CORSARO

Alistair King, Alberto Dainotti
alistair@caida.org, alberto@caida.org
CAIDA
MOTIVATIONS
(for the scientists)

• Several researchers have used the UCSD Network Telescope

• Patchwork of tools and ad-hoc scripts

• All analysis has been with ‘roll your own’ code

• All results have been in ‘proprietary’ formats and locations

• There is no unified framework for analyzing darknet data
MOTIVATIONS
(for the people who pay the bills)

• Desperate times call for desperate measures

• For a decade CAIDA has enjoyed free (and virtually unlimited) archival of scientific data

• No Longer!

• We had >100 TiB of gzip pcap data from 2003-2011 stored on SDSC’s tape archive
9 YEARS OF DATA

Hour of Day (UTC)

Date


% of Median (4.42 GiB) Bytes/hr

0  50  100  150  200

Wednesday, August 1, 12
STORAGE

(the cold hard facts)

- 120 TiB > 2,809,506,377,709 pkts > 37,742 hourly pcap files
- Most files are on tape
- And, it just keeps on coming!
  - ~3 TiB per month
- We can’t afford to store the existing data, let-alone keep up with the new data
TUG O’WAR

(the balancing act)
TUG O’WAR
(the balancing act)
TUG O’WAR
(the balancing act)
TUG O’WAR
(the balancing act)

We asked several scientists what was required if we must aggregate data...

packet counts?

full pcap
TUG O’WAR
(the balancing act)

We asked several scientists what was required if we must aggregate data...

packet counts?

full pcap
TUG O’WAR
(the balancing act)

We asked several scientists what was required if we must aggregate data...
TUG O’WAR

(the balancing act)

We asked several scientists what was required if we must aggregate data...
TUG O’WAR

*(the balancing act)*

We asked several scientists what was required if we **must aggregate data**...
TUG O’WAR
(the balancing act)

We asked several scientists what was required if we must aggregate data...
TUG O’WAR
(the balancing act)

We asked several scientists what was required if we must aggregate data...

packet counts?

full pcap
STRATIFIED STORAGE
(the PG-13 version)
STRATIFIED STORAGE

*(the PG-13 version)*

2003  time  now
STRATIFIED STORAGE

*(the PG-I3 version)*

level of 'information loss'

2003

0

1

2

3

4

time

now

Wednesday, August 1, 12
STRATIFIED STORAGE

(the PG-13 version)

level of 'information loss'

Most Recent 2 Months Raw Pcap
STRATIFIED STORAGE

(the PG-l 3 version)

Most Recent 2 Months Raw Pcap

Quarterly & Monthly Raw Pcap Samples

level of 'information loss'

0

2003

now

time

Wednesday, August 1, 12
STRAIFIED STORAGE
(the PG-13 version)

Most Recent 2 Months Raw Pcap
Quarterly & Monthly Raw Pcap Samples
Published Datasets

level of 'information loss'

2003 time now

0 1 2 3 4
STRATIFIED STORAGE

(the PG-13 version)

Most Recent 2 Months Raw Pcap

Quarterly & Monthly Raw Pcap Samples

Published Datasets

Potentially Interesting Events

level of 'information loss'

0

1

2

3

4

2003

now

time

Wednesday, August 1, 12
STRATIFIED STORAGE

(the PG-13 version)

Most Recent 2 Months Raw Pcap
Quarterly & Monthly Raw Pcap Samples
Published Datasets
Potentially Interesting Events
Aggregated Data

level of 'information loss'

2003

Wednesday, August 1, 12
STRATIFIED STORAGE

*(the PG-13 version)*

- Most Recent 2 Months Raw Pcap
- Quarterly & Monthly Raw Pcap Samples
- Published Datasets
- Potentially Interesting Events
- Aggregated Data

Time line:
- 2003
- Now

Level of 'information loss':
- 0
- 1
- 2
- 3
- 4

Wednesday, August 1, 12
IL CORSARO

• We need a tool that can...

  • **Do Good Things** with every packet
  • Help **Minimize Storage** Costs
  • Do it very **Efficiently**
  • Be **Easy and Useful** for researchers to extend
KEY GOALS

• Compression
• Speed
• Easily Usable
• Portable
• Extensible
• Reliable
A PICTURE

**tools**
- cor2ascii
- cor-agg
- corsaro
- [your name here]

**libtrace**
- libcorsaro

- intervals
- I/O
- logging

- initialize
- process packet
- interval start
- interval end
- finalize

**plugins**
- Flow-Tuple
- DoS
- [yours!]

- .log
- .cor
- .pcap
- .ft.cor
- .dos.cor
- [.you.cor]
LINEAGE

(and not reinventing the wheel)

- framework.c
  - A proof-of-concept darkcap analysis engine by Alberto Dainotti

- libtrace
  - Library for trace processing by WAND group
  - Multi-threaded, actively developed/supported

- libwandio
  - Library for threaded, compressed file IO.
  - Bundled with libtrace (since 3.0.14)
COMPRESSION

- **Aggregates** data into **intervals**.
  - Trade-off time resolution for reduction of redundant data.
- **Highly optimized binary output**.
  - Carefully sorted to exploit characteristics of **gzip**
- Provides transparent **output compression** to plugins.
  - Both **bzip** and **gzip** supported.
SPEED
(and efficiency)
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware

  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:
SPEED
(and efficiency)

• *Libtrace* is designed for speed (zero copy, caching, etc)

• All IO is threaded to take advantage of modern hardware
  
  • E.g. Corsaro with *bzip* runs as fast as when it uses *gzip*

• Minimize rework by plugins:

• We have identified three main types of plugin:
  • General purpose aggregation.
  • Specialized Analysis.
  • “I need to know x right now”
THE FLOW-TUPLE
(the penicillin of aggregated data)

• A **general purpose aggregation** plugin for Corsaro.
• The **Flow-Tuple satisfies** several **common analysis needs**
• Features:
  • Source IP, Dest IP, Source Port, Dest Port, Protocol, TCP Flags, TTL, IP Length
  • Per-interval key/value pair: 
    - **key** => EightTuple
    - **value** => Packet Count *(for the interval)*
  • Also keyed on the packet classification *(e.g. backscatter)*
• >80% **compression** from .pcap.gz using 1 minute aggregation intervals
PUTTING IT TO USE
\cite{eight-tuple}

- **Flow-Tuple data** and **Corsaro** heavily used for analysis in two recent IMC papers:

  - “Analysis of a ‘/0’ Stealth Scan from a Botnet” - A. Dainotti et al.
  - “Entropy-based Classification of IP Darkspace Events” - T. Zseby et al.

![Graph showing observed UDP packet rate per hour vs. entropy](image)

**Figure 4: Entropy during TCP Probe**
SPECIALIZED ANALYSIS
(for that special code in your life)

• Corsaro supports **highly-specialized analysis** plugins

• **Existing code** that does **something complicated** can **leverage Corsaro’s features**

• As an example, we ported our **new_rsdos** tool:

  • DoS detection algorithm
  
  • Optimized for speed and output compression

  • Identifies potential “Attack Vectors” and records statistics about the attack

  • Preserves the ‘initial’ packet for later inspection

AD-HOC ANALYSIS
(aggregate research)

- Parsing tcpdump ASCII output is slow and error prone
- Corsaro makes it quick and easy to add a new plugin
- E.g. we wanted to know # packets and # unique source IPs, that are not part of a DoS attack, in an hour:
  - In < 1 hour, we had a plugin - it runs fast
  - For free we got:
    - DoS identification by a prior plugin (chained results)
    - Threaded I/O
    - Output is compressed
    - Adaptable interval lengths (e.g. we now want daily counts)
CORSARO IN ACTION

(getting it done)

• Corsaro has been in active use at CAIDA since Feb 2012
  • FreeBSD, Linux, Mac OSX, Solaris X

• Combined Corsaro and Marinda
  (http://www.caida.org/projects/ark/)

• Used an ad-hoc cluster to process 100 TiB data in down to 15 TiB

• Has been run with over 30,000 hours of pcap
WHERE ARE WE GOING?

• Beta release **this month**

• Extend Corsaro to provide **realtime** packet capture, analysis and archival of darkspace data.

• **Geolocation** and **AS-mapping** plugins for populating packet meta-data

• Realtime **reporting** and **visualization**

• Data sharing

• Efficient Indexing for fast searches

• IPv6
ACK & QUESTIONS

(we would love some suggestions)

• **Dan Andersen** - for tirelessly maintaining and provisioning CAIDA machines well beyond their intended purposes.

• **Emile Aben** - for relentlessly pursuing Good Science

• **Tanja Zseby** - for valuable input along the way, and for being an eager (and sometimes unfortunate) pre-alpha user.

• **NERSC** - for agreeing to archive every pcap file at the last minute.

• **SDSC** - for being patient while we moved, providing compute resources, and for storing all that data for all those years.