



# CERT

## **Implementing Coarse, Long-Term Traffic Capture**

Michael Collins, CERT/Network  
Situational Awareness

© 2005 Carnegie Mellon University

 Software Engineering Institute

# Outline of Talk

---

Introduction To Work

Logistics of Traffic Analysis

Implementing Traffic Capture

- Relational vs. Flatfile
- Implementing Robust Service

Analytical Tools

Future Goals

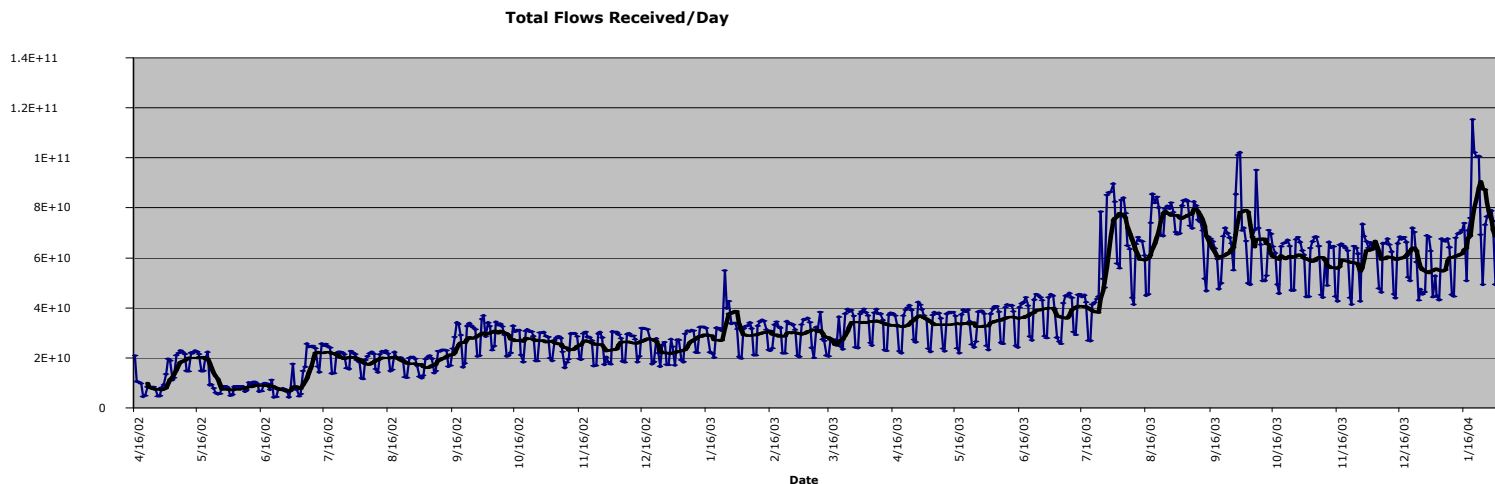
- Reaching the limits of flow
- Enhancing Netflow

# The Work

Originally wanted raw data for security modeling

- Developed techniques for rapidly querying data
- It kind of snowballed from there...

The resulting system is now used by 50+ users, capture 250+GB of traffic/day and is used operationally 24/7



# How well are we doing?

---

We receive  $n$  events per second

Analysts can process/tag/understand  $k$  events per second.  $k = n$  good;  $k \ll n$  normal

- Unfettered Analysis time is *insanely valuable*
- Requirements change as the network changes

Increase  $k$  by:

1. Reducing access time
2. Reducing the amount of lookup/doublechecking done by analysts
3. Classifying, discarding events
4. Making more inclusive events

# Reducing Access Time

---

Large mechanical, governed by fixed rules

- The smaller the record, the better
- The more informative the data in that space, the better

Start by using netflow, then enhancing the netflow format

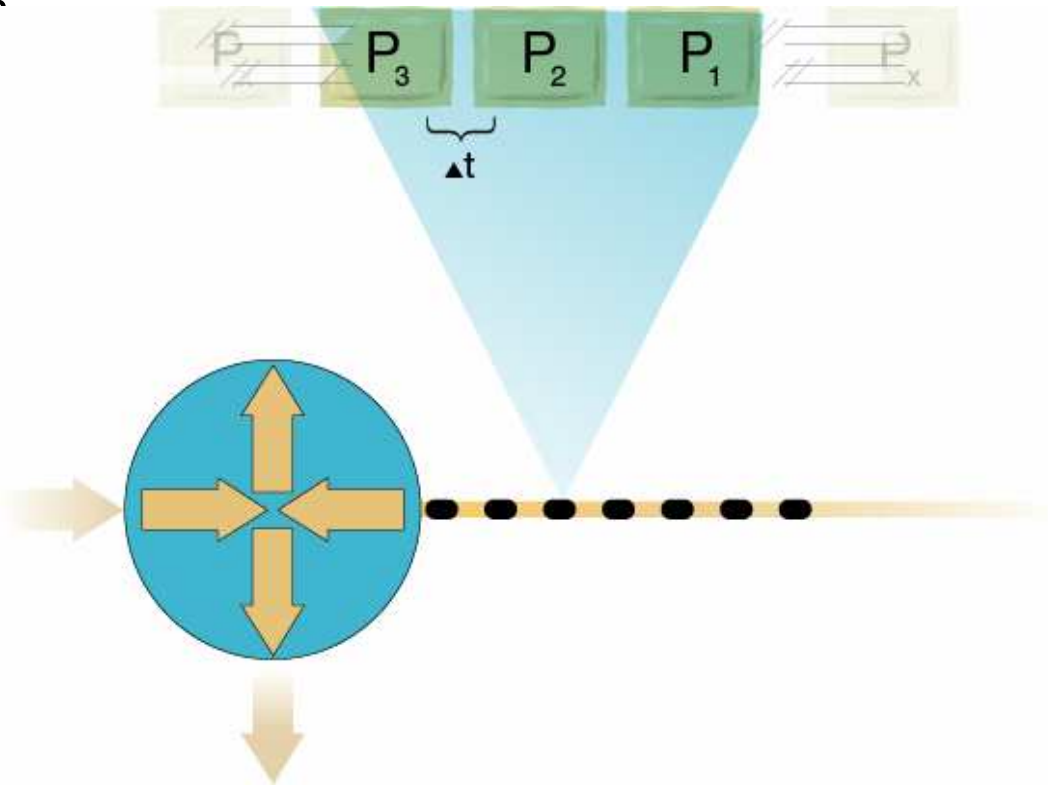
# Source Data

Use flow data: a flow is a summary of packet data between two sites

No payload information

Flows are generated by sensors throughout the network

Flows are logistically manageable: GB vs TB



# NetFlow Data

<b>saddress</b>	source IP		<b>daddress</b>	dest IP
<b>sport</b>	source port		<b>dport</b>	dest port
<b>protocol</b>	IP protocol		<b>packets</b>	# of packets
<b>bytes</b>	# of bytes		<b>flags</b>	TCP Flags
<b>stime</b>	first pkt. time		<b>duration</b>	time taken
<b>etime</b>	last pkt. time		<b>sensor</b>	sensor

Most security relevant information from standard flow record

Routing information is maintained through categories

# Why Flow?

---

Payload: out of the question, too large

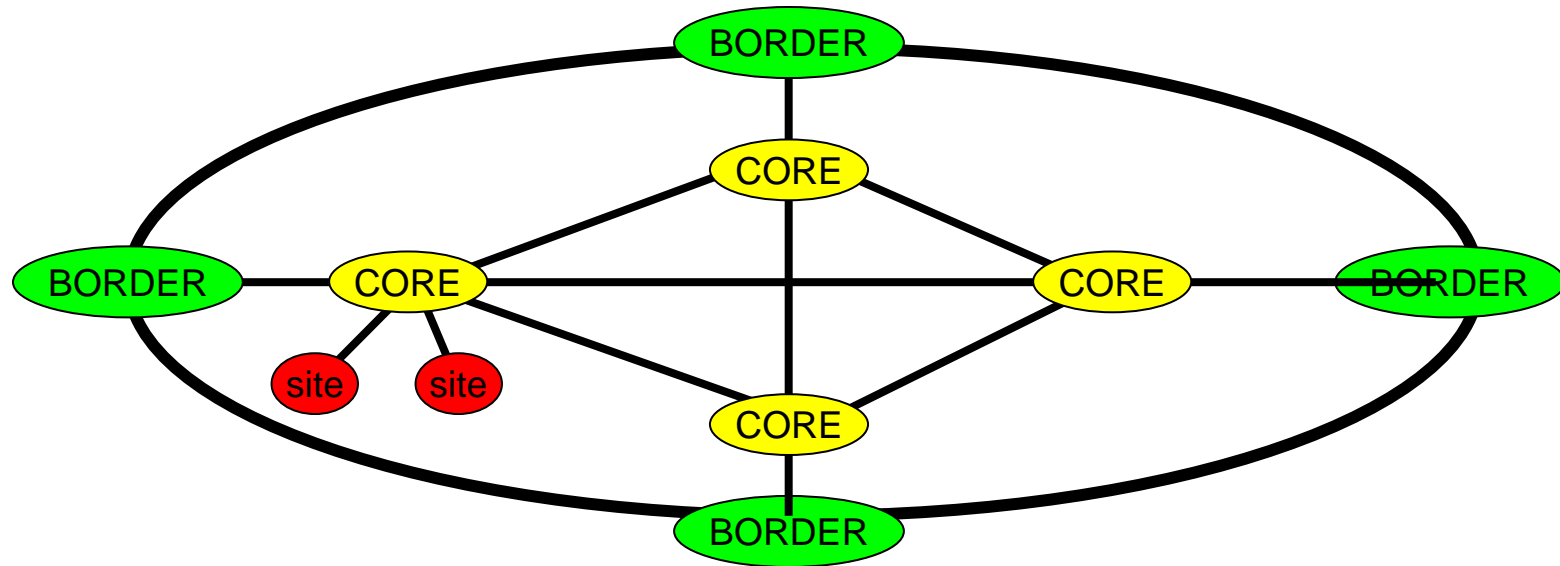
Limited payload: what to keep?

Ultimately, we are governed by what we can store on disk.

- If we had to pick 48 (or, in our case, 22) bytes of information, this is the highest-value 22 we could pick from a group of records

Not complete - flow is intended for traffic analysis, not security. More on this later

# Network Architecture



Network is an outer ring with an inner core. Network itself crosses the globe.

Routing is asymmetrical - chance of {A,B} packet using the same router as {B,A} packet is low. Sensor Abstraction

# Traffic Categorization

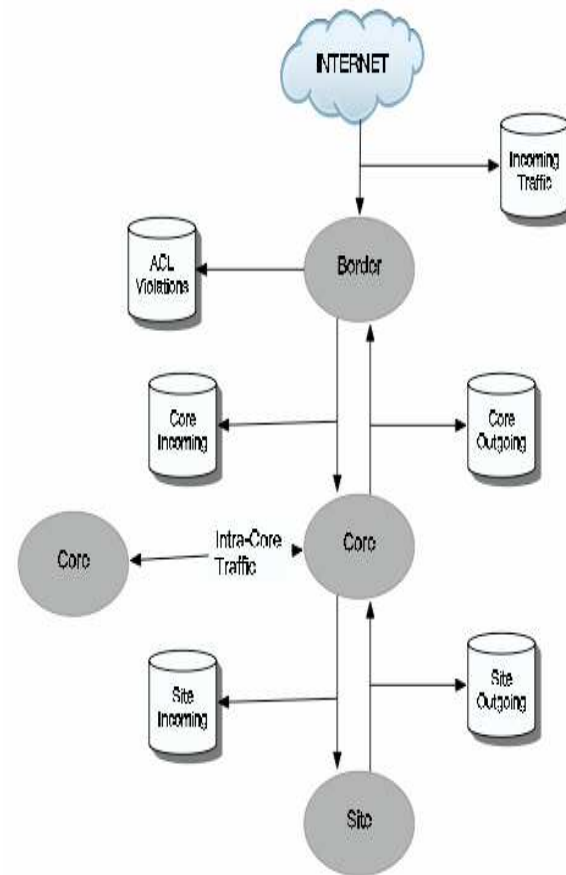
Routing information is used to categorize traffic:

- Incoming, Outgoing
- Aimed at router, null

Interface info is otherwise discarded

- Categories also ensure if you query one class/type combination you don't see duplicates

Special case: internally routed traffic

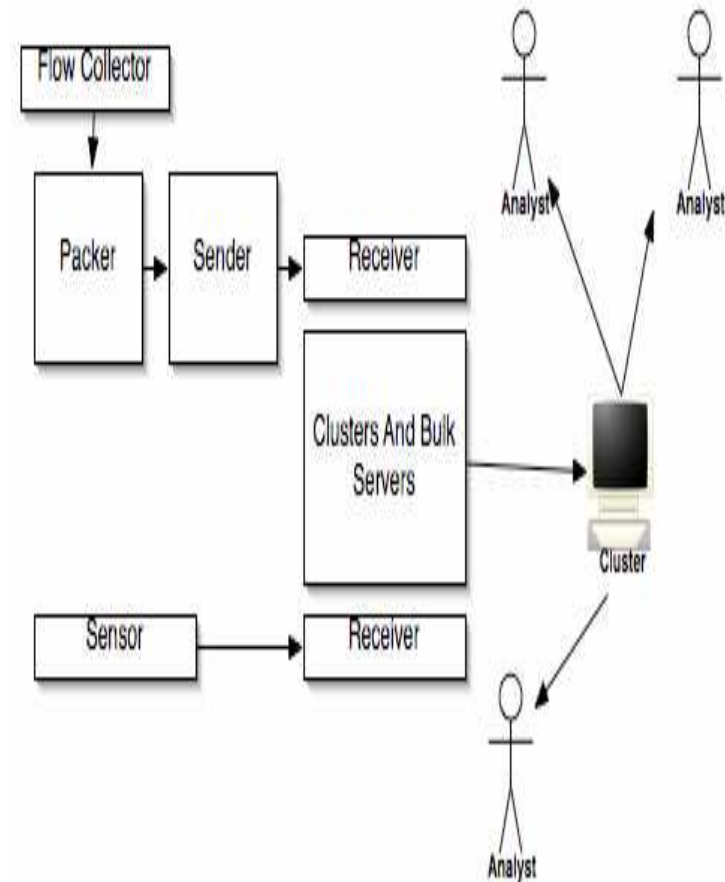


# Collection System Architecture

Collection System consists of distributed sensors writing to a common parallel cluster

Sensors write common format, optimized for specific cases

Data is stored in flat files with fixed formats on disk



# Why Flat Files?

---

Databases provide a lot of functionality for maintaining system integrity

- Locking
- Rollback, Time Travel

SiLK is write once, read many

- No updates, only inserts
- Each sensor writes to its own files

Interface is through a common library

- Formats don't change much, generally in response to our suggestions
  - Optimization, security enhancements
- Plan for an IPFIX update eventually

# The Impact of Global Warming On Computer Security

---

Reliability is handled through multiply redundant storage

- Sensors have buffer space, as does packer
- Multiple ring structure means that a flow will be captured by a sensor
- Assumed a weekend of 3x normal traffic (Code Red)

Sensors have a “trickle” effect, where data is shipped in priority form if there’s an interrupt.

- Exploit business cycle

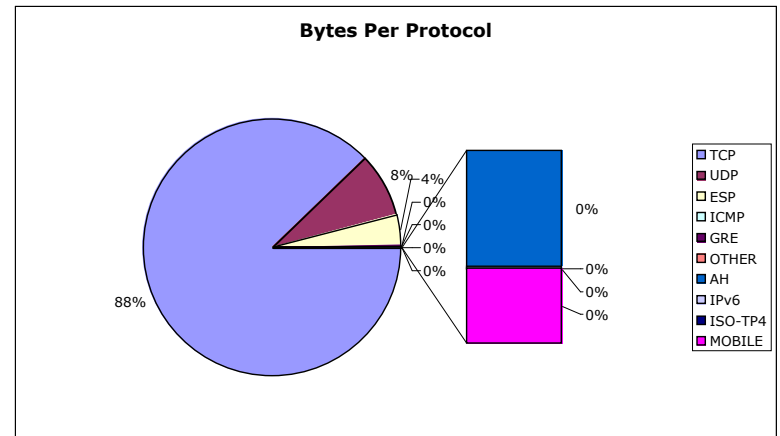
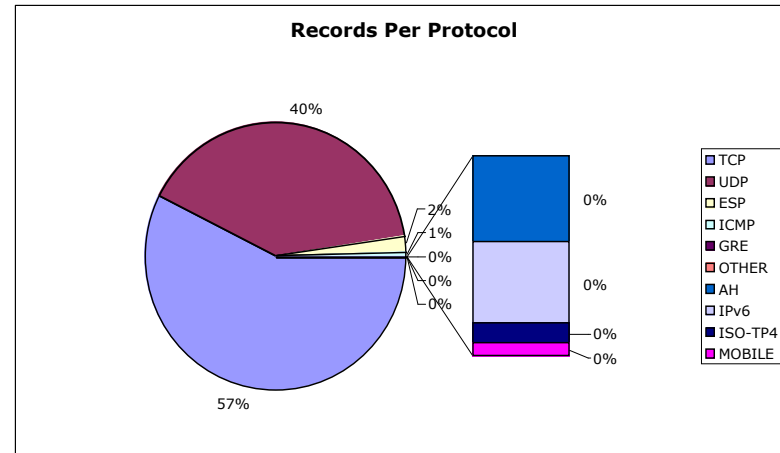
That said, don’t build your datacenter in a hurricane zone if you can avoid it

# Basic Statistics

On a *per-flow* basis, the majority of traffic is aberrant

- Short sessions
- ACL violations

The majority of traffic is TCP, then UDP, then either ICMP or ESP depending on metric used.



# Analytical Tools

---

Security analysis is log analysis, *perl* is the most basic security tool

- Perl tends to be text heavy, I/O bound
- Our data is highly structured

Use the SiLK toolset instead

- Binary applications, optimized in C that provide rapid analysis facilities

# SiLK features

---

Works in binary data

- ASCII at the very last step - usually 6-8x larger than binary per record

Tool Categories

- Basic query/filter tools (per record selection)
- UNIX replacements (sort, uniq)
- Data structure tools (arbitrary sets of IP addresses, groups of flows)
- Decision/mapping tools (scan detection, service mapping)

General goal is to write high-speed applications in C, then stitch together with scripts to write arbitrary detectors

# Future Directions

---

Netflow: still some unexplored domains

- Mapping
- Using per-IP/AS information

Otherwise, the goal is to expand flow and add properties

# Passive Mapping

---

Track network features using flow

- Simple features - router/server presence and configuration
- More advanced: communications networks (ie, bittorrent, email)

Continuously audit the network to figure out how it is configured

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

# Expanding Sensors

---

## Replace netflow

- Put format under our control
- Include security specific data

## First version: flocaps

- Converted UDP netflow broadcasts into tcp signals in compressed format
- Priority transfer

## Next:

- Expand for security

# AMP

---

Developed by clients: DAG Card + TCPDump with flow output

Next questions: What to store

- Expanded time (ms)
- Four byte hash of payload/ICMP message
- Initial flags

# Eventual system

---

## Analyst's Desktop

- Integrate multiple data sources - realtime responses, alerts, archival data, maps

## Heterogenous data sources

- Some sources are more useful “zoomed in”
  - BGP, DNS? (Chosen for criticality)
  - ICMP, IGP? (Chosen for information)
- Maintain a continuous buffer of data?

# Conclusions

---

Introduction to architectural basics

An installation is available at  
<http://silktools.sourceforge.net>

- Provides core packing system and analytical tools

Find out more at <http://www.cert.org/netsa>

CERT runs a regular Netflow workshop; for more info see <http://www.cert.org/flocon>