes per flow

Chain two rwfilter calls

One minute or more -> long
Less than one minute -> short

Answer: Classifies 100,000+-byte web output flows by fast or slow transfer. Bursty vs. Persistent?

What is this? -8 Results

```
 ./wit8.sh
What is this? #8
Classifies 100,000+-byte web output flows by fast or slow transfer.
     |   Recs|   Packets|       Bytes|  Files|
Total|   1724|   1370526|   619721909|      1|
 Pass|     50|    244021|    57616543|       |
 Fail|   1674|   1126505|   562105366|       |

-rw-r--r--. 1 pnk pnk   264 Sep 25 16:41 ex3arecords.rw
-rw-r--r--. 1 pnk pnk  1049 Sep 24 22:28 ex3records.rw
-rw-r--r--. 1 pnk pnk  1595 Sep 28 19:21 long-http.rw
-rw-r--r--. 1 pnk pnk 29148 Sep 28 19:21 short-http.rw

long-http.rw:
  count-records       50
short-http.rw:
  count-records       1674
```

Example Typos

--port= --destport= --sip= or --dip=	No such keywords
--saddress=danset.set	Needs addr not filename
--start-date=2006/06/12--end-date	Space needed
--start-date = 2006/06/12	No spaces around equals
start-date=2006/06/12	Need dashes
---start-date=2006/06/12	Only two dashes
--start-date=2005/11/04:06:00:00 --end-date=2005/05/21:17:59:59	Only down to hour

SiLK Usage Guidelines

1. Use Sets instead of using several rwfilter commands to pull data for multiple IP addresses
2. Store intermediate data on local disks, not network disks.
3. Make initial pulls from the repository, store the results in a file, and work on the file from then on. The repository is slower than processing a single file.
4. Work in binary for as long as possible. ASCII representations are much larger and slower than the binary representations of SiLK data.
5. Only filter a week of traffic at a time. The filter runs for excessive length of time otherwise.
6. Only run a few rwfilter commands at once. Don't overload the system.
7. Always specify the type of traffic to filter. Don't count on the defaults.
8. Appropriately label all output.
9. Always check that SiLK does not provide a feature before building your own.

Lesson II.1 Summary

We learned how to display the fields of interest from flow records.

Files are chosen from the repository with selection options. Records are chosen from those files with partitioning options.

There are lots of ways to partition on IP addresses.

Lesson II.2
Analysis Tools and Categorization

Basic SiLK Counting Tools: `rwcount, rwstats, rwuniq`

"Count [**volume**] by [**key field**] and print [**summary**]"

- basic bandwidth study:
  - "Count **bytes** by **hour** and print the **results**."
- top 10 talkers list:
  - "Count **bytes** by **source IP** and print the **10 highest IPs**."
- user profile:
  - "Count **records** by **dIP-dPort pair** and print **all the pairs**."
- potential scanners:
  - "Count **unique dIPs** by **sIP** and print the **sources that contacted more than 100 destinations**."

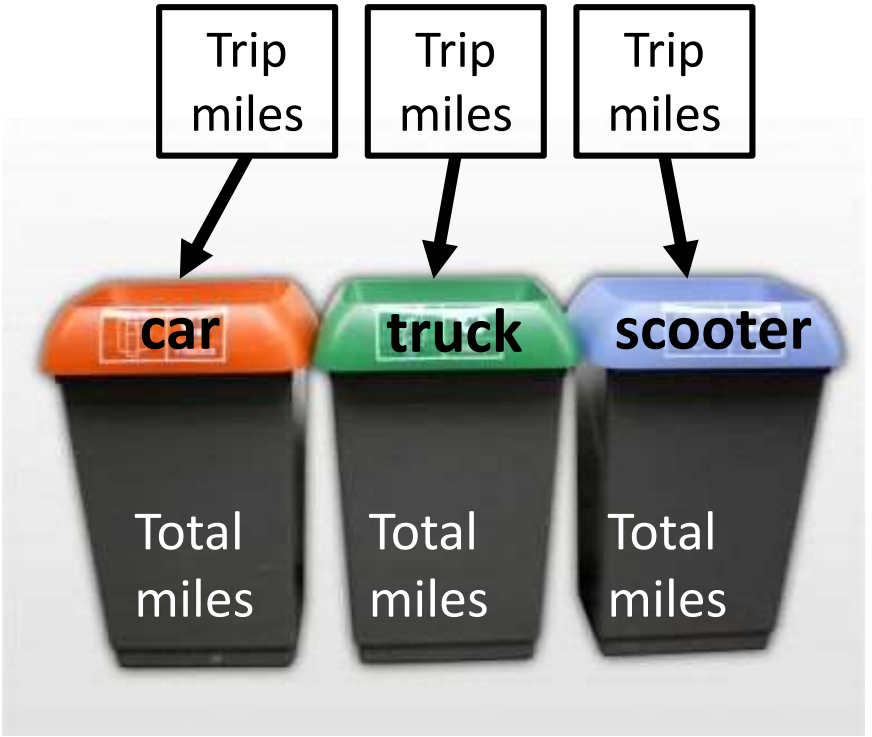
Categorization—Bins

For motor vehicle trips we could bin trip records by

- vehicle style – sedan, coupe, SUV, pickup, van
- highway or city trip
- personal or business trip

We could measure the trips and aggregate in bins

- total miles
- fuel consumption
- oil consumption
- pollutant emission



http://www.prlog.org/10991533-great-value-good-looking-colour-coded-recycling-bins-exclusive-to-imrubbishcouk.html

Bins

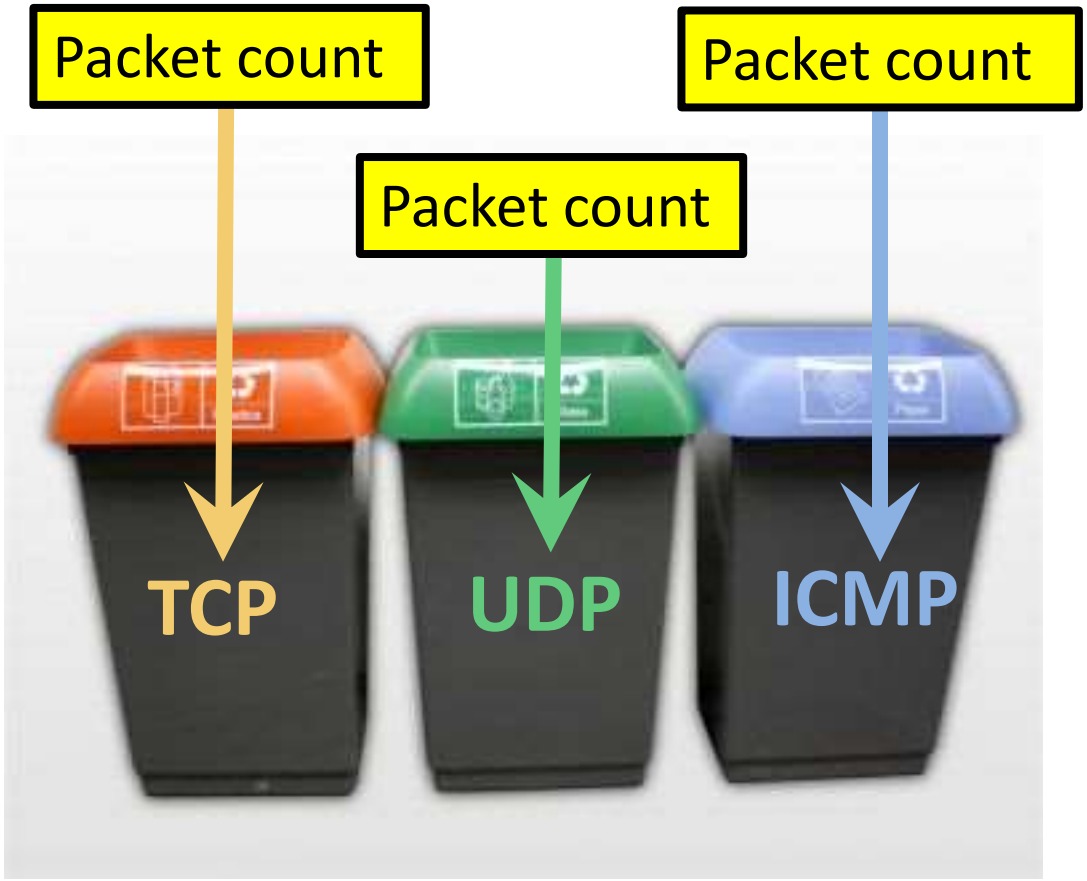
For flows we could bin by

- address or address block
- port
- protocol
- time period

We could measure the flows and aggregate in bins

- count of flow records, packets, bytes
- count of distinct values of other fields, e.g., addr
- earliest sTime, latest eTime

Bins



Total packets

Total packets

Total packets

Value from flow record
e.g., packets

Bin key field
e.g., protocol

Aggregate Value

Basic SiLK Counting Tools: **`rwcount, rwstats, rwuniq`**

`rwcount`: count **volume** across **time periods**

`rwstats`: count **volume** across **IP, port, or protocol** and create descriptive statistics

`rwuniq`: count **volume** across **any combination of SiLK fields**

"Key field" = SiLK fields defining bins

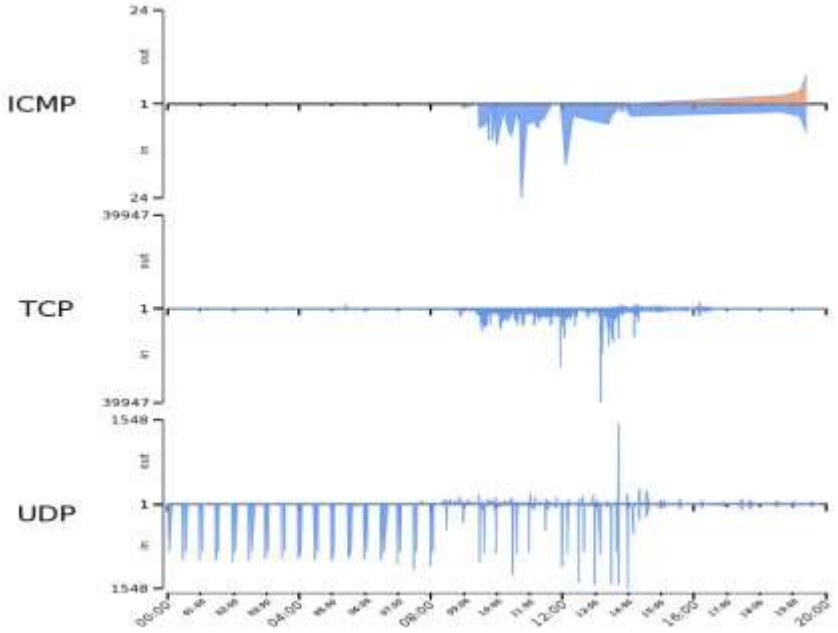
"Volume" = {Records, Bytes, Packets} and a few others

- measure
- aggregate value

Each tool reads raw binary flow records as input.

rwcount

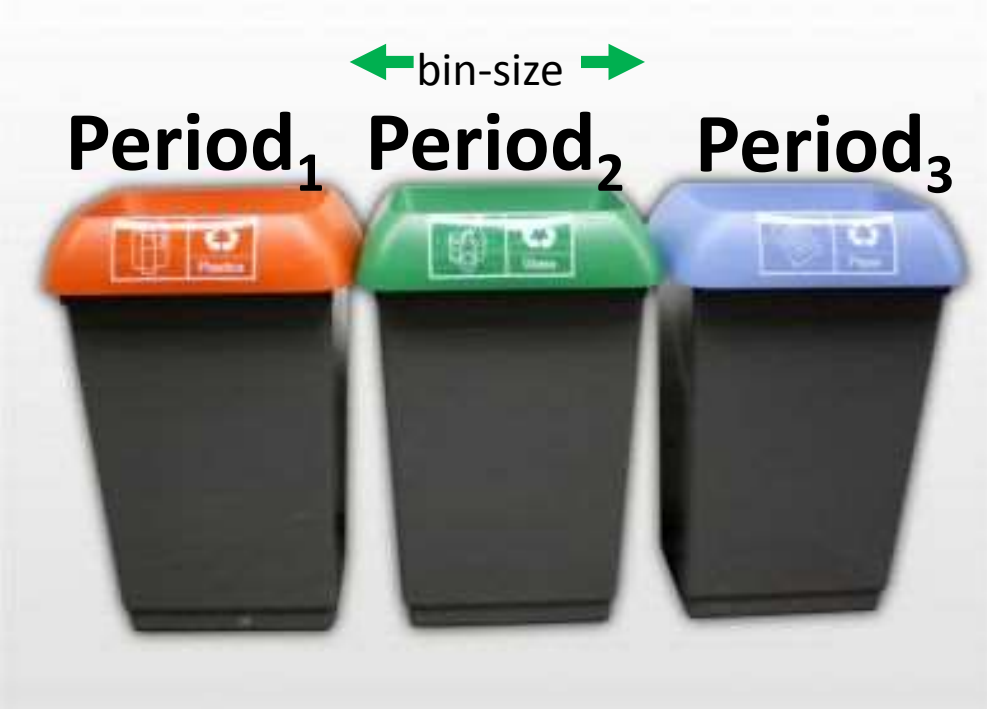
- count records, bytes, and packets by time and display results
  `rwcount --bin-size=300`
- fast, easy way of summarizing volumes as a time series
- great for simple bandwidth studies
- easy to take output and make a graph with graphing S/W



Time Bins

When binning by time, you must specify the period of time for each bin.

- This is called the **bin-size**.
  - It's the size of the bin's opening, not the volume of the container.



rwcount

The bin key is always time. You choose the period.

The aggregate measures are chosen for you. They are flows (records), bytes, packets.

```
rwfilter --sensor=S0 --start=2015/06/17 \
         --type=in --proto=1 \
         --pass=stdout       \
  | rwcount --bin-size=3600
```

```
                 Date|  Records|       Bytes|   Packets|
...
2015/06/17T16:00:00|   696.85| 1714385.00| 18345.11|
2015/06/17T17:00:00|   701.73| 1763170.55| 18781.86|
2015/06/17T18:00:00|   528.15| 1205335.60| 12434.30|
2015/06/17T19:00:00|    51.00|   20048.00|   239.00|
...
```

What is this? -9

```
rwcount MSSP.rw --bin-size=3600
```

```
                Date|    Records|        Bytes|    Packets|
2010/12/08T00:00:00| 1351571.66| 73807086.40| 1606313.61|
2010/12/08T01:00:00| 1002012.43| 54451440.59| 1185143.62|
2010/12/08T02:00:00| 1402404.61| 77691865.26| 1675282.27|
2010/12/08T03:00:00| 1259973.65| 68575249.90| 1491393.08|
2010/12/08T04:00:00|  939313.56| 51410968.24| 1118584.81|
2010/12/08T05:00:00|  459564.75| 80862273.32| 1742058.62|
2010/12/08T06:00:00| 1280651.23| 69881126.41| 1519435.24|
...
```

Demo: **`rwcount`**

The shell can help with the arithmetic: $((24*60*60))

You also can find common periods in the Quick Reference Guide.

Time series for all outgoing traffic on sensor S0:

```
rwfilter --start=2015/06/10     \
         --end=2015/06/19       \
         --sensor=S0            \
         --type=out,outweb      \
         --proto=0-             \
         --pass=stdout          \
 | rwcount --bin-size=$((24*60*60))
```

Exercise 5: `rwcount`

Produce a time-series with 30-minute intervals, analyzing incoming ICMP traffic collected at sensor S0 on June 17, 2015.

Exercise 5: `rwcount`

Produce a time-series with 30-minute intervals, analyzing incoming ICMP traffic collected at sensor S0 on April 20, 2009.

HINT

ICMP is Protocol 1

Exercise 5: `rwcount` solution

```
rwfilter --start=2015/06/17     \
         --sensor=S0            \
         --type=in              \
         --proto=1              \
         --pass=stdout          \
  | rwcount --bin-size=$((30*60))
```

```
                 Date|  Records|        Bytes|   Packets|
2015/06/17T00:00:00|   359.16|    758956.99|   7424.53|
2015/06/17T00:30:00|   347.74|    765794.49|   7491.36|
2015/06/17T01:00:00|   336.27|    813086.05|   7956.58|
2015/06/17T01:30:00|   331.99|    762438.18|   7477.55|
2015/06/17T02:00:00|   345.25|    845688.88|   8297.25|
2015/06/17T02:30:00|   305.97|    801129.61|   7796.63|
2015/06/17T03:00:00|   337.27|    798297.54|   7801.01|
2015/06/17T03:30:00|   344.94|    751994.96|   7366.56|
...
```

rwuniq

rwuniq will display all bins for a particular field or fields.

Output is normally unsorted.

--sort-output causes sorting by the key (bin).

Calling **`rwuniq`**

`rwuniq --fields=KEY --value=VOLUME`

- Choose one or several key fields.
- Aggregate volume count: records, bytes, or packets.
- standard output formatting options (see “man rwuniq”)

Apply thresholds to bins before outputting:

- **`--bytes`**, **`--packets`**, **`--flows`**, **`--sip-distinct`**, **`--dip-distinct`**
- Specify minimum aggregate value or a range

`--sort-output` by key (**`rwstats`** sorts by value)

Profiling a Network with **rwuniq**

```
rwfilter --start=2015/06/17     \
         --sensor=S0            \
         --type=in              \
         --proto=0-             \
         --dport=0-1024         \
         --pass=stdout          \
| rwuniq --fields=proto,dport    \
         --ipv6-policy=ignore   \
         --sort-output
```

pro	dPort	Records
1	0	1835
1	769	7738
1	770	200
1	771	1407
1	778	22
6	21	2
6	53	4
6	88	3568
6	135	1882
6	139	12458

pro	dPort	Records
6	389	4181
6	445	16726
17	0	10
17	53	311066
17	88	188
17	123	1350
17	137	2472
17	138	1206
17	389	4373
17	514	391

What is this? -10

```
rwfilter outtraffic.rw        \
      --stime=2010/12/08:18:00:00-2010/12/08:18:59:59.999 \
      --saddress=71.55.40.62 \
      --pass=stdout          \
  | rwuniq --fields=dip,sport \
           --all-counts       \
           --sort-output
```

```
           dIP|sPort |     Bytes|Packets| Records|     sTime-Earliest|       eTime-Latest|
 12.113.41.190|   80 |     12782|     20|       4|2010/12/08T18:42:51|2010/12/08T18:58:49|
30.182.228.143|   80 | 203907933| 143611|       2|2010/12/08T18:53:59|2010/12/08T19:01:47|
 37.153.24.229|   80 | 205628625| 144829|       2|2010/12/08T18:29:11|2010/12/08T18:42:51|
 82.180.203.87|   80 | 213013145| 150896|      92|2010/12/08T18:06:36|2010/12/08T18:32:33|
82.180.203.197|   80 |       800|      8|       2|2010/12/08T18:43:30|2010/12/08T18:43:30|
88.124.166.233|   80 | 223930369| 158276|      97|2010/12/08T18:08:55|2010/12/08T18:32:25|
88.124.166.233|  443 |    509285|    732|      43|2010/12/08T18:06:57|2010/12/08T18:51:11|
94.239.226.247|   80 | 124833037|  96047|       3|2010/12/08T18:25:22|2010/12/08T19:21:34|
  109.95.61.80|   80 |   8467397|   6325|      90|2010/12/08T18:08:59|2010/12/08T18:10:09|
  139.65.186.4|   80 | 204123360| 143794|       3|2010/12/08T18:19:48|2010/12/08T18:26:36|
139.177.10.136|   80 | 407978375| 287354|       6|2010/12/08T18:20:03|2010/12/08T19:01:30|
198.237.16.172|   80 | 159066748| 112025|       1|2010/12/08T18:18:43|2010/12/08T18:46:55|
219.149.72.154| 1024 |        44|      1|       1|2010/12/08T18:50:40|2010/12/08T18:50:40|
249.216.88.172|   80 |        88|      2|       2|2010/12/08T18:44:42|2010/12/08T18:44:47|
250.211.100.88|   80 |   3295160|   2492|      42|2010/12/08T18:47:50|2010/12/08T18:58:53|
```

What is this? -11

```
rwuniq outtraffic.rw        \
       --fields=dip         \
       --values=sip-distinct,records,bytes  \
       --sip-distinct=400- \
       --sort-output
```

```
             dIP|sIP-Distin|     Bytes| Records|
   13.220.28.183|        512|     20480|     512|
    171.128.2.27|        448|  19069280|  476732|
   171.128.2.179|        448| 139501200| 3487530|
  171.128.212.14|        448| 139467440| 3486686|
 171.128.212.124|        448| 127664480| 3191612|
 171.128.212.127|        448|  66611560| 1665289|
 171.128.212.188|        448| 139467680| 3486692|
 171.128.212.228|        448| 139393160| 3484829|
 245.225.153.120|        763|     30520|     763|
 245.238.193.102|       1339|    179480|    4487|
```

Exercise 6: **`rwuniq`**

For outgoing flows from S0 on June 17, 2015 write and execute the **`rwfilter`** piped to **`rwuniq`** commands to list how many TCP flows (records) there were with each different number of packets. Display sorted by the number of packets.

Are there any odd results you can explain?

Exercise 6: **`rwuniq`**

For outgoing flows from S0 on June 17, 2015 write and execute the **`rwfilter`** piped to **`rwuniq`** commands to list how many TCP flows (records) there were with each different number of packets. Display sorted by the number of packets.

Are there any odd results you can explain?

HINT

TCP is protocol 6

Exercise 6: **`rwuniq`**

For outgoing flows from S0 on June 17, 2015 write and execute the **`rwfilter`** piped to **`rwuniq`** commands to list how many TCP flows (records) there were with each different number of packets. Display sorted by the number of packets.

Are there any odd results you can explain?

HINT

TCP is protocol 6 (**`proto=6`**)

The TCP 3-way handshake requires 3 packets

Exercise 6: `rwuniq` Solution

```
rwfilter --type=out,outweb      \
         --sensor=S0            \
         --start=2015/06/17     \
         --proto=6              \
         --pass=stdout          \
| rwuniq --fields=packets --sort-output
```

```
   packets|   Records|
         1|      3513|
         2|      6857|
         3|     30932|
         4|     10159|
         5|    123506|
         6|     52155|
         7|     46961|
         8|     42182|
         9|     24032|
        10|      4662|
        11|     29573|
        12|     17977|
 ...
     14517|         1|
     14723|         1|
     50069|         1|
```

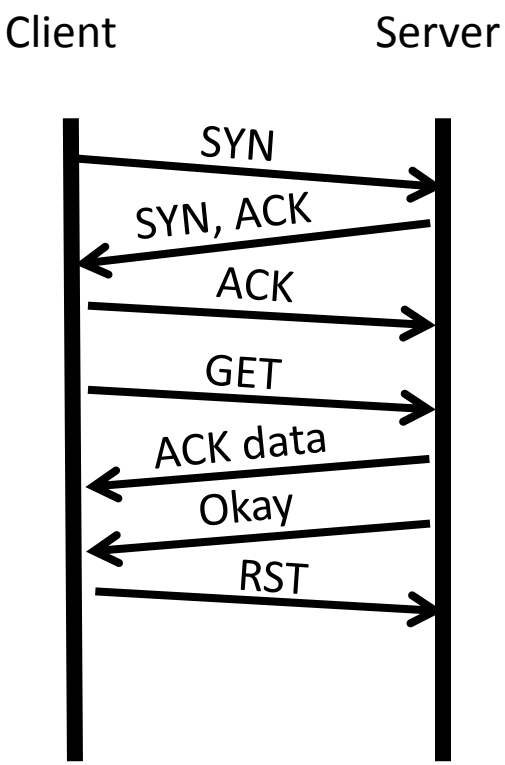
What can you say about flows with 1, 2 and 3 packets?

It seems as though 4 packets is an oddity.

Do you have an explanation? What can be accomplished with 4 TCP packets?

There are, of course, exceptions.

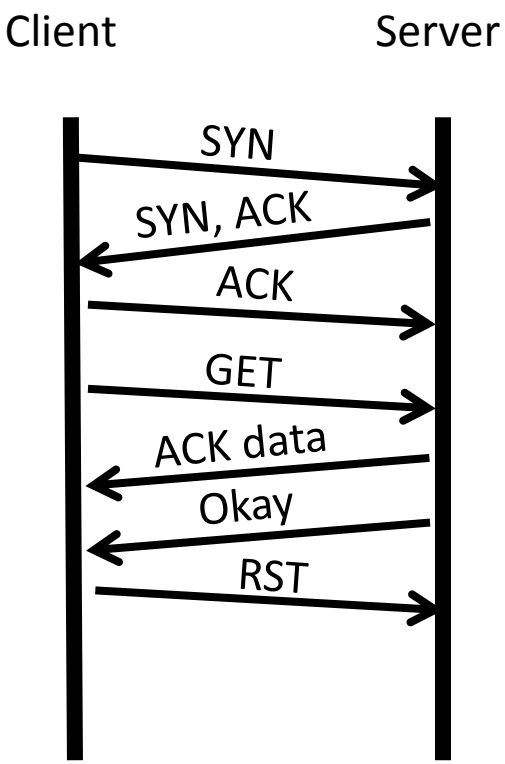
TCP packets per flow -1



How many flows are there?

How many packets are there?

TCP packets per flow -2

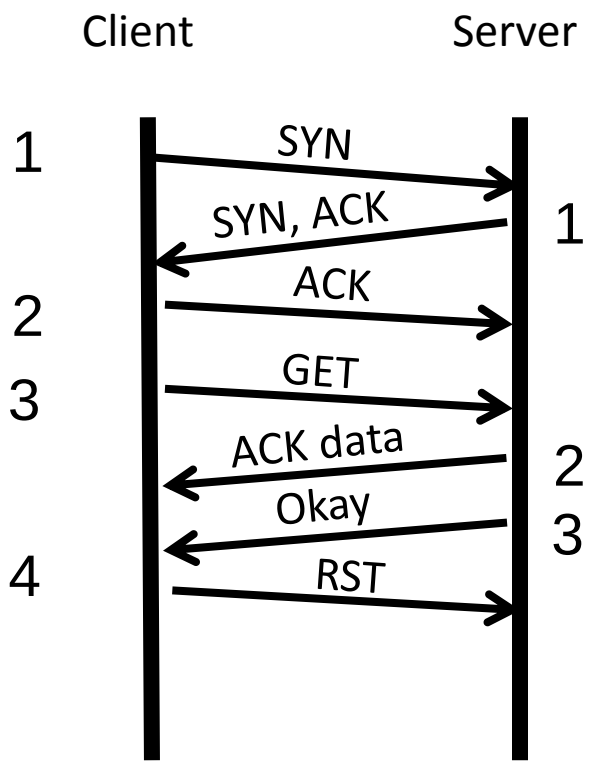


How many flows are there?

There are 2 flows

- One from client to server
- One from server to client

TCP packets per flow -3



How many flows are there? **2**

How many packets are there?

There are 7 packets

- 4 packets from client to server
- 3 packets from server to client

rwstats

Like rwuniq, rwstats displays bins for a field or fields, but only displays the top N or bottom N bins.

The top/bottom N is determined by some traffic volume measurement, such as flows, packets, or bytes.

The bins are displayed sorted by the measurement.

It also provides percentages.

Calling **rwstats**

rwstats --overall-stats

- Descriptive statistics on byte and packet counts by record
- See “man rwstats” for details.

```
rwstats     --fields=KEY --value=VOLUME
            --count=N or --threshold=N or
            --percentage=N
            [--top or --bottom]
```

- Choose one or two key fields.
- Count one of records, bytes, or packets.
- Great for Top-N lists and count thresholds
- Standard output formatting options (see “man rwstats”)

Output from `rwstats --overall-stats`

```
FLOW STATISTICS--ALL PROTOCOLS:  455320 records
*BYTES min 40; max 68282013
  quartiles LQ 198.77566 Med 349.93619 UQ 2334.92758 UQ-LQ 2136.15192
   interval_max|count<=max|%_of_input|   cumul_%|
             40|      2275|  0.499649|  0.499649|
             60|      1235|  0.271238|  0.770886|
            100|      5504|  1.208820|  1.979707|
            150|     19402|  4.261179|  6.240886|
            256|    185623| 40.767592| 47.008478|
           1000|    107882| 23.693666| 70.702144|
          10000|    131933| 28.975885| 99.678029|
         100000|      1072|  0.235439| 99.913467|
        1000000|       385|  0.084556| 99.998023|
     4294967295|         9|  0.001977|100.000000|
*PACKETS min 1; max 50069
  quartiles LQ 5.27501 Med 7.60205 UQ 9.92908 UQ-LQ 4.65407
   interval_max|count<=max|%_of_input|   cumul_%|
              3|     41302|  9.070983|  9.070983|
              4|     10159|  2.231178| 11.302161|
             10|    293498| 64.459721| 75.761882|
             20|     87084| 19.125889| 94.887771|
             50|     21614|  4.746991| 99.634762|
            100|       887|  0.194808| 99.829570|
            500|       706|  0.155056| 99.984626|
           1000|         7|  0.001537| 99.986164|
          10000|        60|  0.013178| 99.999341|
     4294967295|         3|  0.000659|100.000000|
```

```
*BYTES/PACKET min 40; max 1463
  quartiles LQ 42.18817 Med 49.47921 UQ 99.70154
UQ-LQ 57.51338
   interval_max|count<=max|%_of_input|   cumul_%|
             40|     26450|  5.809101|  5.809101|
             44|    159732| 35.081262| 40.890363|
             60|    121121| 26.601291| 67.491654|
            100|     34444|  7.564790| 75.056444|
            200|     74203| 16.296890| 91.353334|
            400|     36073|  7.922560| 99.275894|
            600|      2506|  0.550382| 99.826276|
            800|       299|  0.065668| 99.891944|
           1500|       492|  0.108056|100.000000|
     4294967295|         0|  0.000000|100.000000|
```

What is this? -12

```
rwfilter outtraffic.rw \
         --stime=2010/12/08T18:00:00-2010/12/08T18:59:59 \
         --pass=stdout \
 | rwstats --fields=sip \
           --values=bytes  \
           --count=10
```

```
INPUT: 1085277 Records for 1104 Bins and 4224086177 Total Bytes
OUTPUT: Top 10 Bins by Bytes
                sIP|      Bytes|    %Bytes|   cumul_%|
         71.55.40.62| 1754767148| 41.541935| 41.541935|
        71.55.40.169| 1192063164| 28.220617| 69.762552|
        71.55.40.179|  331310772|  7.843372| 77.605923|
        71.55.40.204|  170966278|  4.047415| 81.653338|
      177.249.19.217|  122975880|  2.911301| 84.564639|
         71.55.40.72|  110726717|  2.621318| 87.185957|
        71.55.40.200|  101593627|  2.405103| 89.591060|
      177.71.129.255|   40166574|  0.950894| 90.541954|
         71.55.40.91|   35316554|  0.836076| 91.378030|
     149.249.114.204|   26634602|  0.630541| 92.008571|
```

Exercise 7: `rwstats`

What are the top 10 incoming protocols on June 17, 2015, collected on sensor S0?

Exercise 7: **`rwstats`**

What are the top 10 incoming protocols on June 17, 2015 , collected on sensor S0?

HINT

Incoming flows have type **`in`** or **`inweb`**

Exercise 7: `rwstats` solution

```
rwfilter --start=2015/06/17  \
       --type=in,inweb       \
       --sensor=S0           \
       --proto=0-  --pass=stdout    \
 | rwstats --fields=protocol \
           --value=records --count=10
```

```
INPUT: 1090580 Records for 3 Bins and 1090580 Total
Records
OUTPUT: Top 10 Bins by Records
pro|   Records|  %Records|   cumul_%|
 17|    642167| 58.883071| 58.883071|
  6|    435275| 39.912249| 98.795320|
  1|     13138|  1.204680|100.000000|
```

Exercise 8: `rwstats`

Find the top 9 inside hosts according to how many outside hosts they communicate with on June 17, 2015 collected on sensor S0?

Exercise 8: `rwstats`

Find the top 9 inside hosts according to how many outside hosts they communicate with on June 17, 2015 , collected on sensor S0?

HINT

Use

`--value=distinct:dip`

Exercise 8: `rwstats` solution

```
rwfilter --start-date=2015 --sensor=S0   \
       --type=out,outweb  --proto=0-     \
       --pass=stdout  \
 | rwstats --fields=sip --value=distinct:dip \
       --no-percents --count=9
```

```
INPUT: 2321845 Records for 47 Bins and 208 Total
dIP-Distinct
OUTPUT: Top 9 Bins by dIP-Distinct
            sIP|dIP-Distin|
    10.0.40.20|       147|
    10.0.40.21|        87|
    10.0.40.53|        73|
    10.0.20.58|        15|
    10.0.40.27|        14|
    10.0.50.21|         9|
    10.0.50.11|         8|
    10.0.50.22|         8|
    10.0.50.16|         6|
```

--no-percents leaves out the percent and cumulative columns

rwuniq VS. **rwstats** -1

rwuniq	both	rwstats in top/bottom mode
all bins except per thresholds	Bin by key	**--top** or **--bottom** bins
	Default aggregate value is flows (records).	
--sort-output by key otherwise unsorted		Sorted by primary aggregate value
Thresholds or ranges: --bytes, --packets, --flows, --sip-distinct, --dip-distinct	Choose which bins have aggregate values significant enough to output.	**--count**, **--threshold**, **--percentage**

`rwuniq` vs. `rwstats` -2

rwuniq	both	rwstats in top/bottom mode
--all-counts (bytes, pkts, flows, earliest sTime, and latest eTime)	Show volume aggregate value[s].	**--no-percents** (good when primary aggregate isn't Bytes, Packets, or Records)
	--bin-time to adjust sTime and eTime	
	--presorted-input (omit when value includes Distinct fields, even if input is sorted)	
--values= sTime-Earliest, eTime-Latest	**--values**=Records, Packets, Bytes, sIP-Distinct, dIP-Distinct, Distinct:KEY-FIELD (KEY-FIELD can't also be key field in --fields)	

Lesson II.2 Summary

We learned how to

- Categorize flow records by time or some other field
- Display a time series of flows with rwcount
- Display all categories (bins) with rwuniq
- Display the top or bottom bins, according to some measurement, with rwstats

Lesson II.3
Tuple Files and PMAPs

Tuple files

Is there a way to search for multiple, independent field values without resorting to multiple rwfilter commands?

Yes, it is called a Tuple and it can be used in addition to or instead of other partitioning parameters

For example, use a tuple file instead of

```
--proto=6
--dport=25,58,143,158,209,366,465,587
```

```
rwfilter … -tuple-file=email-ports.txt …
```

Using a Tuple file

We are interested in flows using certain combinations of ports and protocols, specifically.

First we will try it with what we already know and then with a Tuple File.

Protocol	Port
6	445
17	135 through 139

Protocol, Port Combinations without Tuple File

```
rwfilter --start=2015/06/14     \
         --end=2015/06/19       \
         --type=in,out          \
         --dport=135,136,137,138,139,445 \
         --proto=6,17           \
         --pass=stdout          \
| rwuniq --fields=proto,dport
```

```
pro|dPort|   Records|
 17|  137|    835554|
 17|  138|     13739|
  6|  135|     63455|
  6|  139|    143788|
  6|  137|      6084|
  6|  445|    209266|
```

These aren't the ones we are looking for

Protocol, Port Combinations with Tuple File

```
echo "Tuple file msft.tuple"
cat msft.tuple
echo
rwfilter --start=2015/06/14     \
         --end=2015/06/19       \
         --type=in,out          \
         --tuple-file=msft.tuple \
         --pass=stdout          \
| rwuniq --fields=proto,dport
```

```
Tuple file msft.tuple
proto|dport
  6|445
17|135,136,137,138,139

pro|dPort|   Records|
 17|  137|    835554|
 17|  138|     13739|
  6|  445|    209266|
```

Prefix Map (PMap)

How do I work with, say, service names like HTTP and HTTPS rather than 80 and 443?

Use a PMap, short for Prefix Map

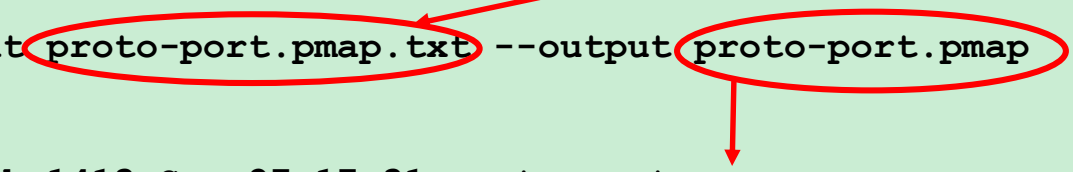
- **`rwpmapbuild`**
- **`rwpmapcat`**
- **`rwpmaplookup`**

Let's build a PMAP -1

```
$ cat proto-port.pmap.txt
map-name   protocol-port-example
mode       proto-port

#    The range is either a single protocol or a protocol and
#    a port separated by a slash.
1      1           ICMP
1/0    1/255       ICMP/Echo Reply
#    Specify the wider categories first, then specialize
6      6           TCP
6/0    6/1024      TCP/Low Port
#    A range of a single port requires both the starting
#    value and the ending value
6/21   6/21        TCP/FTP
6/22   6/22        TCP/SSH
6/25   6/25        TCP/SMTP
6/80   6/80        TCP/HTTP
6/135  6/135       UDP/EPMAP
6/136  6/136       UDP/Unknown
6/137  6/137       UDP/Netbios NS
6/138  6/138       UDP/Netbios DG
6/139  6/139       UDP/Netbios SS
6/443  6/443       TCP/HTTPS
6/445  6/445       TCP/SMB
17     17          UDP
17/0   17/1024     UDP/Low Port
17/53  17/53       UDP/DNS
17/67  17/68       UDP/DHCP
17/135 17/135      UDP/EPMAP
17/136 17/136      UDP/Unknown
17/137 17/137      UDP/Netbios NS
17/138 17/138      UDP/Netbios DG
17/139 17/139      UDP/Netbios SS
50     50          ESP
58     58          ICMPv6
$
```

Let's build a PMAP -2

```
 $ ls -l *pmap*
-rw-r--r--. 1 pnk netsa_analysis 934 Sep 26 22:12 proto-port.pmap.txt

$ rwpmapbuild --input proto-port.pmap.txt --output proto-port.pmap

$ ls -l *pmap*
-rw-r--r--. 1 pnk pnk 1413 Sep 27 17:31 proto-port.pmap
-rw-r--r--. 1 pnk pnk  934 Sep 26 22:12 proto-port.pmap.txt
$
```



Let's build a PMAP -3

```
$ rwpmapcat --map-file=proto-port.pmap
startPair|  endPair|          label|
      0/0|  0/65535|        UNKNOWN|
      1/0|    1/255|ICMP/Echo Reply|
    1/256|  1/65535|           ICMP|
      2/0|  5/65535|        UNKNOWN|
      6/0|     6/20|   TCP/Low Port|
     6/21|     6/21|        TCP/FTP|
     6/22|     6/22|        TCP/SSH|
     6/23|     6/24|   TCP/Low Port|
     6/25|     6/25|       TCP/SMTP|
     6/26|     6/79|   TCP/Low Port|
     6/80|     6/80|       TCP/HTTP|
     6/81|    6/134|   TCP/Low Port|
    6/135|    6/135|      UDP/EPMAP|
    6/136|    6/136|    UDP/Unknown|
    6/137|    6/137| UDP/Netbios NS|
    6/138|    6/138| UDP/Netbios DG|
    6/139|    6/139| UDP/Netbios SS|
    6/140|    6/442|   TCP/Low Port|
    6/443|    6/443|      TCP/HTTPS|
    6/444|    6/444|   TCP/Low Port|
    6/445|    6/445|        TCP/SMB|
    6/446|   6/1024|   TCP/Low Port|
   6/1025|   6/5999|            TCP|
   6/6000|   6/6063|        TCP/X11|
   6/6064|  6/65535|            TCP|
```

```
     7/0| 16/65535|        UNKNOWN|
    17/0|    17/52|   UDP/Low Port|
   17/53|    17/53|        UDP/DNS|
   17/54|    17/66|   UDP/Low Port|
   17/67|    17/68|       UDP/DHCP|
   17/69|   17/134|   UDP/Low Port|
  17/135|   17/135|      UDP/EPMAP|
  17/136|   17/136|    UDP/Unknown|
  17/137|   17/137| UDP/Netbios NS|
  17/138|   17/138| UDP/Netbios DG|
  17/139|   17/139| UDP/Netbios SS|
  17/140|  17/1024|   UDP/Low Port|...
```

Use the Pmap we just made -1

```
#
# show protocol port pmap in use

export SILK_DATA_ROOTDIR=/storage/2/static/FCC-xdata/releasable/silk

rwfilter --start=2015/06/17 \
        --end=2015/06/17 \
        --sensor=0      \
        --type=in,out   \
        --dport=0,21,22,25,80,135-139,443,445,768,1024,770,771  \
        --pass=stdout   \
| rwcut stdin           \
        --pmap-file=ProtoPort:proto-port.pmap   \
        --fields=sip,dip,sport,dst-ProtoPort,packets    \
        --ipv6-policy=ignore
```

Use the Pmap we just made -2

```
$ ./pmap-example.sh
            sIP|            dIP|sPort|  dst-ProtoPort|   packets|
192.168.143.239|     10.0.40.20|61721| UDP/Netbios SS|        24|
192.168.143.239|     10.0.40.20|61722| UDP/Netbios SS|        28|
192.168.143.239|     10.0.40.20|61723| UDP/Netbios SS|        28|
  192.168.40.20|     10.0.40.20|63948|        TCP/SMB|         2|
  192.168.40.20|     10.0.40.54|63952|        TCP/SMB|         2|
     155.6.4.10|     10.0.40.27|    0|ICMP/Echo Reply|         1|
 192.168.166.15|     10.0.40.20| 3709|      UDP/EPMAP|        10|
  192.168.40.20|     10.0.40.53|63954|        TCP/SMB|         2|
  192.168.40.20|     10.0.40.52|63955|        TCP/SMB|         2|
  192.168.40.20|     10.0.40.50|63956|        TCP/SMB|         2|
  192.168.143.5|     10.0.40.20|  137| UDP/Netbios NS|         2|
  192.168.40.25|     10.0.40.51|59510| UDP/Netbios SS|        14|
192.168.121.158|     10.0.40.20|53762|        TCP/SMB|        26|
  192.168.40.25|     10.0.40.50|59512| UDP/Netbios SS|        14|
  192.168.40.25|     10.0.40.50|59513| UDP/Netbios SS|        14|
 192.168.166.15|     10.0.40.20| 3711|      UDP/EPMAP|         4|
 192.168.166.15|     10.0.40.20| 3709|      UDP/EPMAP|         4|
  192.168.40.25|     10.0.40.53|59514| UDP/Netbios SS|        14|
  192.168.40.25|     10.0.40.53|59515| UDP/Netbios SS|        14|
  192.168.70.10|     10.0.40.20|    0|           ICMP|         4|
...
```

Part II Summary—Basic SiLK

rwsiteinfo

rwcut

rwfilter

rwcount

rwstats

rwuniq

Tuple files

rwpmapbuild

Challenge Exercise

Find at least two ways to determine the range of dates for records in the Repository.

Which is more efficient?

Questions?