Network Traffic Analysis - SiLK

Paul Krystosek, PhD
Matthew Heckathorn

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Notices

Copyright 2015 Carnegie Mellon University

This material is based upon work funded and supported by the 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.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon®, CERT®, CERT Coordination Center® and FloCon® are registered marks of Carnegie Mellon University.

DM-0003146
Housekeeping

Restrooms on past registration desk

Breaks and lunch in same location

Follow exit signs in case of emergency

Ask questions any time, don’t be shy
Course Objectives

At the end of this module, you will have the knowledge and skills needed to perform the following tasks:

• Name the major components of SiLK.
• Retrieve network flow records using the rwfilter command.
• Manipulate network flow records using basic SiLK commands.
• Analyze traffic and profile a network using basic SiLK commands.
Agenda

I. Network flow
   I. What is network flow
   II. Interpreting flow records
   III. SiLK commands

II. Basic SiLK tools
   I. SiLK Records, Files, and the Repository
   II. Analysis Tools and Categorization
   III. IP Sets
## Schedule

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 AM</td>
<td>SiLK Part 1 of 4</td>
<td>Basics of Network Flow and Unix Commands</td>
</tr>
<tr>
<td>10:45 AM</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>11:00 AM</td>
<td>SiLK Part 2 of 4</td>
<td>Basics of SiLK</td>
</tr>
<tr>
<td>12:30 PM</td>
<td>Lunch</td>
<td></td>
</tr>
<tr>
<td>1:30 PM</td>
<td>SiLK Part 3 of 4</td>
<td>Network flow analysis with SiLK</td>
</tr>
<tr>
<td>3:15 PM</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>3:30 PM</td>
<td>SiLK Part 4 of 4</td>
<td>More network flow analysis with SiLK</td>
</tr>
<tr>
<td>5:00 PM</td>
<td>Adjourn</td>
<td></td>
</tr>
<tr>
<td>6:00 PM</td>
<td>Welcome Reception</td>
<td>Near reception</td>
</tr>
</tbody>
</table>
Setting up your analysis environment

• SSH to flocon.cloudapp.net
  • Username: demo
  • Password: flocon2016
Analysis environment continued

ISLET: A Linux-based Software Training System

(T)olled, (S)calable, (L)ightweight (E)nvironment for (T)raining

Web: https://github.com/jonschipp/islet

... Press any key to continue or ^C to exit... 57
Analysis environment – Account creation

• Create your own application account
• Remember your information!
Analysis environment – Account information

• Accounts last for 14 days
  • The service will be shutdown on 1/25/2016
• You are limited to 2 GB of Hard drive space
  • Exceeding this limit will cause your account to be wiped
• Do not store anything of value on the server!
  • All information will be wiped
Analysis environment – SiLK Training Image
Analysis environment – At the Prompt

Welcome to SiLK Configuration!

SiLK, the System for Internet-Level Knowledge, is a collection of traffic analysis tools developed by the CERT Network Situational Awareness Team (CERT NetSA) to facilitate security analysis of large networks. The SiLK tool suite supports the efficient collection, storage, and analysis of network flow data, enabling network security analysts to rapidly query large historical traffic data sets.

Web: http://tools.netsa.cert.org

CERT
A place to try out SiLK

Enjoy yourself!
Run rwsiteinfo for information on repository
e.g. $ rwsiteinfo --fields=sensor,describe-sensor,type:list
demo@silk-live:~$
Exercise 0: *NIX

PS1='\W \!> '  # this is not permanent
export SILK_IPV6_POLICY=asv4
cd /data/bluered
ls -l silk.conf
less silk.conf  # type “q” to exit from less
cd
Part I: Network Flow
Part I Lessons: Network Flow

1. What is Network Flow?
2. Interpreting Flow Records
3. Issuing SiLK Commands
Lesson I.1
What is Network Flow?
Lesson I.1 Learning Objective

Given a sequence of packets and some basic knowledge of packets, the learner will be able to identify the uniflows comprising the packets.

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

<table>
<thead>
<tr>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>DATE</th>
<th>TIME</th>
<th>NUMBER</th>
<th>NAME</th>
<th>DURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>44.50880 N</td>
<td>73.18123 W</td>
<td>1/28/2008</td>
<td>60917</td>
<td>802-555-1234</td>
<td>Chittenden Bank</td>
<td>0:10:17</td>
</tr>
<tr>
<td>44.50880 N</td>
<td>73.18123 W</td>
<td>1/28/2008</td>
<td>6942</td>
<td>802-555-6673</td>
<td>Poopie Laff</td>
<td>0:00:03</td>
</tr>
<tr>
<td>44.50880 N</td>
<td>73.18123 W</td>
<td>1/28/2008</td>
<td>6945</td>
<td>802-555-9201</td>
<td>Harley Strappman</td>
<td>0:05:32</td>
</tr>
<tr>
<td>44.27834 N</td>
<td>73.21163 W</td>
<td>1/29/2008</td>
<td>1205</td>
<td>802-555-7756</td>
<td>Verizon Voice Mail</td>
<td>0:01:13</td>
</tr>
<tr>
<td>44.27834 N</td>
<td>73.21163 W</td>
<td>1/29/2008</td>
<td>1532</td>
<td>802-555-4492</td>
<td>Widgets LLC</td>
<td>0:02:47</td>
</tr>
<tr>
<td>44.27834 N</td>
<td>73.21163 W</td>
<td>1/29/2008</td>
<td>1209</td>
<td>802-655-7756</td>
<td>Verizon Voice Mail</td>
<td>0:00:36</td>
</tr>
<tr>
<td>44.50880 N</td>
<td>73.18123 W</td>
<td>1/30/2008</td>
<td>6830</td>
<td>802-555-1813</td>
<td>British Embassy</td>
<td>0:18:12</td>
</tr>
<tr>
<td>44.27834 N</td>
<td>73.21163 W</td>
<td>1/30/2008</td>
<td>1208</td>
<td>802-655-7756</td>
<td>Verizon Voice Mail</td>
<td>0:00:53</td>
</tr>
<tr>
<td>44.27834 N</td>
<td>73.21163 W</td>
<td>1/30/2008</td>
<td>2211</td>
<td>802-555-8673</td>
<td>Poopie Laff</td>
<td>0:06:16</td>
</tr>
<tr>
<td>44.50880 N</td>
<td>73.18123 W</td>
<td>1/31/2008</td>
<td>6903</td>
<td>802-555-1043</td>
<td>British Embassy</td>
<td>0:03:21</td>
</tr>
<tr>
<td>44.50880 N</td>
<td>73.18123 W</td>
<td>1/31/2008</td>
<td>6908</td>
<td>416-555-9834</td>
<td>British Embassy</td>
<td>0:12:20</td>
</tr>
<tr>
<td>44.149 N</td>
<td>73.03561 W</td>
<td>1/31/2008</td>
<td>1047</td>
<td>802-555-9201</td>
<td>Harley Strappman</td>
<td>0:04:02</td>
</tr>
<tr>
<td>44.149 N</td>
<td>73.03561 W</td>
<td>1/31/2008</td>
<td>1056</td>
<td>213-555-2761</td>
<td>M. Hendell</td>
<td>0:09:06</td>
</tr>
<tr>
<td>44.25235 N</td>
<td>73.50129 W</td>
<td>1/31/2008</td>
<td>1127</td>
<td>802-555-9201</td>
<td>Harley Strappman</td>
<td>0:05:30</td>
</tr>
</tbody>
</table>
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

Predictive Analysis

http://www.turkeydog.org/history.html
Did you ever wonder…

What’s on my network?
What happened before the event?
Where are policy violations occurring?
What are the most popular web servers?
By how much would volume be reduced with a blacklist?
Do my users browse to known infected web servers?
Do I have a spammer on my network?
When did my web server stop responding to queries?
Who uses my public servers?
Unidirectional Flows (Uniflows)
Packet Encapsulation

Ethernet frame

Dest MAC address
Source MAC address
Type of packet

IP datagram (packet)

Src IP address
Dst IP address
Type of segment

Transport segment

Src port
Dest port
Flags

Application layer message (HTTP, SMTP, DNS)
Two TCP/IP Sockets Make a Connection

TCP/IP SOCKET
IP address: 10.0.0.1
L4 protocol: TCP
Ephemeral port #

TCP/IP SOCKET
IP address: 203.0.113.1
L4 protocol: TCP
Well-Known Port #

Client Process

Connection

Server Process
Network Flow versus NetFlow

**Network Flow**—a generic term for the summarization of packets related to the same flow or connection into a single record

**NetFlow™**—a Cisco trademarked set of format specifications for storing network flow information in a digital record. Most common are versions 5 and 9.

**IPFIX**—a format specification from the IETF for flow records, an extension of Cisco NetFlow v9

**SiLK**—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
Sequence Diagram

DNS Client
192.168.1.105
UDP port 50744

DNS Server
10.1.10.1
UDP port 53

Request (type A)

Response (type A)
SiLK tool (rwcut) output

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>pro</th>
<th>packets</th>
<th>bytes</th>
<th>sensor</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.105</td>
<td>10.1.10.1</td>
<td>50744</td>
<td>53</td>
<td>17</td>
<td>1</td>
<td>64</td>
<td>S1</td>
<td>out</td>
</tr>
<tr>
<td>10.1.10.1</td>
<td>192.168.1.105</td>
<td>53</td>
<td>50744</td>
<td>17</td>
<td>1</td>
<td>80</td>
<td>S1</td>
<td>in</td>
</tr>
</tbody>
</table>
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.
Next Lesson

In lesson 1.2, you will learn to interpret SiLK flow records and understand the nature of the associated network activity.
Lesson I.2
Interpreting Flow Records
Lesson 1.2 Learning Objective

Given a series of uniflows and general knowledge of TCP/IP, the learner will be able to deduce and infer the nature of the network activity.
Network Monitoring

Internet

Other internetwork

sensor

sensor

sensor

sensor

SiLK repository

iSiLK™

SiLK Console

terminal

iSiLK is a trademark of Carnegie Mellon University
Realistic Sequence Diagram

DNS Client
192.168.1.105
UDP port 50744

Local Server
10.1.10.1

Sensor

Request (type A)
Dest port 53

Response (type A)
Src port 53

Root Server

.com Server

.mudynamics.com Server
What is this? — 1

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>pro</th>
<th>packets</th>
<th>flags</th>
<th>initF</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.105</td>
<td>10.1.10.1</td>
<td>50744</td>
<td>53</td>
<td>17</td>
<td>1</td>
<td></td>
<td></td>
<td>out</td>
</tr>
<tr>
<td>10.1.10.1</td>
<td>192.168.1.105</td>
<td>53</td>
<td>50744</td>
<td>17</td>
<td>1</td>
<td></td>
<td></td>
<td>in</td>
</tr>
<tr>
<td>192.168.1.105</td>
<td>198.51.100.6</td>
<td>49152</td>
<td>80</td>
<td>6</td>
<td>4</td>
<td>SRPA</td>
<td>S</td>
<td>outweb</td>
</tr>
<tr>
<td>198.51.100.6</td>
<td>192.168.1.105</td>
<td>80</td>
<td>49152</td>
<td>6</td>
<td>3</td>
<td>S PA</td>
<td>S A</td>
<td>inweb</td>
</tr>
</tbody>
</table>
HTTP Sequence Diagram

HTTP Client  192.168.1.105
HTTP Server   198.51.100.6
DNS Server    10.1.10.1

DNS Request (type A)
DNS Response (type A)

SYN
SYN, ACK
ACK
GET
ACK
Okay
RST
<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>proto</th>
<th>packets</th>
<th>bytes</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.22.105.250</td>
<td>71.55.40.253</td>
<td>52415</td>
<td>25</td>
<td>6</td>
<td>22</td>
<td>14045</td>
<td>FSRPA</td>
</tr>
<tr>
<td>71.55.40.253</td>
<td>30.22.105.250</td>
<td>25</td>
<td>52415</td>
<td>6</td>
<td>19</td>
<td>1283</td>
<td>FS PA</td>
</tr>
<tr>
<td>30.22.105.250</td>
<td>71.55.40.253</td>
<td>52415</td>
<td>25</td>
<td>6</td>
<td>1</td>
<td>40</td>
<td>R</td>
</tr>
</tbody>
</table>
### What Is This? — 3

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>pro</th>
<th>packets</th>
<th>bytes</th>
<th>sTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.89</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:04:30.172</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.149.249</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:04:37.302</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.52</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:04:37.312</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.127</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:04:58.363</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.196</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:04.327</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.149.30</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:09.242</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.149.173</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:12.174</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.13</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:14.114</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.56</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:15.383</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.114</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:18.228</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.202.92</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:22.466</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.202.68</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:23.497</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.24.161</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:30.256</td>
</tr>
<tr>
<td>99.217.139.155</td>
<td>177.252.202.238</td>
<td>1</td>
<td>2</td>
<td>122</td>
<td>2010/12/08T00:05:33.088</td>
</tr>
<tr>
<td>sIP</td>
<td>dIP</td>
<td>sPort</td>
<td>dPort</td>
<td>pkts</td>
<td>bytes</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>88.187.13.78</td>
<td>71.55.40.204</td>
<td>40936</td>
<td>80</td>
<td>83</td>
<td>3512</td>
</tr>
<tr>
<td>71.55.40.204</td>
<td>88.187.13.78</td>
<td>80</td>
<td>40936</td>
<td>84</td>
<td>104630</td>
</tr>
<tr>
<td>88.187.13.78</td>
<td>71.55.40.204</td>
<td>40938</td>
<td>80</td>
<td>120</td>
<td>4973</td>
</tr>
<tr>
<td>71.55.40.204</td>
<td>88.187.13.78</td>
<td>80</td>
<td>40938</td>
<td>123</td>
<td>155795</td>
</tr>
<tr>
<td>88.187.13.78</td>
<td>71.55.40.204</td>
<td>56172</td>
<td>80</td>
<td>84</td>
<td>3553</td>
</tr>
<tr>
<td>71.55.40.204</td>
<td>88.187.13.78</td>
<td>80</td>
<td>56172</td>
<td>83</td>
<td>103309</td>
</tr>
<tr>
<td>88.187.13.78</td>
<td>71.55.40.204</td>
<td>56177</td>
<td>80</td>
<td>123</td>
<td>5093</td>
</tr>
<tr>
<td>71.55.40.204</td>
<td>88.187.13.78</td>
<td>80</td>
<td>56177</td>
<td>124</td>
<td>157116</td>
</tr>
</tbody>
</table>
Standard SiLK Types

- **Outweb:** External to web
- **Outicmp:** External to icmp
- **Out:** General out
- **Inweb:** Internal to web
- **Inicmp:** Internal to icmp
- **In:** General in
- **Int2int:** Internal to internal
- **Ext2ext:** External to external
- **Outnull:** Out from null
- **Innull:** In to null
- **Null:** Null
- **Other*: To/from network that is neither internal nor external
More complete SiLK Types

Internal network

External network

Sensor

Null

outweb, outicmp, out, ipv6out

int2int, ipv6int2int

ext2ext, ipv6ext2ext

inweb, inicmp, in, ipv6in

outnull

innull
## SiLK Types in SiLK

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

Names in **bold** are often default types.
Lesson 1.2 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.
Next Lesson

In lesson I.3 we will learn how to issue *NIX commands, how to obtain online help for SiLK commands, and how to obtain information about the SiLK repository using a SiLK command.
Lesson I.3
Issuing SiLK Commands
Lesson I.3 Learning Objectives

• The learner will be able to issue simple SiLK commands correctly.

• The learner will be able to obtain online help text for SiLK commands and other *NIX commands.

• The learner will be able to obtain information about a SiLK sensor network and a SiLK flow-record repository.
*NIX commands

**System prompt**

Info + prompt character
e.g., ~ 101>

**User command**

command name  \texttt{rwfilter} (case sensitive)
options  \texttt{-h} \, \texttt{--help} \, \texttt{-k2} \, \texttt{--key=2}
arguments  \texttt{results.rw}
redirections  >  \,  >>  <  \,  <<
pipe  |  

For example:

\begin{verbatim}
rwcut \hspace{1em} --all-fields \texttt{results.rw} >\texttt{results.txt}
rwcut \hspace{1em} --fields=1-6 \texttt{results.rw} | more
\end{verbatim}
Some standard *NIX commands

ls – list name and attributes of files and directories

cd – change the current working directory

cat – output the contents of a file

head – output the first lines of a file

echo – output the argument

more and less – display a file one page at a time

cut – output only selected fields of a file

sort – reorder the records (lines) of a file

wc – word count (optionally, character and line count) of a file

exit – logout and terminate a terminal window
**NIX Standard Streams**

Standard In (stdin) – where normal (especially interactive) input comes from

Standard Out (stdout) – where normal/expected (especially interactive) output goes to

Standard Error (stderr) – where messages (especially unexpected) go to

**Defaults:**

stdin – keyboard

stdout – screen/window

stderr – screen/window

Defaults are overridden by redirections and pipes
Shell Scripts

Put a complicated command, pipeline, or sequence of pipelines into a script file.

- It saves your commands for reuse or learning.
- It eases making changes.

Create your script with `nano` or `vi (vim)`. Vi or vim can be found on every Linux/UNIX (*NIX) system.

Name your shell script something like `dothis.sh`

Add execute permission `chmod +x dothis.sh`

Execute (run) your script: `./dothis.sh`

SSH is the registered trademark of SSH Communications Security Corp
Exercise 1: Use a few relevant Linux commands

Create a new directory, change to it and use the `echo` command with

redirect “>”

and

append “>>”

to create a file.

Then examine it with `ls`, `cat` and `wc`

```
mkdir ex1
cd ex1
echo 10.1.60.25 > adr1.txt
echo 10.2.190.254 >> adr1.txt
ls adr1.txt
ls -l adr1.txt
cat adr1.txt
wc adr1.txt
cd
```
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

- tcpdump
- PCAP
- YAF

- Idle-timeout, Active-timeout
- Termination-attribute, Application, Start-time, Duration, Packets, Bytes, Flags...

- IPFIX
Analysis

SiLK repository

SiLK tool chain

Raw (binary) flow records in a file

Text

Raw (binary) flow records in a file
Reporting

UNIX text tools (sed, awk, ...)

Visualization tools (gnuplot, Rayon, Excel)

Text

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
```

Which record-types are defined?

rwsiteinfo --fields=type,mark-defaults

while this is true for most wines, there are always exceptions to the rule.

http://winefolly.com/review/types-dessert-wine/
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
Answers to questions you haven’t asked yet

At this point you probably have dozens of questions. Typical answers are:

• Yes, it does, and here is how to do it
• Yes, read about it in <reference>
• Yes, but it will take to long to describe right now
• Yes, but it is not a good idea because <some lame excuse>
• Because <long silence>
• No, it doesn’t because <really good reason>
• No, it doesn’t <long silence>
• No, but that’s a really good idea, please email it to me. Thanks!
Lesson I.3 Summary

We learned the parts of *NIX 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.
Next Lesson

In lesson II.1 we will learn how to choose just the flow records that are applicable to our inquiry.
Part II: Basic SiLK Tools
Part II Lessons: Network Flow

1. SiLK Records, Files, and the Repository
2. Analysis Tools and Categorization
3. IP Sets
Lesson II.1
SiLK Records, Files, and the Repository
Lesson II.1 Learning Objectives

• The learner will be able to display selected fields from a sequence of flow records.
• The learner will be able to determine which flow-record fields will be useful for a given analysis.
• The learner will be able to identify which rwfilter keywords are selection options.
• The learner will be able to pull flow records from a SiLK repository.
## Basic SiLK Tools: `rwcut`

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.

<table>
<thead>
<tr>
<th>sip</th>
<th>packets</th>
<th>type</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>dip</td>
<td>bytes</td>
<td>in</td>
<td>initialflags</td>
</tr>
<tr>
<td>sport</td>
<td>sensor</td>
<td>out</td>
<td>sessionflags</td>
</tr>
<tr>
<td>dport</td>
<td><code>scc</code></td>
<td><code>dur</code></td>
<td>application</td>
</tr>
<tr>
<td>protocol</td>
<td><code>dcc</code></td>
<td><code>stime</code></td>
<td>attributes</td>
</tr>
<tr>
<td>class</td>
<td><code>nhip</code></td>
<td><code>etime</code></td>
<td><code>itype &amp; icode</code></td>
</tr>
</tbody>
</table>

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

```bash
--all-fields  # way too much info
--fields=1-5,sTime  # just right
```
Create the ex3records.rw file

# rwfilter will not overwrite a file
rm ex3records.rw

rwfilter --type=all
  --sensor=S0
  --start=2009/04/20T11
  --proto=0-
  --pass=stdout
| rwsort --fields=stime
| rwfilter --input-pipe=stdin
  --max-pass=30
  --proto=0-
  --pass=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

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>pro</th>
<th>packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>80</td>
<td>50398</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>50189</td>
<td>5222</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>5222</td>
<td>50189</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>49592</td>
<td>443</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>443</td>
<td>49592</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>276</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>276</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>279</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>56515</td>
<td>53</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>53</td>
<td>56515</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>276</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>276</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.25</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>356</td>
</tr>
<tr>
<td>10.1.60.25</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>356</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.25</td>
<td>60515</td>
<td>25</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>1031</td>
<td>53</td>
<td>17</td>
<td>32</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>53</td>
<td>1031</td>
<td>17</td>
<td>32</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>53</td>
<td>3507</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>
# Exercise 3: What do the data look like?

```
rwcut ex3records.rw

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>pro</th>
<th>packets</th>
<th>bytes</th>
<th>flags</th>
<th>sTime</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td>5</td>
<td>383</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.439</td>
<td>0.006</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>80</td>
<td>50398</td>
<td>6</td>
<td>5</td>
<td>674</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.440</td>
<td>0.005</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>50189</td>
<td>5222</td>
<td>6</td>
<td>6</td>
<td>433</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.446</td>
<td>0.009</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>5222</td>
<td>50189</td>
<td>6</td>
<td>5</td>
<td>446</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.447</td>
<td>0.008</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>49592</td>
<td>443</td>
<td>6</td>
<td>10</td>
<td>1085</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.463</td>
<td>0.056</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>443</td>
<td>49592</td>
<td>6</td>
<td>10</td>
<td>4162</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.464</td>
<td>0.055</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>2048</td>
<td>0</td>
<td>1</td>
<td>276</td>
<td>23184</td>
<td></td>
<td>2009/04/20T11:35:19.490</td>
<td>1799.899</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>276</td>
<td>23184</td>
<td></td>
<td>2009/04/20T11:35:19.492</td>
<td>1799.899</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td>23436</td>
<td></td>
<td>2009/04/20T11:35:19.494</td>
<td>1799.900</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td>23436</td>
<td></td>
<td>2009/04/20T11:35:19.495</td>
<td>1799.900</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td>23436</td>
<td></td>
<td>2009/04/20T11:35:19.496</td>
<td>1799.900</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td>23436</td>
<td></td>
<td>2009/04/20T11:35:19.500</td>
<td>1799.900</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td>23436</td>
<td></td>
<td>2009/04/20T11:35:19.501</td>
<td>1799.900</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td>23436</td>
<td></td>
<td>2009/04/20T11:35:19.502</td>
<td>1799.900</td>
</tr>
</tbody>
</table>
```

---

"Carnegie Mellon University\"
Exercise 3: What do the data look like?

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

<table>
<thead>
<tr>
<th>SIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>pro</th>
<th>packets</th>
<th>flags</th>
<th>sTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td>5</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.439</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>80</td>
<td>50398</td>
<td>6</td>
<td>5</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.440</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>50189</td>
<td>5222</td>
<td>6</td>
<td>6</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.446</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>5222</td>
<td>50189</td>
<td>6</td>
<td>5</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.447</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>49592</td>
<td>443</td>
<td>6</td>
<td>10</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.463</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>443</td>
<td>49592</td>
<td>6</td>
<td>10</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.464</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>2048</td>
<td>1</td>
<td></td>
<td>276</td>
<td></td>
<td>2009/04/20T11:35:19.490</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>276</td>
<td></td>
<td>2009/04/20T11:35:19.491</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td></td>
<td>2009/04/20T11:35:19.494</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>279</td>
<td></td>
<td>2009/04/20T11:35:19.495</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.5</td>
<td>56515</td>
<td>53</td>
<td>17</td>
<td>1</td>
<td></td>
<td>2009/04/20T11:35:19.500</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>53</td>
<td>56515</td>
<td>17</td>
<td>1</td>
<td></td>
<td>2009/04/20T11:35:19.502</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>276</td>
<td></td>
<td>2009/04/20T11:35:19.514</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>276</td>
<td></td>
<td>2009/04/20T11:35:19.516</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.25</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>356</td>
<td></td>
<td>2009/04/20T11:35:19.528</td>
</tr>
<tr>
<td>10.1.60.25</td>
<td>10.1.60.203</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>356</td>
<td></td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.25</td>
<td>60515</td>
<td>25</td>
<td>6</td>
<td>4</td>
<td>S</td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>1031</td>
<td>53</td>
<td>17</td>
<td>32</td>
<td></td>
<td>2009/04/20T11:35:23.415</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>1031</td>
<td>17</td>
<td>32</td>
<td></td>
<td>2009/04/20T11:35:23.417</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>5307</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:23.443</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>3507</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:23.444</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:23.445</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>3508</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:23.446</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.084</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.085</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>3507</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:24.085</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>3508</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:24.086</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.632</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.632</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>3508</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:24.633</td>
</tr>
</tbody>
</table>
Exercise 3: What do the data look like?

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

<table>
<thead>
<tr>
<th>sip</th>
<th>dip</th>
<th>sport</th>
<th>dport</th>
<th>pro</th>
<th>iTy</th>
<th>iCo</th>
<th>packets</th>
<th>flags</th>
<th>sTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td></td>
<td></td>
<td>5</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.439</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td></td>
<td></td>
<td>5</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.440</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>50189</td>
<td>5222</td>
<td>6</td>
<td></td>
<td></td>
<td>6</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.446</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>5222</td>
<td>50189</td>
<td>6</td>
<td></td>
<td></td>
<td>5</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.447</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>49592</td>
<td>443</td>
<td>6</td>
<td></td>
<td></td>
<td>10</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.463</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>443</td>
<td>49592</td>
<td>6</td>
<td></td>
<td></td>
<td>10</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.464</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td>276</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.490</td>
</tr>
<tr>
<td>10.1.60.187</td>
<td>10.1.60.203</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td>276</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.491</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td>276</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.491</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td>276</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.491</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>0</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td>276</td>
<td>FS PA</td>
<td>2009/04/20T11:35:19.491</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>56515</td>
<td>53</td>
<td>17</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>2009/04/20T11:35:19.500</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.203</td>
<td>53</td>
<td>56515</td>
<td>17</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>2009/04/20T11:35:19.502</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.73</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td>276</td>
<td></td>
<td>2009/04/20T11:35:19.514</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>276</td>
<td></td>
<td>2009/04/20T11:35:19.516</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.25</td>
<td>2048</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td>356</td>
<td></td>
<td>2009/04/20T11:35:19.528</td>
</tr>
<tr>
<td>10.1.60.25</td>
<td>10.1.60.203</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>356</td>
<td></td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.25</td>
<td>60515</td>
<td>25</td>
<td>6</td>
<td></td>
<td></td>
<td>4</td>
<td>S</td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>1031</td>
<td>1031</td>
<td>17</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td>2009/04/20T11:35:23.415</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>53</td>
<td>1031</td>
<td>17</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td>2009/04/20T11:35:23.417</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:23.443</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:23.444</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:23.445</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:23.446</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.084</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.085</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:24.085</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:24.086</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.632</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>R A</td>
<td>2009/04/20T11:35:24.632</td>
</tr>
<tr>
<td>10.1.10.5</td>
<td>10.1.10.5</td>
<td>3507</td>
<td>3507</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>S</td>
<td>2009/04/20T11:35:24.632</td>
</tr>
</tbody>
</table>
```
Exercise 3: What do the data look like?

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

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>dPort</th>
<th>sPort</th>
<th>pro</th>
<th>packets</th>
<th>flags</th>
<th>initialF</th>
<th>sessionF</th>
<th>stime</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td>5</td>
<td>FS PA</td>
<td>S</td>
<td>F  PA</td>
<td>2009/04/20T11:35:19.439</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>50398</td>
<td>80</td>
<td>6</td>
<td>5</td>
<td>FS PA</td>
<td>S  A</td>
<td>F  PA</td>
<td>2009/04/20T11:35:19.440</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>10.1.60.203</td>
<td>5222</td>
<td>50189</td>
<td>6</td>
<td>6</td>
<td>FS PA</td>
<td>S</td>
<td>F  PA</td>
<td>2009/04/20T11:35:19.446</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.187</td>
<td>49592</td>
<td>443</td>
<td>6</td>
<td>10</td>
<td>FS PA</td>
<td>S</td>
<td>F  PA</td>
<td>2009/04/20T11:35:19.463</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>443</td>
<td>49592</td>
<td>6</td>
<td>10</td>
<td>FS PA</td>
<td>S  A</td>
<td>F  PA</td>
<td>2009/04/20T11:35:19.464</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>276</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.490</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.494</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.495</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.528</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.60.203</td>
<td>10.1.60.203</td>
<td>2048</td>
<td>1</td>
<td>356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2009/04/20T11:35:19.529</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>R A</td>
<td></td>
<td>2009/04/20T11:35:24.086</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3507</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>R A</td>
<td></td>
<td>2009/04/20T11:35:24.086</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>R A</td>
<td></td>
<td>2009/04/20T11:35:24.086</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>10.1.60.5</td>
<td>3508</td>
<td>53</td>
<td>6</td>
<td>1</td>
<td>R A</td>
<td>R A</td>
<td></td>
<td>2009/04/20T11:35:24.086</td>
</tr>
</tbody>
</table>
Exercise 3a: I wonder what a raw file looks like?

cd  # make home directory the working directory

rm -f ex3arecords.rw  # remove file; ok if not there

rwfilter --type=in \  
   --start-date=2009/4/20:14 --protocol=0- \  
   --compress=none \  
   --max-pass=1  \  
   --pass=ex3arecords.rw

ls -l ex3arecords.rw

rwfileinfo ex3arecords.rw

rwcut --fields=1-5,packets ex3arecords.rw

rwcut --all-fields ex3arecords.rw

hexdump -C ex3arecords.rw  # any readable text?
Exercise 3a Output

```
ex3a.sh
-rw-r--r-. 1 pnk pnk 264 Jan  7 21:10 ex3arecords.rw
rwfileinfo ex3arecords.rw

ex3arecords.rw:
  format(id)          FT_RWIPV6ROUTING(0x0c)
  version             16
  byte-order          littleEndian
  compression(id)     none(0)
  header-length       176
  record-length       88
  record-version      1
  silk-version        3.10.0
  count-records       1
  file-size           264

command-lines
  1  rwfilter --type=in --start-date=2009/4/20:11 --protocol=0- --compress=none --max-
      pass=1 --pass=ex3arecords.rw

rwcut --fields=1-5,packets ex3arecords.rw
  sIP|   dIP|sPort|dPort|pro|   packets|
  10.1.10.5|10.1.60.5| 3507|   53|   6|         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|
  10.1.10.5|10.1.60.5| 3507|   53|   6|         1|        48| S      |2009/04/20T11:35:23.443|   0.001|2009/04/20T11:35:23.444| S0|    0|    0|        0.0.0.0| S      |        |        |    0|all|
in|2009/04/20T11:35:23.443|2009/04/20T11:35:23.444| 0.001|   |
```
Exercise 3a Output

```
hexdump -C ex3arecords.rw  # any readable text?
00000000 de ad be ef 00 0c 10 00 00 2d ed d0 00 58 00 01 | ..........---X..|
00000010 00 00 00 02 00 00 00 76 72 77 66 69 6c 74 65 72|........vwrfilter|
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 30 39 2f 34 2f 32|rt-date=2009/4/2|
00000040 30 3a 31 31 20 2d 2d 70 72 6f 74 6f 63 6f 6c 74|0:11 --protocol=|
00000050 20 2d 20 2d 63 6f 6d 70 72 65 73 73 3d 6e 6f 20|0- --compress=no|
00000060 6e 65 20 2d 2d 6d 61 78 2d 70 61 73 73 3d 31 20|ne --max-pass=1 |
00000070 2d 2d 70 61 73 73 3d 65 78 33 61 72 65 63 6f 72|--pass=ex3arecor|
00000080 64 73 2e 72 77 00 00 00 00 00 00 00 00 00 00 2a |ds.rw........*..|
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.................|
* 00000b0 33 1e 4f c3 20 01 00 00 01 00 00 00 b3 0d 35 00 |3.O. .........5.|
00000c0 06 00 00 00 02 02 00 01 00 00 00 00 00 00 00 00 |.................|
00000d0 01 00 00 00 30 00 00 00 00 00 00 00 00 00 00 00 |.................|
00000e0 00 00 ff ff 0a 01 0a 05 00 00 00 00 00 00 00 00 00 |.................|
00000f0 00 00 ff ff 0a 01 3c 05 00 00 00 00 00 00 00 00 00 |.................|
0000100 00 00 ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 |.................|
```
Example Repository

RootDir

- sensor1
- sensor2
- silk.conf

- in
- inweb
- int2int
- out
- outweb
- ext2ext

- year
  - month
    - day
      - hour
        - sensor

Example Repository

Example Repository

flowtype-SENSOR_yyyymmdd.hh

e.g., in-SENS1_20091231.23
Basic SiLK Tools: **rwfilter**

- Pick files from the repository
- Plug in additional tools
- Direct flow output
- Advanced flow-by-flow filtering
- Compression
- Basic statistics

Swiss Army knife logo is a registered trademark of Victorinox AG
rwfilter Syntax

General form

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

Example call

```
rwfilter --sensor=S0 --type=in \ 
--start-date=2015/8/5T13 \ 
--end-date=2015/8/5T20 \ 
--protocol=0- --pass=workday-5.rw
```
Selection and Input Criteria

Selection options control access to repository files:

- `--start-date=2009/4/21`
- `--end-date=2009/4/21T03`
- `--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
Specified IP address or CIDR block

--saddress= --daddress= --any-address=
--not-saddress= --not-daddress= --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

--scidr= --dcidr= --any-cidr=
--not-scidr= --not-dcidr= --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

--sipset=  --dipset=  --anyset=
--not-sipset=  --not-dipset=  --not-anyset=

Specifies the name of a file storing the IP set:

--sipset=internalservers.set
--dipset=RussianBizNtwk.set
--anyset=TorNodes.set
--not-dipset=whitelist.set
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
rwfilter --sensor=S0 --type=in \ 
   --start=2009/4/21T00 --end=2009/4/21T07 \ 
   --daddress=10.1.0.0/16 --print-volume-stat

<table>
<thead>
<tr>
<th></th>
<th>Recs</th>
<th>Packets</th>
<th>Bytes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1436</td>
<td>2615</td>
<td>158084</td>
<td>8</td>
</tr>
<tr>
<td>Pass</td>
<td>1436</td>
<td>2615</td>
<td>158084</td>
<td></td>
</tr>
<tr>
<td>Fail</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 4: rwfilter

1) Find all traffic going outbound to external HTTPS servers on April 20, 2009. Save these flows in file https0420.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 April 20, 2009. Save these flows in file https0420.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 --sensor=S0 --type=outweb \ 
  --start=2009/4/20 --dport=443 \ 
  --pass=https0420.rw --print-volume-statistics

<table>
<thead>
<tr>
<th></th>
<th>Recs</th>
<th>Packets</th>
<th>Bytes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1308</td>
<td>37588</td>
<td>39354028</td>
<td>13</td>
</tr>
<tr>
<td>Pass</td>
<td>174</td>
<td>2413</td>
<td>223465</td>
<td></td>
</tr>
<tr>
<td>Fail</td>
<td>1134</td>
<td>35175</td>
<td>39130563</td>
<td></td>
</tr>
</tbody>
</table>

rwfileinfo https0420.rw --fields=count

https0420.rw:

  count-records   174
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 \n  --start-date=2010/12/08 \n  --type=outweb \n  --bytes=100000- \n  --pass=stdout \n| rwfilter \n  stdin \n  --duration=60- \n  --pass=long-http.rw \n  --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?
## Example Typos

<table>
<thead>
<tr>
<th>Example</th>
<th>Corrected</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--port=</code></td>
<td><strong>No such keywords</strong></td>
</tr>
<tr>
<td><code>--destport=</code></td>
<td></td>
</tr>
<tr>
<td><code>--sip= or --dip=</code></td>
<td></td>
</tr>
<tr>
<td><code>--saddress=danset.set</code></td>
<td><strong>Needs addr not filename</strong></td>
</tr>
<tr>
<td><code>--start-date=2006/06/12--end-date</code></td>
<td><strong>Space needed</strong></td>
</tr>
<tr>
<td><code>--start-date = 2006/06/12</code></td>
<td><strong>No spaces around equals</strong></td>
</tr>
<tr>
<td><code>start-date=2006/06/12</code></td>
<td><strong>Need dashes</strong></td>
</tr>
<tr>
<td><code>---start-date=2006/06/12</code></td>
<td><strong>Only two dashes</strong></td>
</tr>
<tr>
<td><code>--start-date=2005/11/04:06:00:00</code></td>
<td><strong>Only down to hour</strong></td>
</tr>
<tr>
<td><code>--end-date=2005/05/21:17:59:59</code></td>
<td></td>
</tr>
</tbody>
</table>
SiLK Commandments

1. Thou shalt use Sets instead of using several rwfilter commands to pull data for multiple IP addresses.

2. Thou shalt store intermediate data on local disks, not network disks.

3. Thou shalt 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. Thou shalt work in binary for as long as possible. ASCII representations are much larger and slower than the binary representations of SiLK data.

5. Thou shalt filter no more than a week of traffic at a time. The filter runs for excessive length of time otherwise.

6. Thou shalt only run a few rwfilter commands at once.

7. Thou shalt specify the type of traffic to filter. Defaults work in mysterious ways.

8. Thou shalt appropriately label all output.

9. Thou shalt 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.
Next Lesson

In lesson II.2 we will learn to reduce large numbers of flow records to meaningful information and statistics.
Lesson II.2
Analysis Tools and Categorization
Lesson II.2 Learning Objectives

• The learner will be able to create a time series of given flow records.

• The learner will be able to determine all the different values of a given field for given flow records and determine the traffic volumes for those field values.

• The learner will be able to display the top/bottom $n$ values of a given field as measured by some measure of volume.
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

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

Value from flow record
e.g., packets

Bin key field
e.g., protocol

Aggregate Value

Packet count

Packet count

Packet count

Packet count

Packet count

Bins

Total packets

Total packets

Total packets

TCP

UDP

ICMP
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
  
  \texttt{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

http://www.cs.uoregon.edu/research/tau/docs/paraprof/ch05s02.html
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.
\textbf{rwcount}

The bin key is always time. You choose the period. The aggregate measures are chosen for you. They are flows (records), bytes, packets.

\begin{verbatim}
rwfilter --sensor=S0 --start=2009/4/21 \  --type=in --proto=1 --pass=stdout \  | rwcount --bin-size=3600
\end{verbatim}

<table>
<thead>
<tr>
<th>Date</th>
<th>Records</th>
<th>Bytes</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009/04/21T13:00:00</td>
<td>10.00</td>
<td>2460.00</td>
<td>41.00</td>
</tr>
<tr>
<td>2009/04/21T14:00:00</td>
<td>29.00</td>
<td>8036.00</td>
<td>107.00</td>
</tr>
<tr>
<td>2009/04/21T15:00:00</td>
<td>22.00</td>
<td>2214.00</td>
<td>47.00</td>
</tr>
<tr>
<td>2009/04/21T16:00:00</td>
<td>10.00</td>
<td>1586.00</td>
<td>23.00</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### What Is This? — 9

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Records</th>
<th>Bytes</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010/12/08T00:00:00</td>
<td>1351571.66</td>
<td>73807086.40</td>
<td>1606313.61</td>
</tr>
<tr>
<td>2010/12/08T01:00:00</td>
<td>1002012.43</td>
<td>54451440.59</td>
<td>1185143.62</td>
</tr>
<tr>
<td>2010/12/08T02:00:00</td>
<td>1402404.61</td>
<td>77691865.26</td>
<td>1675282.27</td>
</tr>
<tr>
<td>2010/12/08T03:00:00</td>
<td>1259973.65</td>
<td>68575249.90</td>
<td>1491393.08</td>
</tr>
<tr>
<td>2010/12/08T04:00:00</td>
<td>939313.56</td>
<td>51410968.24</td>
<td>1118584.81</td>
</tr>
<tr>
<td>2010/12/08T05:00:00</td>
<td>459564.75</td>
<td>80862273.32</td>
<td>1742058.62</td>
</tr>
<tr>
<td>2010/12/08T06:00:00</td>
<td>1280651.23</td>
<td>69881126.41</td>
<td>1519435.24</td>
</tr>
</tbody>
</table>
```

...
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 --sensor=S0 --type=out,outweb \ 
    --start=2009/04/21 --end=2009/04/23 \ 
    --proto=0- --pass=stdout \ |
| rwcount --bin-size=$((24*60*60))
```
Exercise 5: \texttt{rwcount}

Produce a time-series with 30-minute intervals, analyzing incoming ICMP traffic collected at sensor S0 on April 20, 2009.
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 --sensor=S0 --type=in \  
   --start=2009/4/20 --protocol=1 --pass=stdout \  
| rwcount --bin-size=1800

<table>
<thead>
<tr>
<th>Date</th>
<th>Records</th>
<th>Bytes</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009/04/20T13:30:00</td>
<td>5.05</td>
<td>1588.92</td>
<td>26.48</td>
</tr>
<tr>
<td>2009/04/20T14:00:00</td>
<td>21.92</td>
<td>5480.87</td>
<td>91.35</td>
</tr>
<tr>
<td>2009/04/20T14:30:00</td>
<td>8.03</td>
<td>3610.21</td>
<td>60.17</td>
</tr>
<tr>
<td>2009/04/20T15:00:00</td>
<td>14.58</td>
<td>5432.54</td>
<td>90.54</td>
</tr>
<tr>
<td>2009/04/20T15:30:00</td>
<td>17.33</td>
<td>6519.74</td>
<td>108.66</td>
</tr>
<tr>
<td>2009/04/20T16:00:00</td>
<td>13.69</td>
<td>5702.65</td>
<td>95.04</td>
</tr>
<tr>
<td>2009/04/20T16:30:00</td>
<td>12.89</td>
<td>5105.11</td>
<td>85.09</td>
</tr>
<tr>
<td>2009/04/20T17:00:00</td>
<td>11.50</td>
<td>5135.57</td>
<td>85.59</td>
</tr>
<tr>
<td>2009/04/20T17:30:00</td>
<td>7.00</td>
<td>2704.40</td>
<td>45.07</td>
</tr>
</tbody>
</table>
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)
```bash
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
```

<table>
<thead>
<tr>
<th>dIP</th>
<th>sPort</th>
<th>Bytes</th>
<th>Packets</th>
<th>Records</th>
<th>sTime-Earliest</th>
<th>eTime-Latest</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.113.41.190</td>
<td>80</td>
<td>12782</td>
<td>20</td>
<td>4</td>
<td>2010/12/08T18:42:51</td>
<td>2010/12/08T18:58:49</td>
</tr>
<tr>
<td>30.182.228.143</td>
<td>80</td>
<td>203907933</td>
<td>143611</td>
<td>2</td>
<td>2010/12/08T18:53:59</td>
<td>2010/12/08T19:01:47</td>
</tr>
<tr>
<td>82.180.203.87</td>
<td>80</td>
<td>213013145</td>
<td>150896</td>
<td>92</td>
<td>2010/12/08T18:06:36</td>
<td>2010/12/08T18:32:33</td>
</tr>
<tr>
<td>82.180.203.197</td>
<td>80</td>
<td>800</td>
<td>8</td>
<td>2</td>
<td>2010/12/08T18:43:30</td>
<td>2010/12/08T18:43:30</td>
</tr>
<tr>
<td>88.124.166.233</td>
<td>80</td>
<td>223930369</td>
<td>158276</td>
<td>97</td>
<td>2010/12/08T18:08:55</td>
<td>2010/12/08T18:32:25</td>
</tr>
<tr>
<td>88.124.166.233</td>
<td>443</td>
<td>509285</td>
<td>732</td>
<td>43</td>
<td>2010/12/08T18:06:57</td>
<td>2010/12/08T18:51:11</td>
</tr>
<tr>
<td>94.239.226.247</td>
<td>80</td>
<td>124833037</td>
<td>96047</td>
<td>3</td>
<td>2010/12/08T18:25:22</td>
<td>2010/12/08T18:21:34</td>
</tr>
<tr>
<td>109.95.61.80</td>
<td>80</td>
<td>8467397</td>
<td>6325</td>
<td>90</td>
<td>2010/12/08T18:08:59</td>
<td>2010/12/08T18:10:09</td>
</tr>
<tr>
<td>139.65.186.4</td>
<td>80</td>
<td>204123360</td>
<td>143794</td>
<td>3</td>
<td>2010/12/08T18:19:48</td>
<td>2010/12/08T18:26:36</td>
</tr>
<tr>
<td>139.177.10.136</td>
<td>80</td>
<td>407978375</td>
<td>287354</td>
<td>6</td>
<td>2010/12/08T18:20:03</td>
<td>2010/12/08T19:01:30</td>
</tr>
<tr>
<td>219.149.72.154</td>
<td>1024</td>
<td>44</td>
<td>1</td>
<td>1</td>
<td>2010/12/08T18:50:40</td>
<td>2010/12/08T18:50:40</td>
</tr>
<tr>
<td>249.216.88.172</td>
<td>80</td>
<td>88</td>
<td>2</td>
<td>2</td>
<td>2010/12/08T18:44:42</td>
<td>2010/12/08T18:44:47</td>
</tr>
<tr>
<td>250.211.100.88</td>
<td>80</td>
<td>3295160</td>
<td>2492</td>
<td>42</td>
<td>2010/12/08T18:47:50</td>
<td>2010/12/08T18:58:53</td>
</tr>
</tbody>
</table>
rwuniq outtraffic.rw --fields=dip \
--values=sip-distinct,records,bytes --sip-distinct=400- \
--sort-output

<table>
<thead>
<tr>
<th>dIP</th>
<th>sIP-Distinct</th>
<th>Bytes</th>
<th>Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.220.28.183</td>
<td>512</td>
<td>20480</td>
<td>512</td>
</tr>
<tr>
<td>171.128.2.27</td>
<td>448</td>
<td>19069280</td>
<td>476732</td>
</tr>
<tr>
<td>171.128.2.179</td>
<td>448</td>
<td>139501200</td>
<td>3487530</td>
</tr>
<tr>
<td>171.128.212.14</td>
<td>448</td>
<td>139467440</td>
<td>3486686</td>
</tr>
<tr>
<td>171.128.212.124</td>
<td>448</td>
<td>127664480</td>
<td>3191612</td>
</tr>
<tr>
<td>171.128.212.127</td>
<td>448</td>
<td>66611560</td>
<td>1665289</td>
</tr>
<tr>
<td>171.128.212.188</td>
<td>448</td>
<td>139467680</td>
<td>3486692</td>
</tr>
<tr>
<td>171.128.212.228</td>
<td>448</td>
<td>139393160</td>
<td>3484829</td>
</tr>
<tr>
<td>245.225.153.120</td>
<td>763</td>
<td>30520</td>
<td>763</td>
</tr>
<tr>
<td>245.238.193.102</td>
<td>1339</td>
<td>179480</td>
<td>4487</td>
</tr>
</tbody>
</table>
Exercise 6: rwuniq

For outgoing flows from S0 on 2009/04/20, 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 2009/04/20, write and execute the `rwf` 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 2009/04/20, 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=2009/4/20
    --proto=6
    --pass=stdout
| rwuniq --fields=packets --sort-output
```

<table>
<thead>
<tr>
<th>packets</th>
<th>Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2573</td>
</tr>
<tr>
<td>2</td>
<td>129</td>
</tr>
<tr>
<td>3</td>
<td>133</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>271</td>
</tr>
<tr>
<td>6</td>
<td>289</td>
</tr>
<tr>
<td>7</td>
<td>182</td>
</tr>
<tr>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td>9</td>
<td>61</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
</tr>
</tbody>
</table>

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
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`  
```bash
--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”)
**What Is This? – 12**

```bash
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

<table>
<thead>
<tr>
<th>sIP</th>
<th>Bytes</th>
<th>%Bytes</th>
<th>cumul_%</th>
</tr>
</thead>
<tbody>
<tr>
<td>71.55.40.62</td>
<td>1754767148</td>
<td>41.541935</td>
<td>41.541935</td>
</tr>
<tr>
<td>71.55.40.169</td>
<td>1192063164</td>
<td>28.220617</td>
<td>69.762552</td>
</tr>
<tr>
<td>71.55.40.179</td>
<td>331310772</td>
<td>7.843372</td>
<td>77.605923</td>
</tr>
<tr>
<td>71.55.40.204</td>
<td>170966278</td>
<td>4.047415</td>
<td>81.653338</td>
</tr>
<tr>
<td>177.249.19.217</td>
<td>122975880</td>
<td>2.911301</td>
<td>84.564639</td>
</tr>
<tr>
<td>71.55.40.72</td>
<td>110726717</td>
<td>2.621318</td>
<td>87.185957</td>
</tr>
<tr>
<td>71.55.40.200</td>
<td>101593627</td>
<td>2.405103</td>
<td>89.591060</td>
</tr>
<tr>
<td>177.71.129.255</td>
<td>40166574</td>
<td>0.950894</td>
<td>90.541954</td>
</tr>
<tr>
<td>71.55.40.91</td>
<td>35316554</td>
<td>0.836076</td>
<td>91.378030</td>
</tr>
<tr>
<td>149.249.114.204</td>
<td>26634602</td>
<td>0.630541</td>
<td>92.008571</td>
</tr>
</tbody>
</table>
Exercise 7: rwstats

What are the top 10 incoming protocols on April 20, 2009, collected on sensor S0?
Exercise 7: rwstats

What are the top 10 incoming protocols on April 20, 2009, collected on sensor S0?

HINT

Incoming flows have type in or inweb
### Exercise 7: `rwstats` solution

```
rwfilter --sensor=S0 --type=in,inweb \
   --start=2009/4/20 --prot=0- --pass=stdout \n| rwstats --fields=protocol --value=records --count=10
```

**INPUT:** 5512 Records for 3 Bins and 5512 Total Records

**OUTPUT:** Top 10 Bins by Records

<table>
<thead>
<tr>
<th>pro</th>
<th>Records</th>
<th>%Records</th>
<th>cumul_%</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4476</td>
<td>81.204644</td>
<td>81.204644</td>
</tr>
<tr>
<td>17</td>
<td>896</td>
<td>16.255443</td>
<td>97.460087</td>
</tr>
<tr>
<td>1</td>
<td>140</td>
<td>2.539913</td>
<td>100.000000</td>
</tr>
</tbody>
</table>
Exercise 8: rwstats

Top 9 inside hosts according to how many outside hosts they communicate with on April 20, 2009, collected on sensor S0?
Exercise 8: rwstats

Top 9 inside hosts according to how many outside hosts they communicate with on April 20, 2009, collected on sensor S0?

**HINT**

Use

```
--value=distinct:dip
```
**Exercise 8: rwstats solution**

```
rwfilter --sensor=S0 --type=out,outweb \ 
        --proto=0- --start-date=2009/4/20 --pass=stdout \ 
| rwstats --fields=sip --value=distinct:dip --count=9
```

**INPUT:** 5001 Records for 14 Bins  
**OUTPUT:** Top 9 Bins by dIP-Distinct

<table>
<thead>
<tr>
<th>sip</th>
<th>dIP-Distinct</th>
<th>%dIP-Distinct</th>
<th>cumul_%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.60.187</td>
<td>17</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.5</td>
<td>11</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.25</td>
<td>11</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.191</td>
<td>9</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.73</td>
<td>5</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.253</td>
<td>3</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.251</td>
<td>3</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>212.117.116.35</td>
<td>3</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>10.1.60.4</td>
<td>2</td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>

--no-percents will clean up the question marks.
## rwuniq vs. rwstats - 1

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

**rwuniq**
- Scripts for producing rwuniq and rwstats in top/bottom mode.

**both**
- Scripts for producing rwuniq and rwstats.

**rwstats in top/bottom mode**
- Scripts for producing rwuniq and rwstats in top/bottom mode.
<table>
<thead>
<tr>
<th>rwuniq</th>
<th>both</th>
<th>rwstats in top/bottom mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>--all-counts</strong> (bytes, pkts, flows, earliest sTime, and latest eTime)</td>
<td>Show volume aggregate value[s].</td>
<td><strong>--no-percents</strong> (good when primary aggregate isn’t Bytes, Packets, or Records)</td>
</tr>
<tr>
<td><strong>--bin-time</strong></td>
<td>to adjust sTime and eTime</td>
<td></td>
</tr>
<tr>
<td><strong>--presorted-input</strong> (omit when value includes Distinct fields, even if input is sorted)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>--values</strong>= sTime-Earliest, eTime-Latest</td>
<td><strong>--values</strong>=Records, Packets, Bytes, sIP-Distinct, dIP-Distinct, Distinct:KEY-FIELD (KEY-FIELD can’t also be key field in --fields)</td>
<td></td>
</tr>
</tbody>
</table>
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
IP Sets
Lesson II.3 Learning Objectives

• Given a collection of IP addresses and CIDR blocks, the learner will be able to create an IP Set SiLK-file.

• Given an IP Set, the learner will be able to display the contents and characteristics of the set.

• Given an IP Set, the learner will be able to partition flow records based on the presence/absence of IP addresses in the set.

• Given a sequence of flow records, the learner will be able to extract IP addresses from the records and create an IP Set.
Blacklists, Whitelists, Books of Lists...

Too many addresses for the command line?
• spam block list
• malicious websites
• arbitrary list of any type of addresses

Create an IP set!
• From individual IP address in dotted decimal or integer format
• From CIDR blocks, e.g., 192.168.0.0/16
• From flow records

Use it directly within your rwfilter commands.
• --sipset, --dipset, --anyset
• --not-sipset, --not-dipset, --not-anyset
Set Tools

**rwsetbuild**: Create a set from text.

**rwsetcat**: Display an IP set as text.

**rwset**: Create sets from binary flow records.

**rwsetmember**: Test if an address is in given IP sets.

**rwsettool**: Perform set algebra (intersection, union, set difference) on multiple IP sets.
Creating a Set from a text file

Start with a text file containing IP addresses
IPv4 in dotted quad notation
IPV6 in canonical format (e.g. 2001:db8::f00)

Run `rwsetbuild` to make the conversion from text to set

```bash
$ cat sample.set.txt
192.168.1.1
172.16.0.1
10.1.2.3
$

$ rwsetbuild sample.set.txt sample.set
$ ls -l sample*
-rw-r--r--. 1 pnk pnk 124 Jan 7 17:22 sample.set
-rw-r--r--. 1 pnk pnk  32 Jan 7 17:21 sample.set.txt
$ rwsetcat sample.set
10.1.2.3
172.16.0.1
192.168.1.1
$
Exercise 9: Create a set file

In Exercise 1 you created the text file `adr1.txt`

It should contain two IPv4 addresses in dotted quad notation

Create a set file from it
Exercise 9: Create a set file

In Exercise 1 you created the text file `adr1.txt`
It should contain two IPv4 addresses in dotted quad notation
Create a set file from it

HINT

Use

```
Rwsetbuild <text file> <set file>
```
Exercise 9: Create a set file

In Exercise 1 you created the text file `adr1.txt`. It should contain two IPv4 addresses in dotted quad notation. Create a set file from it.

**HINT**

If you run it twice, `rwsetbuild` will not overwrite the set file. You’ll have to delete it first.
Create a Set of IP CIDR Blocks

$ cp ~rbandes/public/private_example.set.txt .  # copy file

$ cat private_example.set.txt  # display file
10.0.0.0/8      # RFC 1918 private
172.16.0.0/12   # RFC 1918 private
192.0.2.0/24    # documentation (example.com or example.net)
192.168.0.0/16  # RFC 1918 private
198.51.100.0/24 # documentation (example.com or example.net)
203.0.113.0/24  # documentation (example.com or example.net)

$ rwsetbuild private_example.set.txt private_example.set

$ rwsetcat private_example.set | head -n 5
10.0.0.0
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4

$ rwsetcat --count-ips private_example.set
17892096
Use IP Set as Partitioning Criterion

$ rwfilter --type=in,inweb --start=2009/4/20 --end=2009/4/24 \ 
   --sipset=private_example.set --print-volume-statistics

<table>
<thead>
<tr>
<th>Recs</th>
<th>Packets</th>
<th>Bytes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>2563253</td>
<td>9609775</td>
<td>5501740288</td>
</tr>
<tr>
<td>Pass</td>
<td>2557016</td>
<td>9603538</td>
<td>5501284187</td>
</tr>
<tr>
<td>Fail</td>
<td>6237</td>
<td>6237</td>
<td>456101</td>
</tr>
</tbody>
</table>
Find Addresses from Traffic NOT in the IP Set

```bash
$ rwfilter --type=in,inweb --start=2009/4/20 --end=2009/4/24 \ 
   --not-sipset=private_example.set --pass=stdout \ 
   | rwset --sip-file=outside_not_private.set

$ rwsetcat --count-ips outside_not_private.set
6237
```
Examine the IP Set

$ rwsetcat outside_not_private.set | less

$ rwsetcat --cidr-blocks outside_not_private.set | less

$ rwsetcat --network-structure=8 outside_not_private.set
   100.0.0.0/8 | 6237

$ rwsetcat --network-structure=16 outside_not_private.set \ 
   | wc -l
   2

$ rwsetcat --network-structure=16 outside_not_private.set
   100.1.0.0/16 | 5932
   100.2.0.0/16 | 305

$ rwsetcat --network-structure=24 outside_not_private.set \ 
   | wc -l
   264
Exercise 10 Sets

1) For April 21, 2009 on sensor S0, make a set-file of addresses of all actual inside hosts.

   Should we examine incoming or outgoing traffic?

2) Make a set-file of all outside addresses.

   Can you make both sets with one command?
Exercise 10 Sets

1) For April 21, 2009 on sensor S0, make a set-file of addresses of all actual inside hosts.

Should we examine incoming or outgoing traffic?

2) Make a set-file of all outside addresses.

Can you make both sets with one command?

HINT

Pipe rwfilter to rwset
Set Exercise 10 solution

```
rwfilter --sensor=S0 --type=out,outweb \
    --start-date=2009/4/21 \
    --proto=0- --pass=stdout \
| rwset --sip-file=insidehosts.set \
    --dip-file=outsidehosts.set
```
Exercise 11 Sets

Examine the two set-files from Exercise 9.
Exercise 11 Sets

Examine the two set-files from Exercise 9.

HINT

How big are the set files?
What can you say about the files?
How many addresses in each set?
What are they?
Set Exercise 11 solution

```
ls -l insidehosts.set
rwfileinfo insidehosts.set
rwsetcat insidehosts.set --count
rwsetcat insidehosts.set | less

ls -l outsidehosts.set
rwfileinfo outsidehosts.set
rwsetcat outsidehosts.set --count
rwsetcat outsidehosts.set | less
```
Exercise 12 Sets

Which /16 networks are on the inside?
Which /8 networks are on the outside?

Bonus question
How many /24 networks are on the outside?
Exercise 12 Sets

Which /16 networks are on the inside?
Which /8 networks are on the outside?

Bonus question

How many /24 networks are on the outside?

HINT

Use --network-struc=N
Exercise 12 Sets

Which /16 networks are on the inside?
Which /8 networks are on the outside?

Bonus question
How many /24 networks are on the outside?

HINT
Use --network-struc=N
Where N comes from CIDR notation /N
Exercise 12 solution

```
rwsetcat --network-struc=16 insidehosts.set
rwsetcat --network-struc=8 outsidehosts.set
```

**Bonus question**

```
rwsetcat --network-struc=24 outsidehosts.set \ \
  | wc -l
```
Set-like Files: Bags

Wouldn’t it be nice to count something per address and associate the two?

Yes, it would, it exists and it is called a Bag

- rwbag
- rwbagbuild
- rwbagcat
- rwbagtool
Bag Example

rm -f sf.bag
rwfilter --type=out,outweb
    --sensor=S0
    --start=2009/4/20
    --proto=0-
    --pass=stdout
| rwbag --sip-flows=sf.bag

$ rwbagcat sf.bag
    10.1.60.4| 20|
    10.1.60.5| 3155|
    10.1.60.25| 182|
    10.1.60.53| 1|
    10.1.60.73| 171|
    10.1.60.74| 1|
    10.1.60.153| 11|
    10.1.60.187| 1045|
    10.1.60.191| 250|
    10.1.60.251| 115|
    10.1.60.253| 12|
    212.117.116.35| 8|
    212.117.116.36| 28|
    212.117.116.38| 2|
Set-like Files: 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`
Pmap Example

```
rwfilter --sensor=S0 \ 
    --start=2009/4/21 \ 
    --proto=0- \ 
    --pass=stdout \ 
| rwuniq --pmap-file=pname:/data/bluered/protocols.pmap \ 
    --fields=src-pname,proto \ 
    --values=bytes --sort-out
```

<table>
<thead>
<tr>
<th>src-pname</th>
<th>pro</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>1</td>
<td>17228</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>53954032</td>
</tr>
<tr>
<td>UDP</td>
<td>17</td>
<td>1175172</td>
</tr>
</tbody>
</table>
Set-like Files: Tuple

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 (use it instead of, say, proto=6 dport=25,58,143,158,209,366,465,587)

```
rwfilter ... -tuple-file=email-ports.txt ...  
```
## Comparison of IP Set, Bag, Tuple, PMap

<table>
<thead>
<tr>
<th></th>
<th>IP Set</th>
<th>Bag</th>
<th>Tuple</th>
<th>PMap</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Semantics</strong></td>
<td>presence</td>
<td>volume</td>
<td>conditionals</td>
<td>categories</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>1</td>
<td>2</td>
<td>1–5</td>
<td>2</td>
</tr>
<tr>
<td><strong>1st Column</strong></td>
<td>IP Addr</td>
<td>various</td>
<td>Flow-Label Field</td>
<td>IP Addr or Proto/Port</td>
</tr>
<tr>
<td><strong>2nd Column</strong></td>
<td>—</td>
<td>measure</td>
<td>Flow-Label Field or none</td>
<td>Label</td>
</tr>
<tr>
<td><strong>Used for Partitioning</strong></td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Used for Field Output</strong></td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Binary/Text</strong></td>
<td>binary</td>
<td>binary</td>
<td>text</td>
<td>binary</td>
</tr>
<tr>
<td><strong>Combine</strong></td>
<td>set algebra</td>
<td>arithmetic</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Usual Role</strong></td>
<td>input, interim, output</td>
<td>interim, output</td>
<td>input, interim</td>
<td>input</td>
</tr>
</tbody>
</table>
An IP Set is a collection of IP addresses. There are no duplicates in a set. An IP address is either in a given set or it is not.

rwsetbuild creates an IP Set from a text file. rwset creates an IP Set from flow records.

IP sets can be used for partitioning flow records with rwfilter.

rwsetcat displays the contents or summaries of an IP Set.
Part II Summary—Basic SiLK

- rwsiteinfo
- rwcut
- rwfILTER
- rwcount
- rwstats
- rwuniq
- rwsetbuild
- rwsetcat
- rwset
- rwsetmember
A look ahead

There is more to the SiLK Analysis Tool Suite than the above

- TCP Flags
- Application Label
- PySiLK
- Plug-ins
- Rayon

The Analyst’s Handbook is a great resource for learning more


As is the report: Network Profiling Using Flow

http://www.sei.cmu.edu/reports/12tr006.pdf
Resources

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

<table>
<thead>
<tr>
<th>SiLK</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Analysis Suite - overview of the analysis tools</td>
</tr>
<tr>
<td></td>
<td>- Filtering, displaying, and sorting</td>
</tr>
<tr>
<td></td>
<td>- SiLK Python extension (PySiLK)</td>
</tr>
<tr>
<td></td>
<td>- Counting, grouping, and mating</td>
</tr>
<tr>
<td></td>
<td>- IPset, Bag, and Prefix Map manipulation</td>
</tr>
<tr>
<td></td>
<td>- IP and port labeling files</td>
</tr>
<tr>
<td></td>
<td>- Run-time plug-ins</td>
</tr>
<tr>
<td></td>
<td>- Packet and IPFIX processing</td>
</tr>
<tr>
<td></td>
<td>- Scan detection</td>
</tr>
<tr>
<td></td>
<td>- Flow file utilities</td>
</tr>
<tr>
<td></td>
<td>- Utilities</td>
</tr>
<tr>
<td></td>
<td>- Packing System - summary of the SiLK packing tools</td>
</tr>
<tr>
<td></td>
<td>- Configuration and Overview - brief description of SiLK's configuration files</td>
</tr>
<tr>
<td></td>
<td>- Analysis Handbooks and References - documents and additional references describing the analysis tools</td>
</tr>
<tr>
<td></td>
<td>- Installation Information - information for installing and configuring SiLK</td>
</tr>
<tr>
<td></td>
<td>- Alphabetized index of Manuals - links to all of the SiLK manual pages</td>
</tr>
</tbody>
</table>

Analysis Suite

The SiLK analysis suite is a collection of command-line tools for processing SiLK Flow records created by the SiLK packing system. These tools read binary files containing SiLK Flow records and partition, sort, and count these records. The most important analysis tool is `nfilter`, an application for querying the central data repository for SiLK Flow records that satisfy a set of filtering options. The tools are intended to be combined in various ways to perform an analysis task. A typical analysis uses UNIX pipes and intermediate data files to share data between invocations of the tools.

The tools and plug-in modules that make up the analysis tools are listed below, roughly grouped by functionality.

**Filtering, displaying, and sorting**

- `nfilter`: Select SiLK Flow records from the data repository and partition the records into one or more ‘pass’ and/or ‘fail’ output streams.
- `rasut`: Print the attributes of SiLK Flow records in a delimited, columnar, human-readable format. Users can define new printable attributes using plug-ins written in C or PySiLK.
- `nresort`: Sort SiLK Flow records using a user-specified key comprised of record attributes, and write the records to the named output path or to the standard output. Users can define new key fields using plug-ins written in C or PySiLK.

**SiLK Python Extension (PySiLK)**

- `PySiLK, SiLK in`: Read, manipulate, and write SiLK Flow records, IPsets, and Bags from within Python.
Resources
http://tools.netsa.cert.org/
Questions?
Contact Information

Paul Krystosek — pnk@cert.org
Matt Heckathorn — maheckathorn@cert.org

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA
Extra slides
How well does compression work?

FILE=compress-none.rw
for RECS in 1 2 3 4 5 6 7 8 9 10 11
do
    rm -r $FILE
    rwfilter --type=all \ 
        --start-date=2009/4/20:11 \ 
        --protocol=0- \ 
        --compress=none \ 
        --max-pass=$RECS \ 
        --pass=$FILE
    ls -l $FILE
done
No compression vs compression

-rw-r--r-. 1 pnk pnk 264 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 352 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 440 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 528 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 616 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 704 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 792 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 880 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 968 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 1056 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 1144 Jan 5 22:02 compress-none.rw
-rw-r--r-. 1 pnk pnk 256 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 272 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 282 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 290 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 298 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 303 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 324 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 333 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 344 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 355 Jan 5 22:02 compress-best.rw
-rw-r--r-. 1 pnk pnk 364 Jan 5 22:02 compress-best.rw
Visualization
Flow Visualization

Visualization has many uses

• Analysis
• Explanation
• Discovery

One of the best results of visualization is to speed up whatever you are doing
Popular types of visualization:

• Bar Chart
• Time Series
• Scatter Plot
• Histogram
• Link diagram (directed graph)
• Heat Map
• Other
  - Timelines
  - Geographic maps
    - Pie charts
Software to do visualization: Rayon

- Rayon was written to work and play well with SiLK and Python
- It fits in with the Unix pipe mode of scripting
- It doesn’t (yet?) handle everything we want to do

<table>
<thead>
<tr>
<th>Viz Type</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar Chart</td>
<td>rycategories</td>
</tr>
<tr>
<td>Time Series</td>
<td>rytimeseries</td>
</tr>
<tr>
<td>Scatter Plot</td>
<td>ryscatterplot</td>
</tr>
<tr>
<td>Histogram</td>
<td>rycategories</td>
</tr>
<tr>
<td>Link diagram</td>
<td>GraphViz</td>
</tr>
<tr>
<td>Heat Map</td>
<td>ryhilbert</td>
</tr>
</tbody>
</table>
What does our data look like?

Volume of Flow Types

Number of Flows:
- 100,000,000
- 80,000,000
- 60,000,000
- 40,000,000
- 20,000,000
- 0

Flow Types:
- in
- out
- icmp
- udp

ICMP Type and Code Clustering

ICMP Code:
- 255

ICMP Type:
- 0
- 1
Let's take a closer look 2009/04/20
Lets take a closer look 2009/04/21
Let's take a closer look 2009/04/22
Let's take a closer look 2009/04/23
Let's take a closer look 2009/04/24

![Network Traffic Analysis - SiLK](image-url)