Software Engineering Institute

FloCon 2013 Proceedings

January 2013

CERT Program

http://www.sei.cmu.edu

Carnegie Mellon

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.

DM-0000129 (v-2.0-20130111-0818)

Thinking Security

Steven M. Bellovin
Federal Trade Commission/
Columbia University

Security's Progress

1. There is good research on a new defense
2. Using this defense becomes a recognized "best practice"
3. It is inscribed on assorted auditors' checklists
4. A change in technology or the threat model renders it all but useless
5. It stays on the checklists… (Do you still shred your old punch cards and paper tapes?)

Technology Changes

- Single-job batch systems
- Multi-user timesharing systems
  - Mainframes; Unix; “superminis”
- Stand-alone microcomputers
  - DOS (no OS protection)
- Dial-up PCs
- Networked PCs running full-blown OSes
- Smartphones, tablets, etc
- The “Internet of things”?

I’ve used all except, perhaps, the last…

Threat Model Changes

- Joy hackers
  - "Pursuit of knowledge"
  - Manual hacking, often via stepping stones
  - Annoying viruses and worms
  - Random spread; most did little damage
- The spammer/hacker alliance
  - Worms that don't shut down the Internet; bots as payloads
- Cyberespionage
- Cyberattacks (Stuxnet, Flame, Shamoon)
- "Preparing the battlefield"?

Security Advice

- Pick strong passwords
- Use a firewall
- Run current antivirus software
- Stay up to date on patches

Security Advice

- Pick strong passwords
  - *The Morris-Thompson paper is from 1979, an era of electromechanical terminals and few logins*
- Use a firewall
  - *Smartphones, tablets, and laptops move around*
- Run current antivirus software
  - *It's increasingly ineffective*
- Stay up to date on patches
  - *What about 0-day attacks?*

Passwords (1979)

- Password strength rationale is from the days of electromechanical terminals
- No local computational capability
- No keystroke loggers or user malware
- Moore's Law change since 1978: about 4,000,000× improvement

(Picture courtesy Perry Metzger)

Passwords

- Old scenario: hacker steals hashed system password file from timesharing machine
- New scenarios:
  - Hacker steals application—not system—password file from web server
  - May be plaintext, for password recovery
  - Secondary authentication questions are jokes
  - Malware plants keystroke loggers
  - Users are lured to phishing websites

Firewalls

- Firewalls are topological barriers
  - They work best if they themselves are small and simple, and enforce a limited security policy
- A large company will have hundreds of *authorized* links that go through or around the firewall

Foresight?

"The advent of mobile computing will also stress traditional security architectures... It will be more important in the future. How does one create a firewall that can protect a portable computer, one that talks to its home network via a public IP network? Certainly, all communication can be encrypted, but how is the portable machine itself to be protected from network-based attacks? What services *must* it offer, in order to function as a mobile host? What about interactions with local facilities, such as printers or disk space?"

Firewalls and Internet Security, Cheswick and Bellovin (1994)

Antivirus

- "The antivirus industry has a dirty little secret: its products are often not very good at stopping viruses."(*NY Times, 1/1/2013)*
- Most A/V programs are *reactive*; they work by looking for signatures of known malware
- The new stuff can spread quite widely before the vendors update their signature databases
- Tailored viruses may not be widespread enough to make it into some A/V programs

Patches

- Patches are necessary, to fix known vulnerabilities
- It can take a long time produce a high-quality patch
- Despite that, production software is incompatible with new patches; testing is needed
- But—"Patch Tuesday" is followed by "Exploit Wednesday"; the bad guys reverse-engineer the patches

Where Did We Go Wrong?

- Static advice
- Static advice to use static defenses
- Dynamic, adaptive adversaries in a world of rapidly changing technology

"Life is a dynamic process and can't be made static. '—and they all lived happily ever after' is fairy-tale stu—" (Robert Heinlein, *Sixth Column* (1941))

How Do We Improve?

- We cannot predict important new applications
- We cannot predict radically new devices, e.g., smartphones
- We cannot predict new classes of attacks
- We can make decent projections of improvements in CPU power, storage capacity, and price
- Is that enough?

Sometimes, Raw Power is the Threat

- One major threat to DES was brute force; this has been known since 1979
- It happened, though later than forecast by Diffie and Hellman
  - Their analysis said $20,000,000; straight-line Moore's Law would make that about $5K in 1997
  - The actual cost was about $250K
- But—we cannot predict cryptanalytic (or any other) *breakthroughs*

What Are Our *Assumptions*?

- Most security mechanisms rest on *assumptions*
- Often, these are implicit, and are not recognized even by the architects
- When our hardware, software, or usage patterns change, our assumptions can be invalidated
- But—since we never wrote them down, we don't know to look out for danger

Password Assumptions

- Attacker computing power
  - PDP 11/70?
  - Ratio of attacker/defender CPU power?
- Threat model
  - Theft of hashed password file
  - Serious limits to online guessing rate
- Limited number of passwords to be remembered
- Iterated cryptographic function can't be inverted

Only the last has held up!

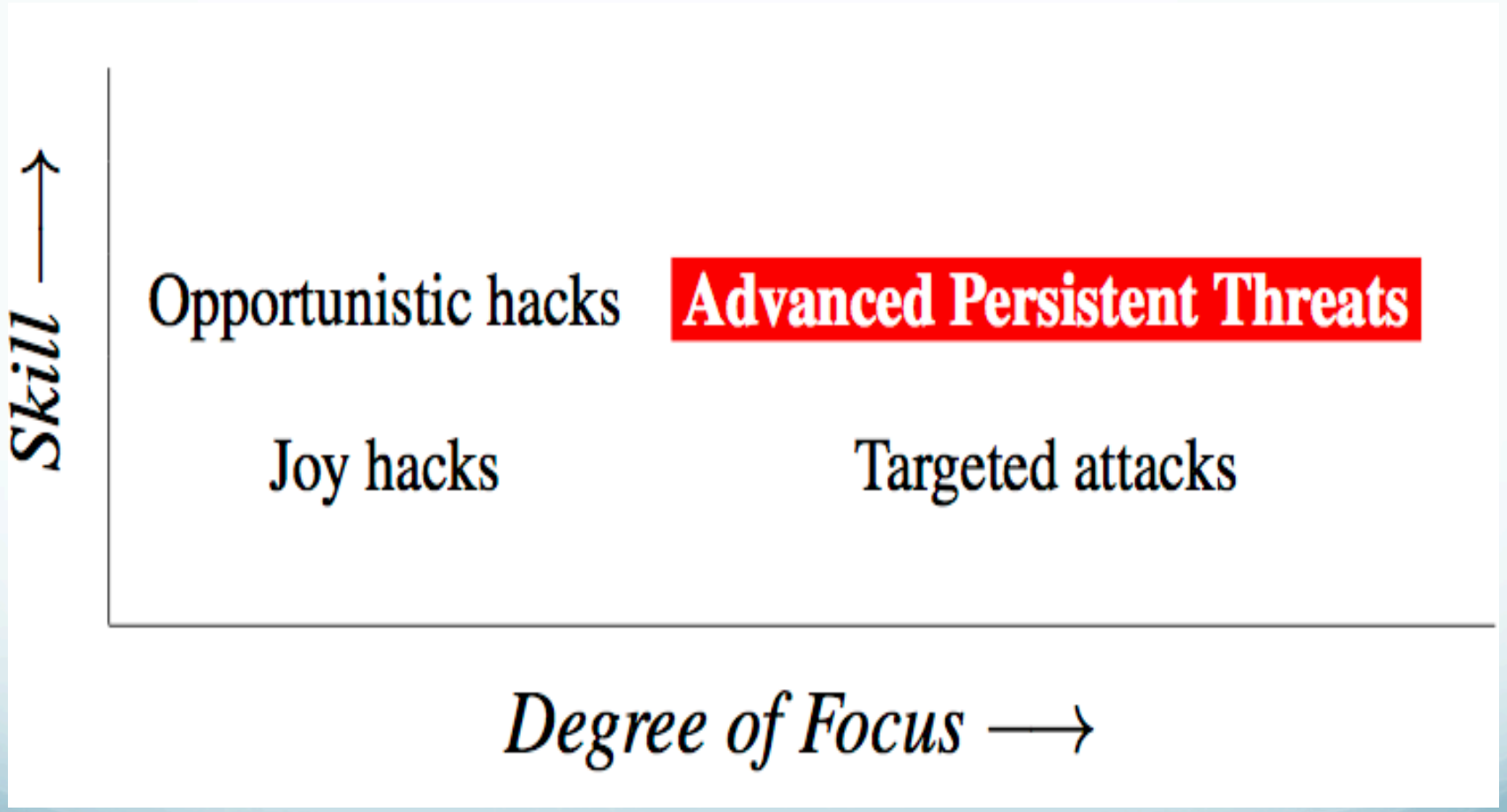
When Did These Fail?

- Attacker computing power has been increasing gradually
  - Sharp increase after 2000, with the rise of botnets
  - More recent jump with the use of GPUs
- Threat model changed around 2003, with the rise of for-profit hacking
- Number of logins has been going up since the rise of the web—hard to pinpoint a number, but it was obviously an issue 10 years ago
- But—our password policies remain about the same

Why is Threat Model Important?

- More precisely, why is it an *assumption*?
- We implicitly assume certain limits to the behavior of our enemies
  - Is someone going to break into your house to bug your keyboard?
- "Amateurs worry about algorithms; pros worry about economics" (Allan Schiffman, 2004)
- A stronger threat means the attacker has more resources

The Threat Matrix



Attacker Resources

- Joy hackers: few; primarily downloaded scripts and exploits
  - *The 1990s threat model*
- Targetiers: considerable knowledge about your systems and procedures; possibly inside access
- Opportunistic attackers: sophisticated tools; often, plenty of money
- APTs: everything, up to and including "the 3 Bs" (burglary, bribery, and blackmail)
  - We see this—*to some extent*—today

Assumptions Behind Firewalls

- Obvious: topological nature
- Less obvious: simple—i.e., comprehensible and correct—security policy
- Less obvious: all interesting protocols are efficiently protectable by a firewall
- Crucial but often ignored today: assumption that the firewall's implementation of a protocol is itself correct and secure

To some extent, *all* of these are now false

Are Firewalls Themselves Secure?

- There are far more protocols in use today
- To function, the firewall must understand all of these
- This implies a lot of code; often, a lot of very complex code
- Why should we think this code is correct?

Firewalls and Threat Models

- Joy hackers are probably stopped
- Opportunistic hackers can get through, especially with worms, phishing, and drive-by downloads
- Targetiers have detailed knowledge of topology and behavior; they may or may not be blocked
- To APTs, firewalls are just a speed-bump

Flow Monitoring Assumptions

- What are the assumptions?
- Why *should* it work?
- We assume:
  - We can capture “enough” flows
  - We will capture the evil ones
  - We will be able to spot the flows of interest

Flow Rate

- Assume actual traffic of *P* packets per second and *F* flows/second
  - Implies *P*/*F* packets per flow
- Assume maximum capture rate of *C* flows/sec
- What is the relationship of *F* and *C*?
- If *F*>>*C*, we must down-sample and will miss important flows. Ultimate success may depend on technology changes: relative growth of *F* and *C*
- Statistical sampling may mean we'll something—and with an intelligent adversary, we may miss what the attackers want us to miss
  - Assumption: the attacker can't manage that. True?

Limits to Flow Monitoring

- Size of the traffic matrix—it goes up as the *square* of the number of endpoints
- Memory bandwidth has only been increasing slowly
  - Number of endpoints and bandwidth have both increased far more quickly
  - Memory speeds haven't kept up
- Conclusion: sampling is *necessary*—but does it hurt us?
- That it doesn't is another assumption

Packets per Flow

- What is the behavior of the monitoring system for low *P/F*?
  - Is there considerable overhead for creating state for a flow?
  - Can the attacker use that to evade detection?
- Underlying assumption: behavior at low *P/F* just affects the random percentage picked up. Is this a way to hide?

Spotting Evil Flows

- Suppose the percentage of evil flows is very low—can we spot them?
- Can the attacker create enough benign-looking flows to hide amongst?
- Another assumption: evil flows have certain characteristics—size, destination, etc.—that we can spot. Can the attacker hide, via proxies and the like?
  - Attack: compromise legitimate web site your users visit; serve malware from there
- "Low and slow" attacks?

Spotting Exfiltration

- Underlying assumption: all traffic to a given destination is equivalent
  - But—sites like gmail, Facebook, etc., are multipurpose
- Second assumption: looking more deeply at flows can show anomalies
  - Can the attacker mimic them?

"And by the way, we are belittling our opponents and building up a disastrous overconfidence in ourselves by calling them pirates. They are not—they can't be. Boskonia must be more than a race or a system—it is very probably a galaxy-wide culture. It is an absolute despotism, holding its authority by means of a rigid system of rewards and punishments. In our eyes it is fundamentally wrong, but it works—*how it works*! It is organized just as we are, and is apparently as strong in bases, vessels, and personnel."

E.E. "Doc" Smith, *Galactic Patrol* (1950)

Final Thoughts

- Our defenses are built for a given threat and a given set of technologies
- Neither of these are static—and we can't be, either

Network Analysis with SiLK

Ron Bandes

SEI/CERT Network Situational Awareness

NO WARRANTY

Learning 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.
- Count and profile network flow records using basic SiLK commands.

Outline

Introduction: SiLK

Network flow

Basic SiLK tools

Advanced SiLK tools

Summary

What SiLK Does

Optimized for extremely large data collections

- Very compact record format
- Large amount of history can stay online.

Command line interface

- Good for scripting & repeating commands with small modifications.

Retrospective analysis

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

Outline

Introduction: SiLK

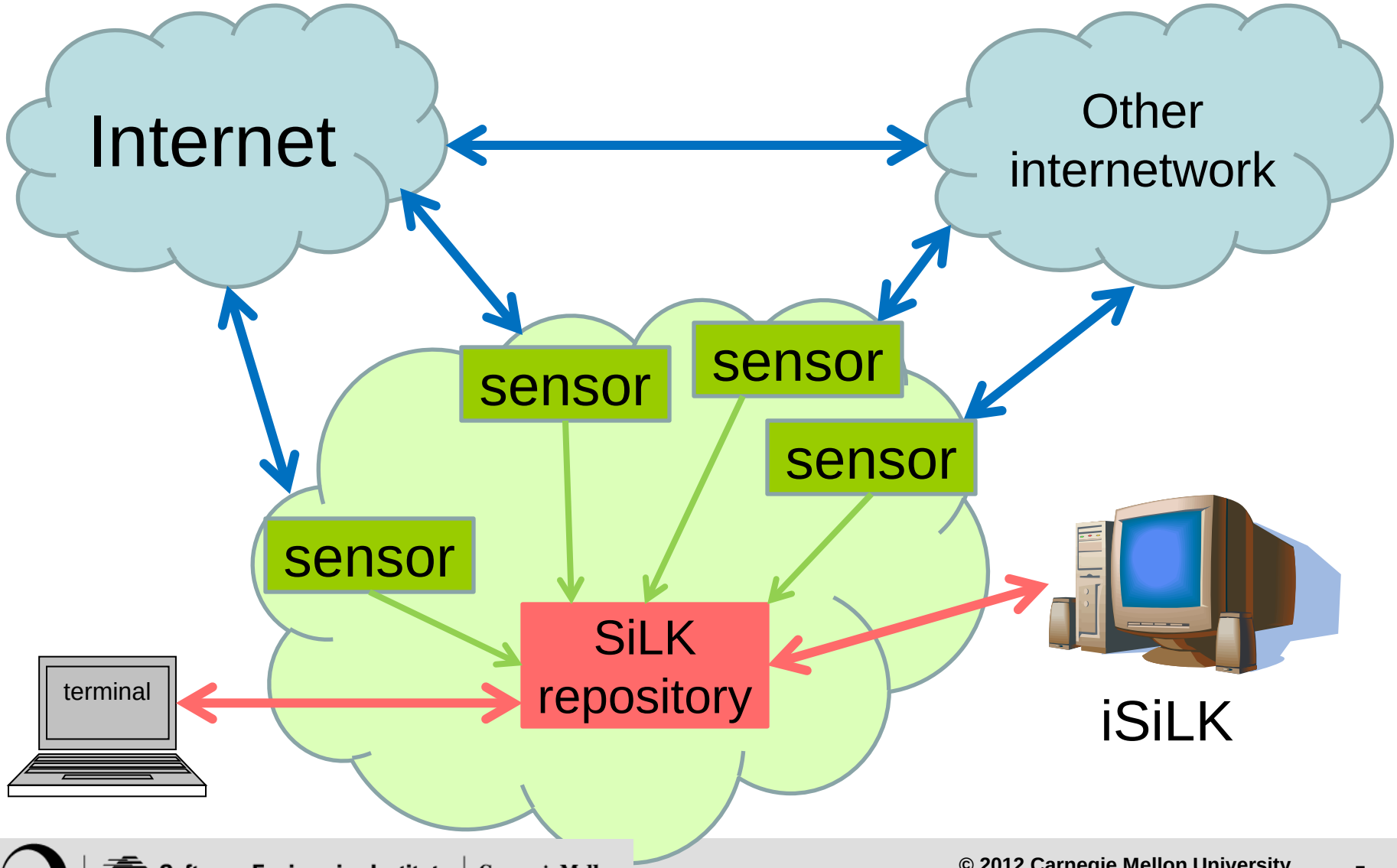
Network flow

Basic SiLK tools

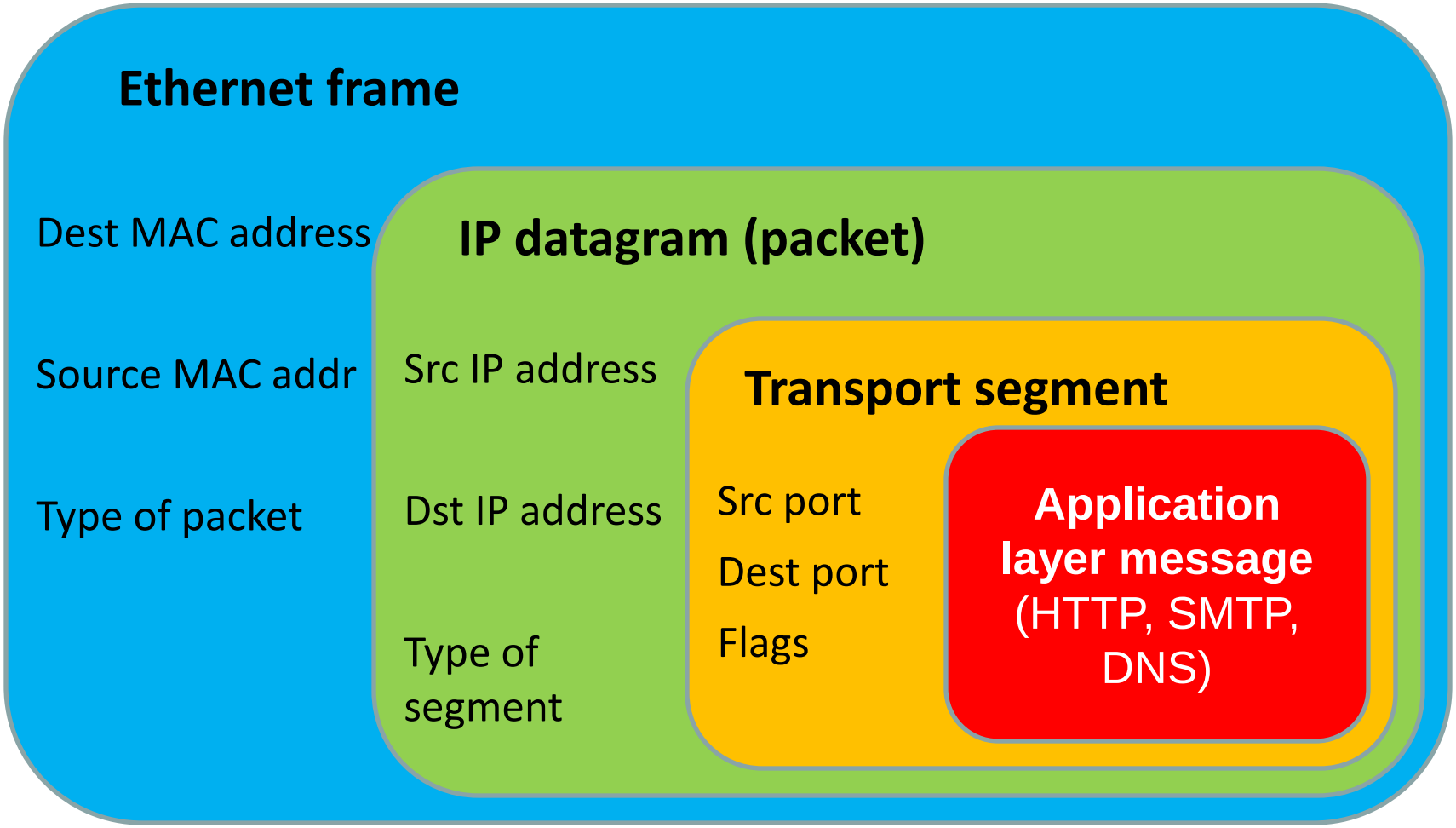
Advanced SiLK tools

Summary

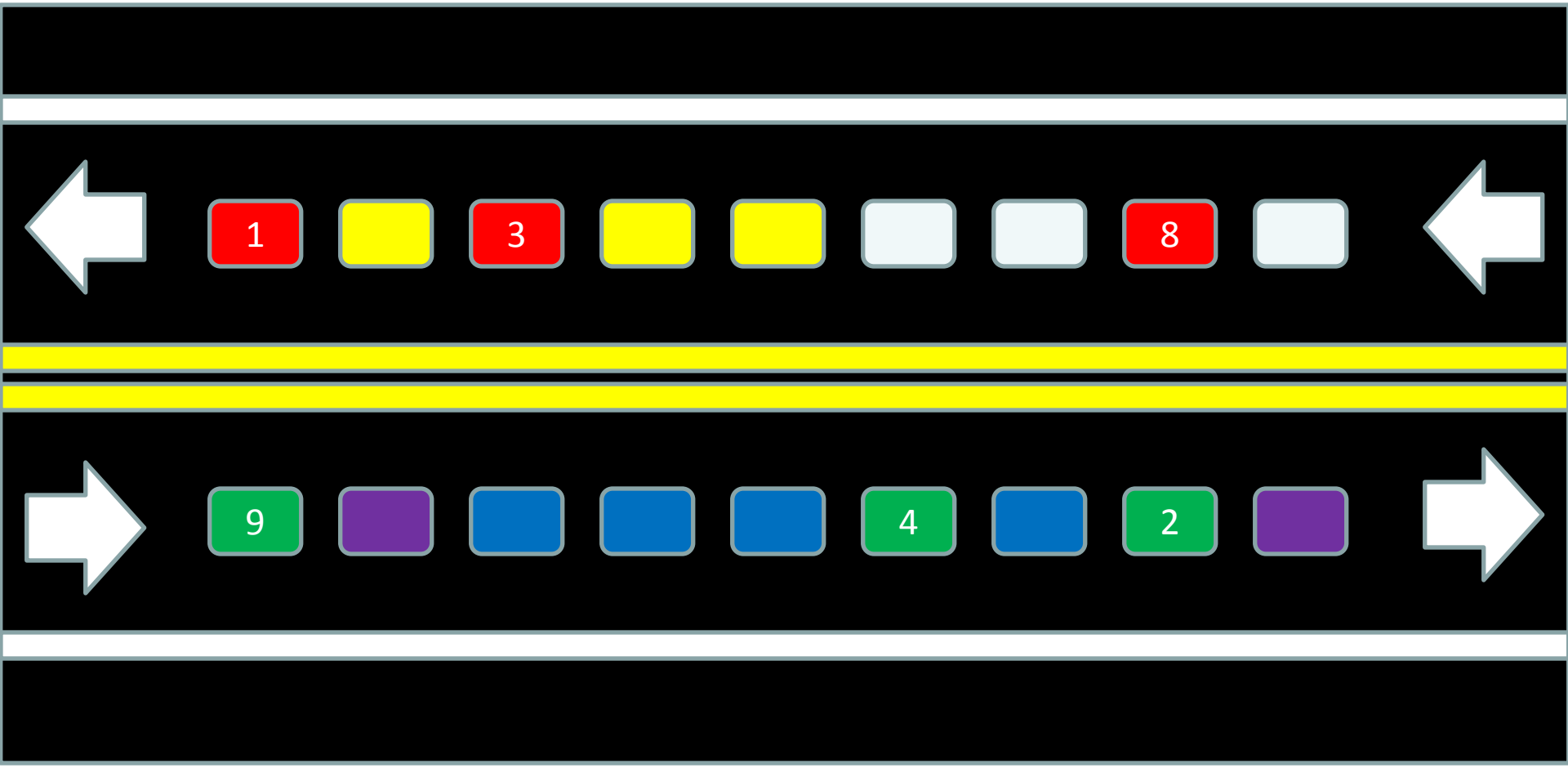
Network Monitoring



Packet Encapsulation



Flows

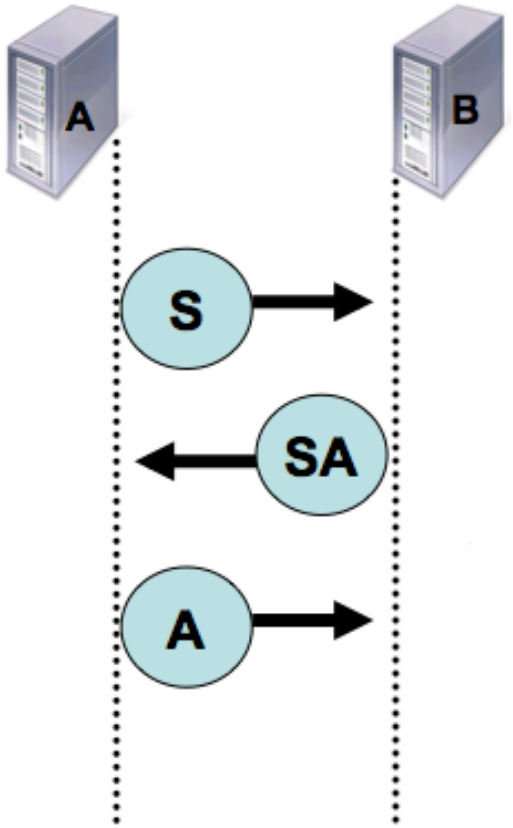


What Is a Flow?

A flow is an aggregated record of packets.

SiLK flows are ID'd by five attributes:

- source IP address
- destination IP address
- source port
- destination port
- transport protocol (any of about 130 in use)



SiLK flows are unidirectional:

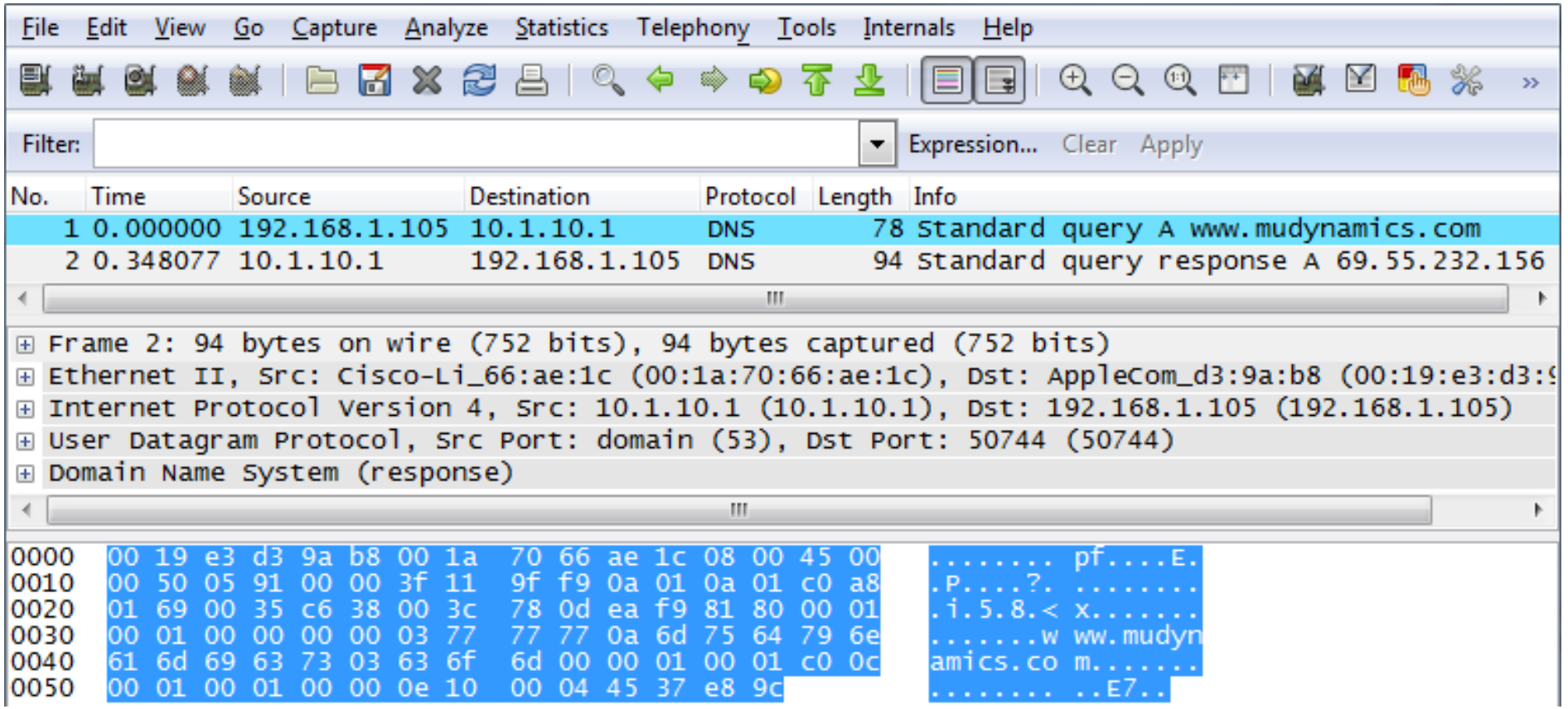
- Newly observed attributes, new flow
- Previously observed attributes, update flow

What's in a Record?

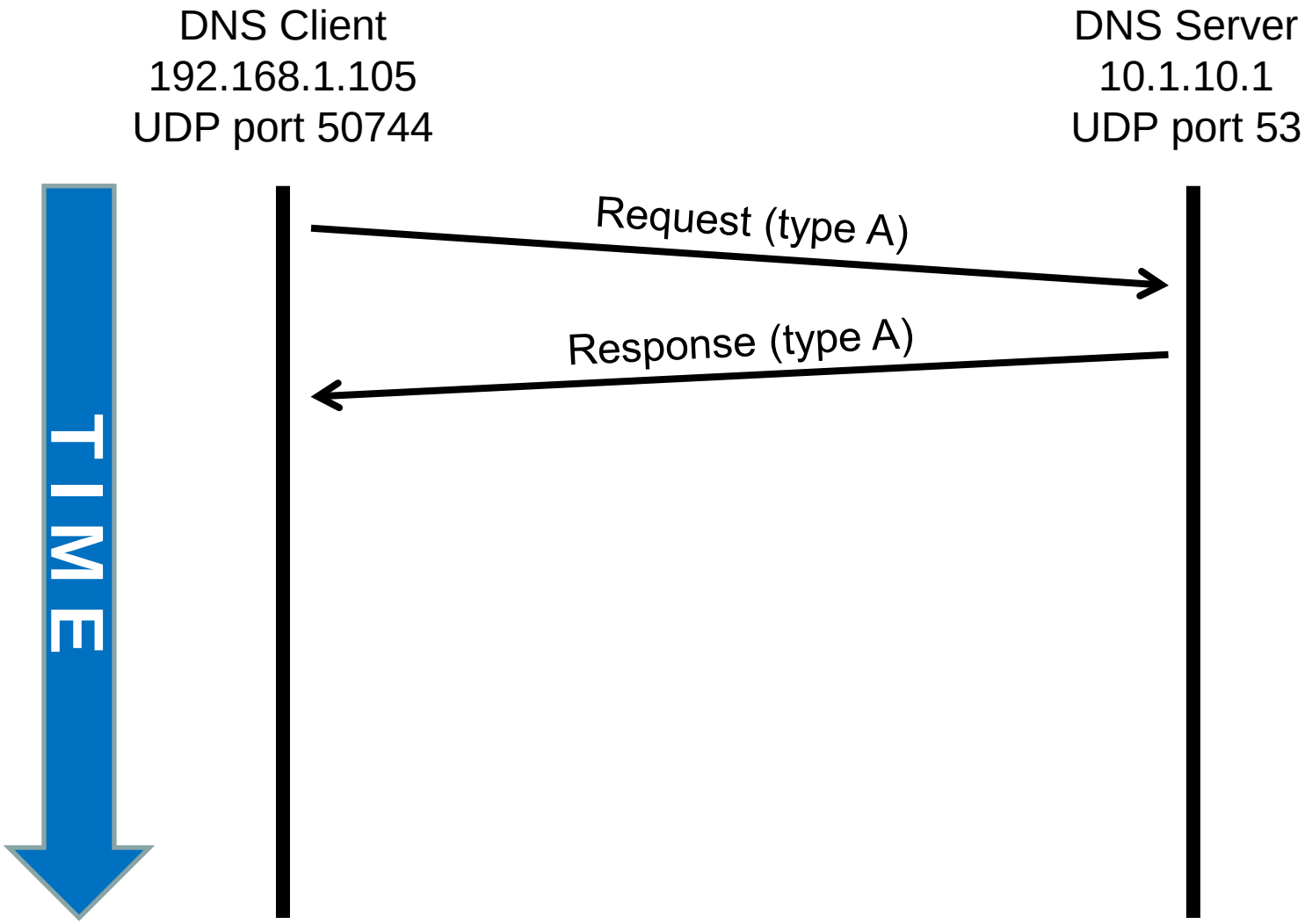
Fields found to be useful in analysis:

- 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



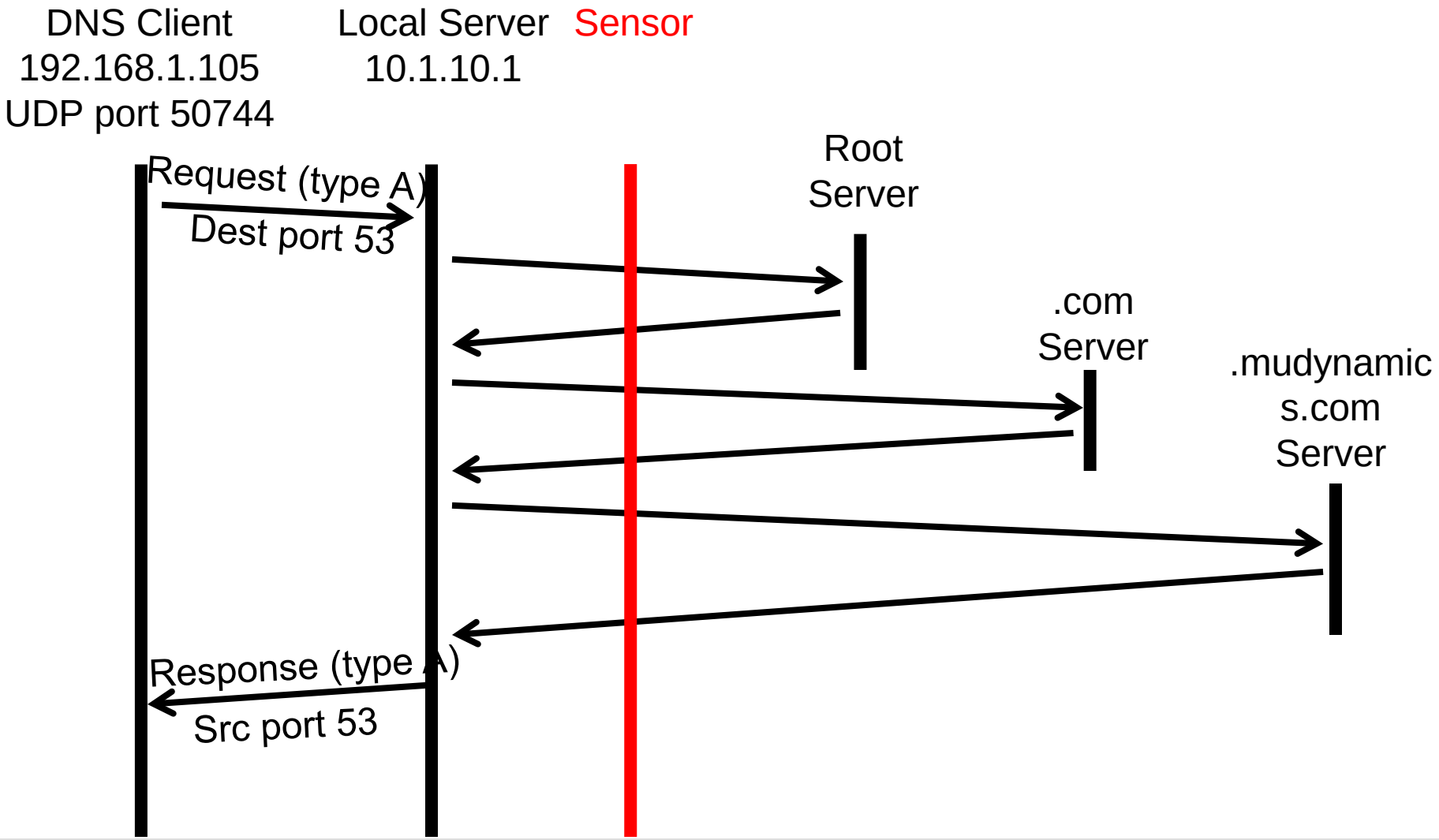
Sequence Diagram



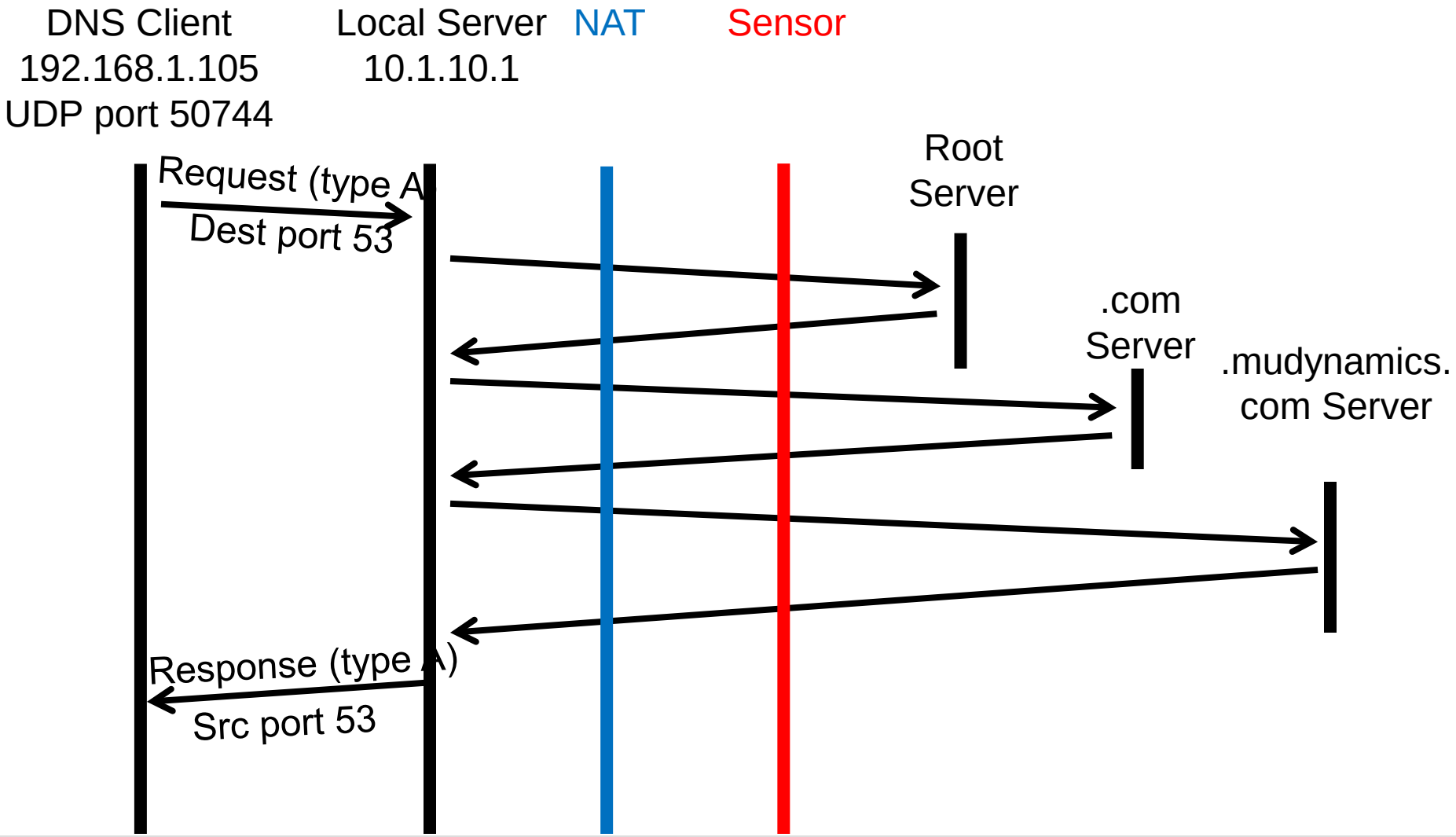
SiLK tool (rwcut) output

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

Realistic Sequence Diagram



More Realistic Sequence Diagram



What is this?

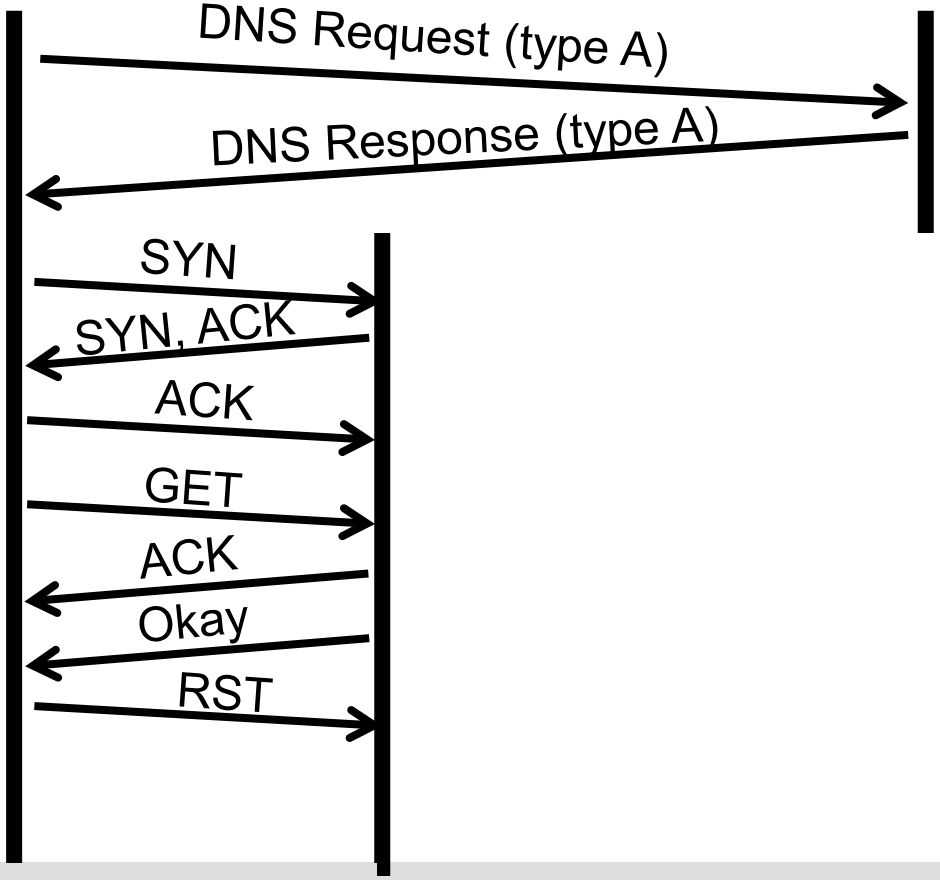
sIP	dIP	sPort	dPort	pro	packets	flags	initF	type
192.168.1.105	10.1.10.1	50744	53	17	1			out
10.1.10.1	192.168.1.105	53	50744	17	1			in
192.168.1.105	198.51.100.6	49152	80	6	4	SRPA	S	outweb
198.51.100.6	192.168.1.105	80	49152	6	3	S PA	S A	inweb

HTTP Sequence Diagram

HTTP Client
192.168.1.105

HTTP Server
198.51.100.6

DNS Server
10.1.10.1



What Is This? #1

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

What Is This? #2

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

What Is This? #3

sIP	dIP	sPort	dPort	pro	packets	bytes	flags	sTime
88.187.13.78	71.55.40.204	40936	80	6	83	3512	FS PA	2010/12/08T11:00:01.318
71.55.40.204	88.187.13.78	80	40936	6	84	104630	FS PA	2010/12/08T11:00:01.336
88.187.13.78	71.55.40.204	40938	80	6	120	4973	FS PA	2010/12/08T11:00:04.483
71.55.40.204	88.187.13.78	80	40938	6	123	155795	FS PA	2010/12/08T11:00:05.001
88.187.13.78	71.55.40.204	56172	80	6	84	3553	FS PA	2010/12/08T12:00:02.116
71.55.40.204	88.187.13.78	80	56172	6	83	103309	FS PA	2010/12/08T12:00:02.133
88.187.13.78	71.55.40.204	56177	80	6	123	5093	FS PA	2010/12/08T12:00:05.276
71.55.40.204	88.187.13.78	80	56177	6	124	157116	FS PA	2010/12/08T12:00:05.294

It’s All a Matter of Timing

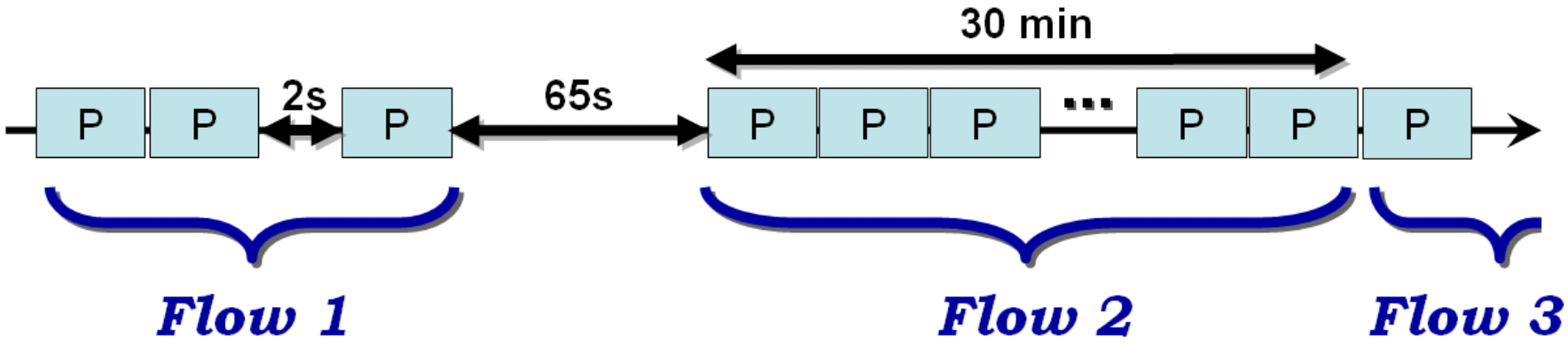
The flow buffer needs to be kept manageable.

Idle timeout

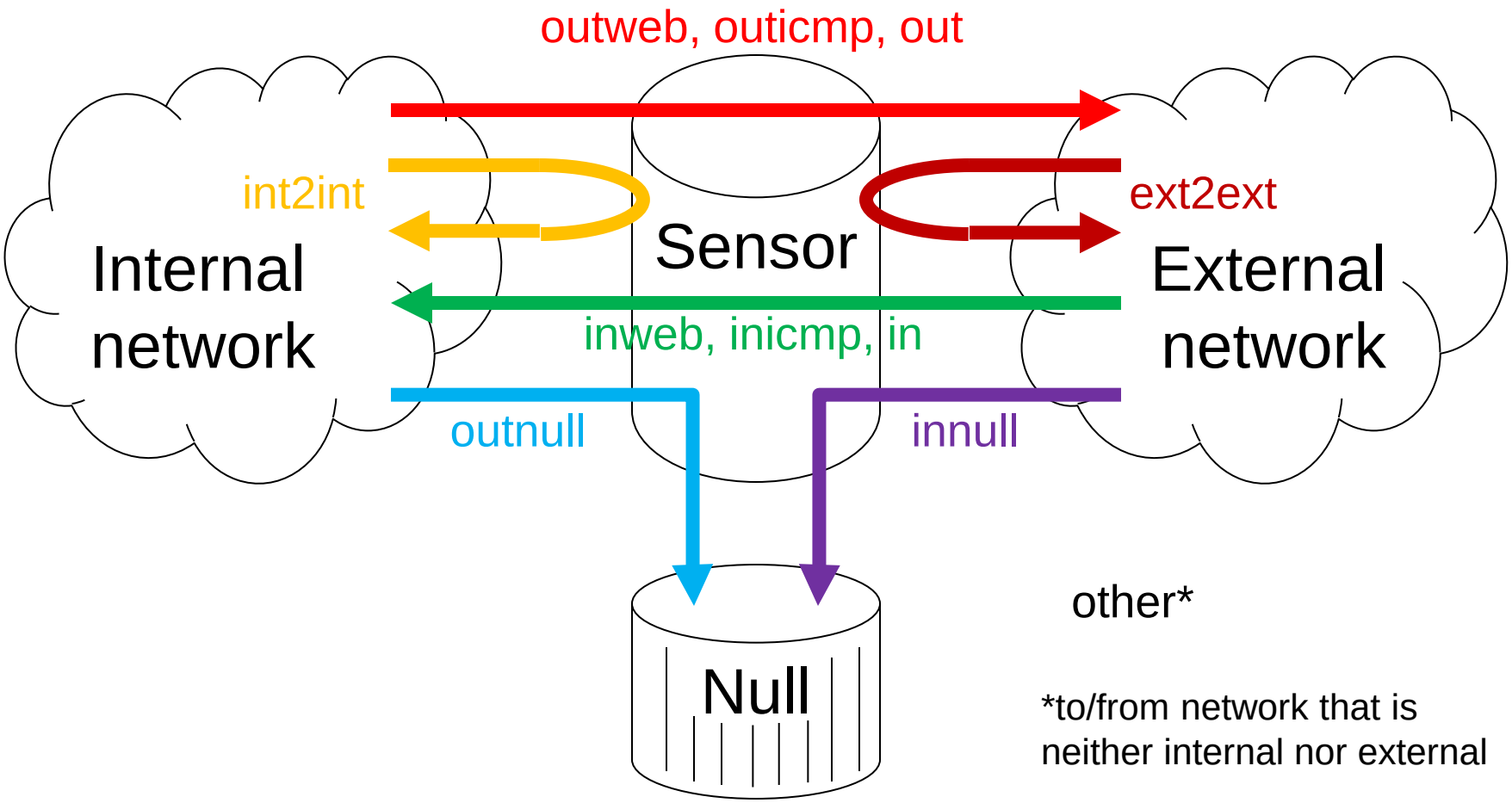
- If there is no activity within [30] thirty seconds, flush the flow.

Active timeout

- Flush all flows open for [30] thirty minutes.



SiLK Types



SiLK Types in SiLK

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

Names in **bold** are default types

Got a Question? Flow Can Help

What's on my network?

What happened before the event?

Where are policy violations occurring?

What are the most popular websites?

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?

Outline

Introduction: SiLK

Network flow

Basic SiLK tools

Advanced SiLK tools

Summary

UNIX / Linux commands

System prompt

Info + prompt character

e.g., **~ 101>**

User command

command name

options

arguments

redirections

pipe

e.g., rwcut --all-fields results.rw >results.txt

e.g., rwcut --fields=1-6 results.rw | more

Some Terms

SiLK: A traffic analysis tool-suite which processes flow data.

Flow: the collection of packets travelling in the same direction in a TCP or UDP connection.

Flow Record: a single record containing summary information for a flow.

Flow Repository: a tree structure of flat files containing flow records.

Collection, Packing, and Analysis

Collection of flow data

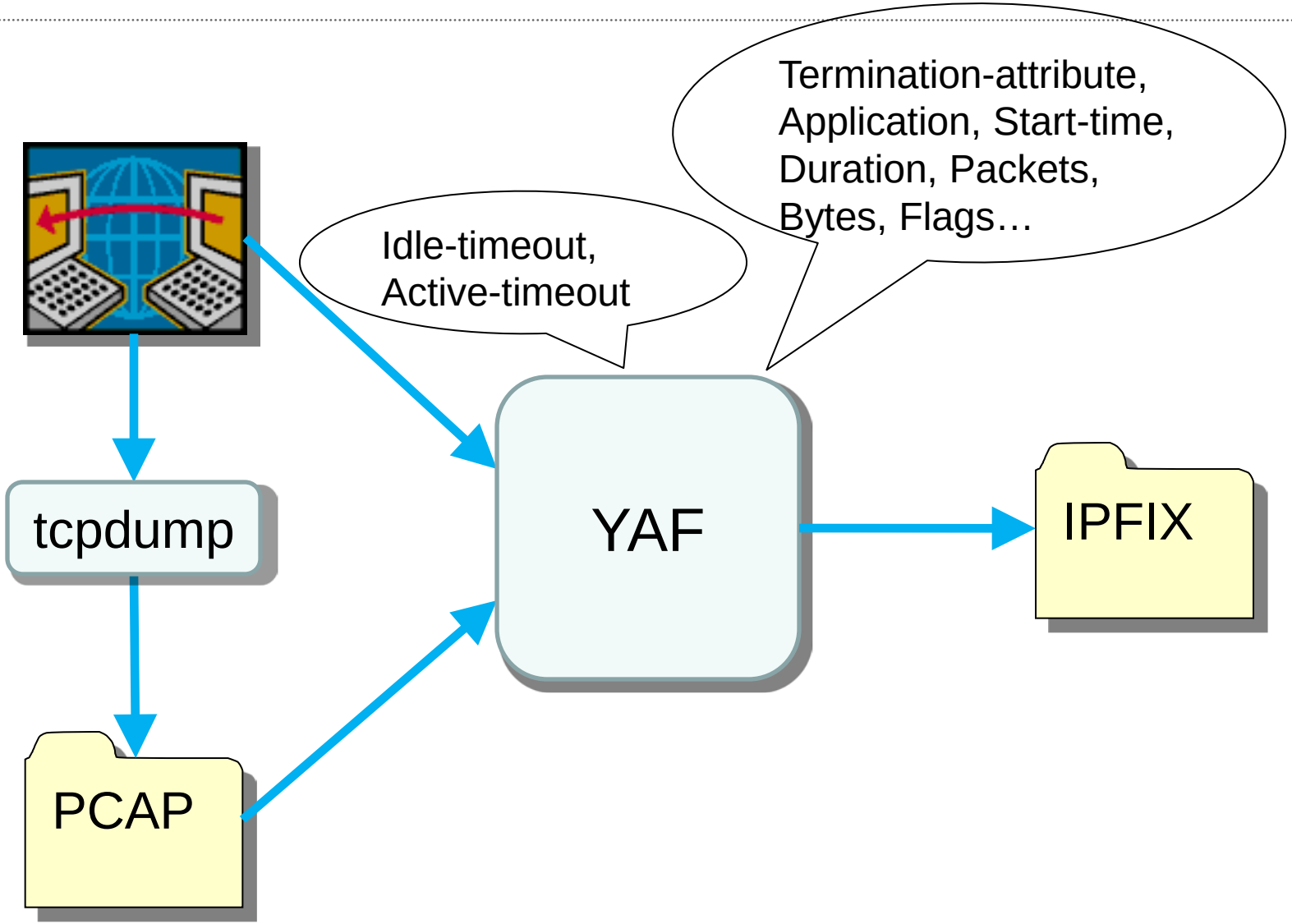
- Examines packets and summarizes into standard flow records
- Timeout and payload-size values are established during collection

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

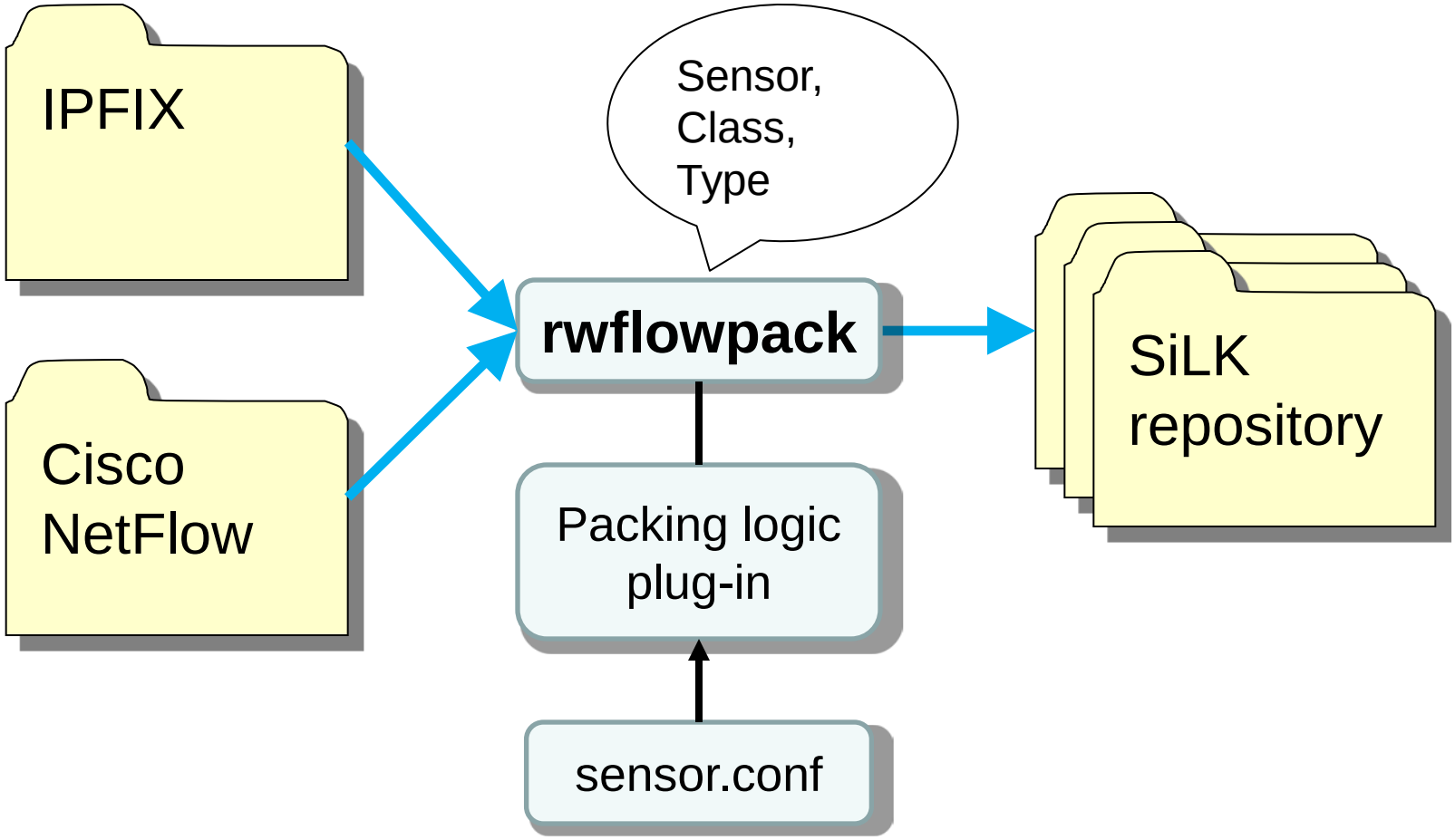
Analysis of flow data

- Investigation of flow records using SiLK tools

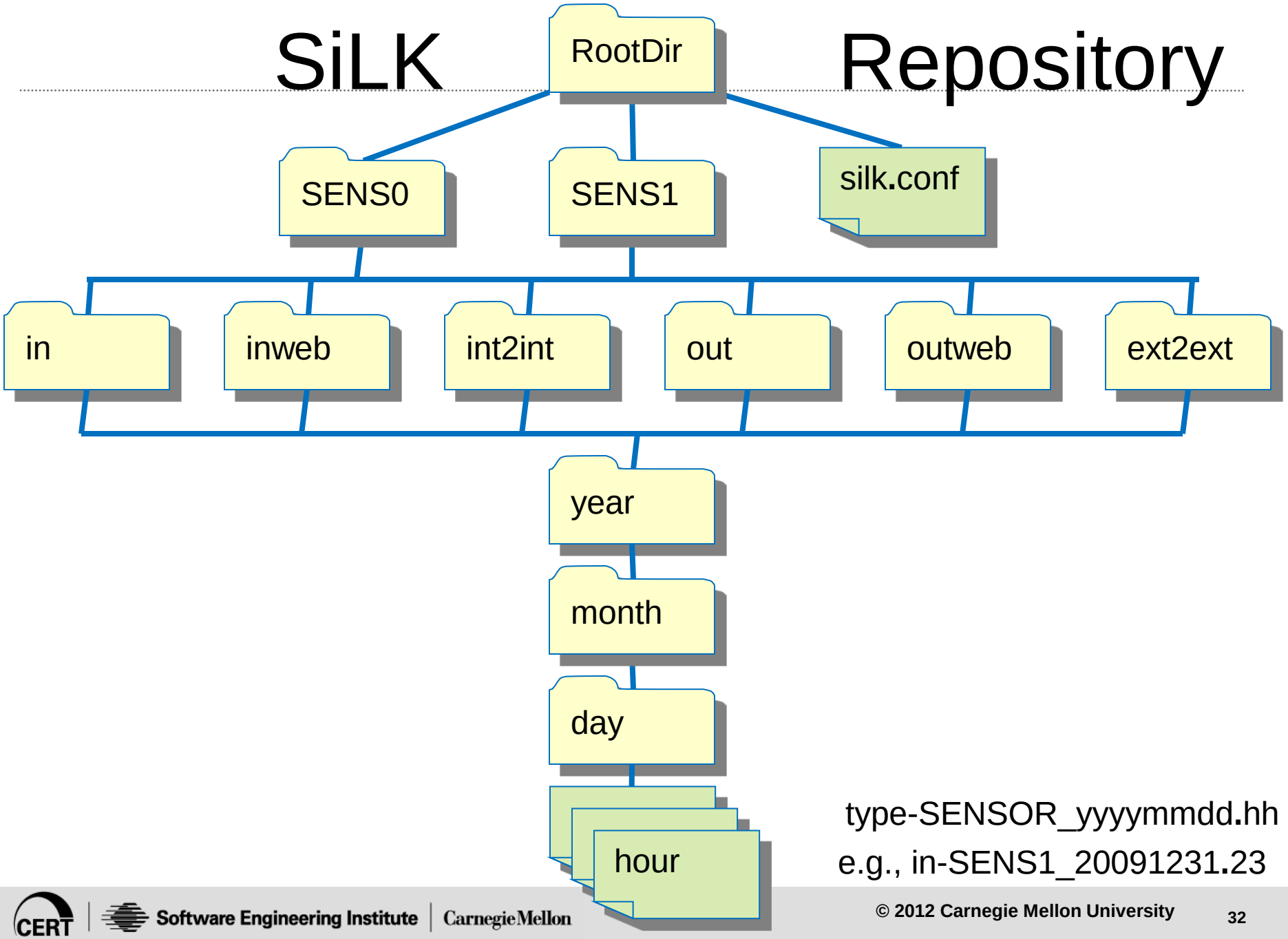
Collection



Packing



SiLK Repository



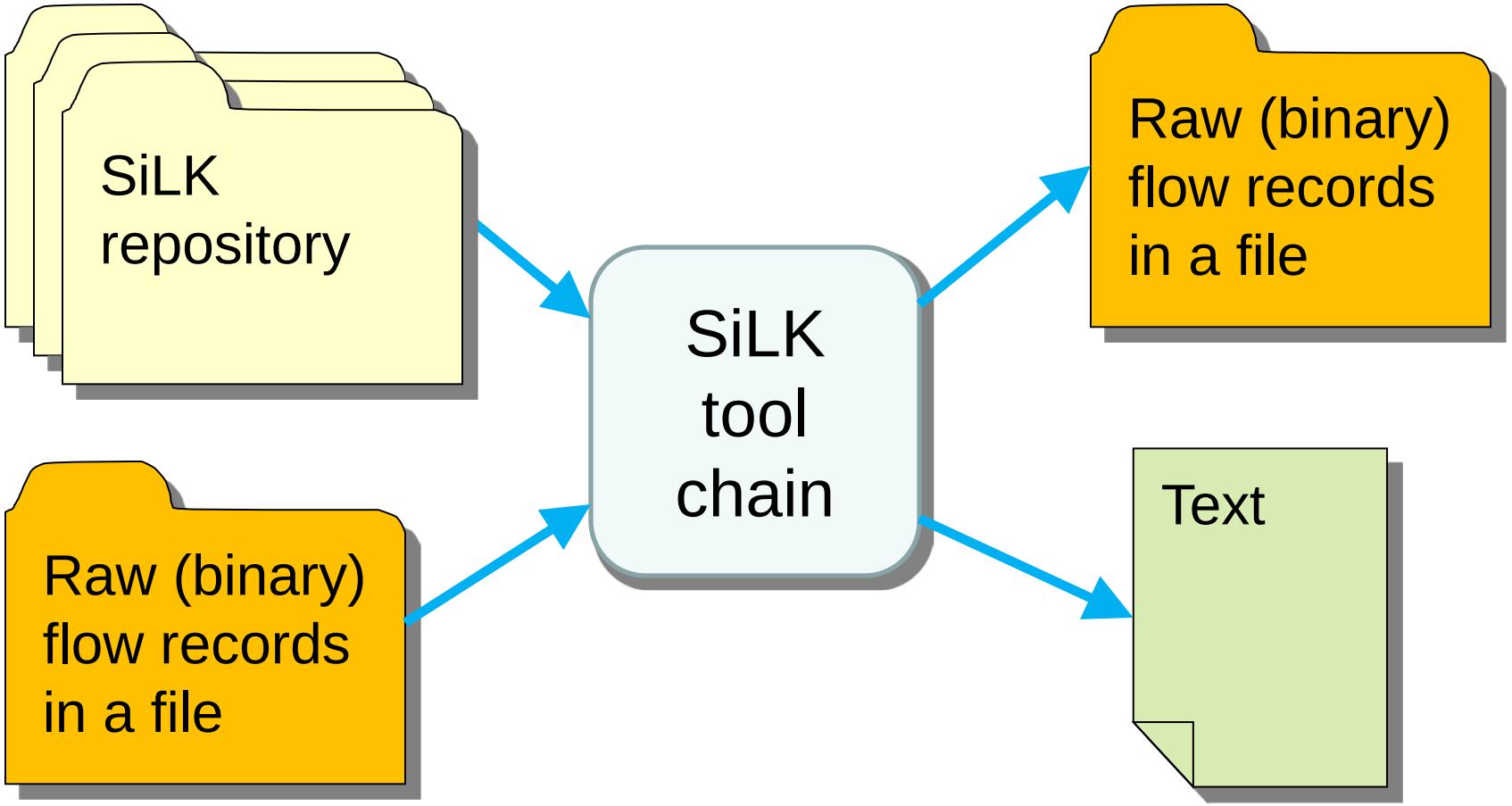
type-SENSOR_yyyymmdd.hh

e.g., in-SENS1_20091231.23

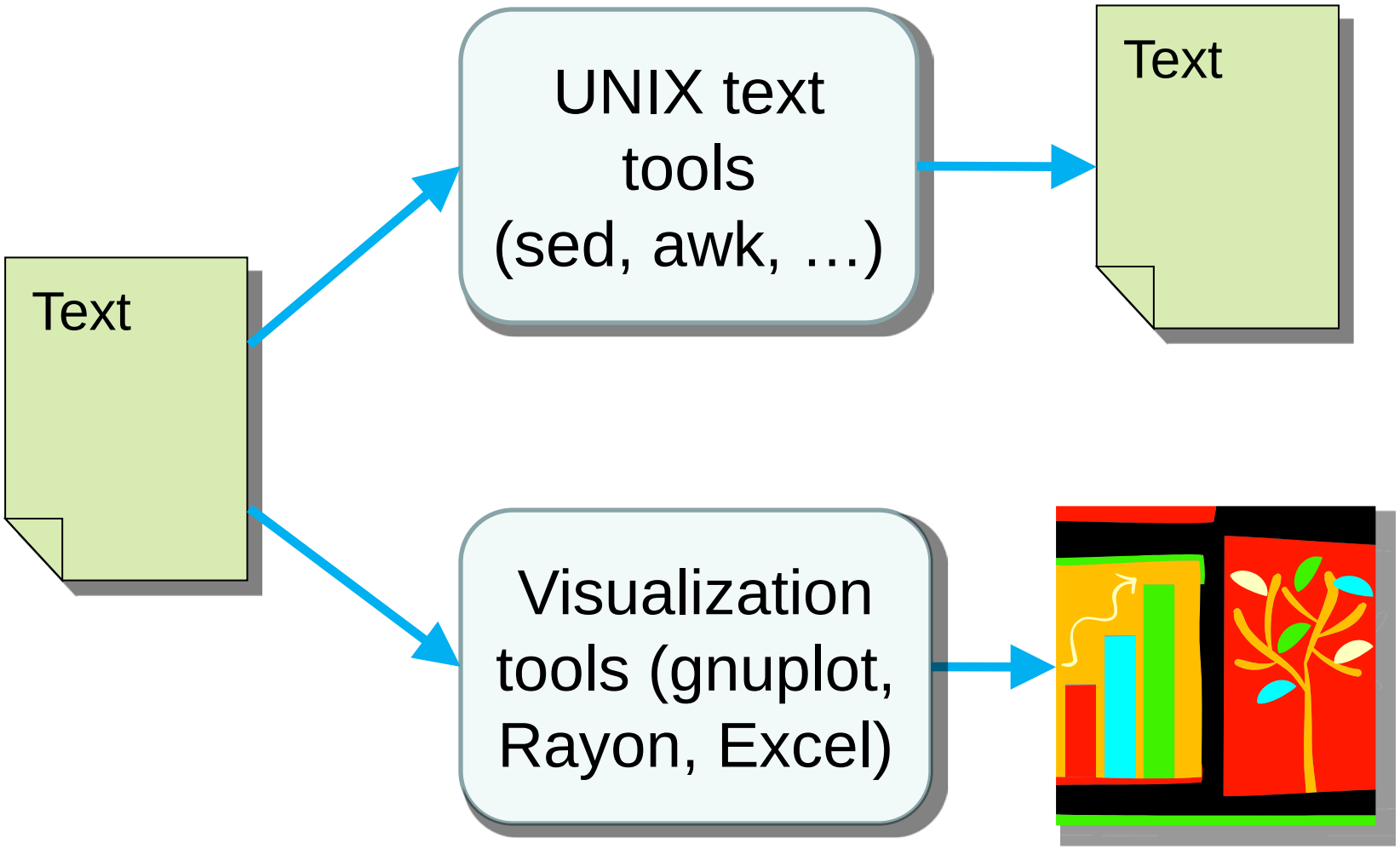
Exercise

```
PS1='\W \!> '
export SILK_IPV6_POLICY=asv4
cd /data/bluered
ls -l silk.conf
less silk.conf   # type “q” to exit from less
export SILK_DATA_ROOTDIR=/data/bluered
cd
```

Analysis



Reporting



So Much to Do, So Little Time...

We can’t discuss all parameters for every tool.

Resources

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

Exercise

```
mapsid --help
man mapsid   # type “q” to exit from man
mapsid
mapsid --print-descriptions
```

Basic SiLK Tools: `rwfileinfo`

`rwfileinfo` displays a variety of characteristics for each file format produced by the SiLK tool suite.

It is very helpful in tracing how a file was created and where it was generated.

rwfileinfo Example

```
SiLK> rwfileinfo sportSMTP.rw
sportSMTP.rw:
  format(id)           FT_RWGENERIC(0x10)
  version              16
  byte-order           littleEndian
  compression(id)      lzo1x(2)
  header-length        352
  record-length        88
  record-version       1
  silk-version         2.4.4
  count-records        5
  file-size            523
  command-lines
   1  rwfilter --type=in,out --start=2010/12/08
--end=2010/12/10 --pass=sportSMTP.rw
--any-address=139.72.231.133 --print-file --print-vol
```

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`** argument.

sip	packets	type	flags	application
dip	bytes	in	initialflags	*icmptypecode*
sport	sensor	out	sessionflags	attributes
dport	*scc*	dur	*dur+msec*	*stype*
protocol	*dcc*	stime	*stime+msec*	*dtype*
class	nhip	*etime*	*etime+msec*	

Field names in italics are *derived* fields

Rwcut Default Display

By default

- sIP, sPort
- dIP, dPort
- protocol
- packets, bytes
- flags
- sTime, eTime, duration
- sensor

`--all-fields`

`--num-recs --start --end`

These allow analyst to specify a slice of the records to display.

`num-recs`: how many records should **`rwcut`** display

`start`: how far from the top should **`rwcut`** start

`end`: how far from the bottom should **`rwcut`** start

Quick data look:

```
rwcut myfile.raw --num-recs=20 --fields=1-7,9
```

Pretty Printing SiLK Output

Default output is fixed-width, pipe-delimited data.

```
            sIP|            dIP|pro|pkts|bytes|
207.240.215.71| 128.3.48.203|  1|   1|   60|
207.240.215.71|  128.3.48.68|  1|   1|   60|
207.240.215.71|  128.3.48.71|  1|   1|   60|
```

Tools with text output have these formatting options:

- **`--no-titles`**: suppress the first row
- **`--no-columns`**: suppress the spaces
- **`--delimited`**: change how columns are marked
- **`--column-separator:`** just change the bar to something else
- **`--legacy-timestamps`**: better for import to Excel

rwcut exercise

```
cd /data/bluered
rwcut --num-recs=20 --fields=1-6 \
  S0/in/2009/04/21/in-S0_20090421.00
cd
```

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

Try **`--end=2`** and **`--no-titles`**.

Basic SiLK Tools: `rwfilter`

`rwfilter` Syntax

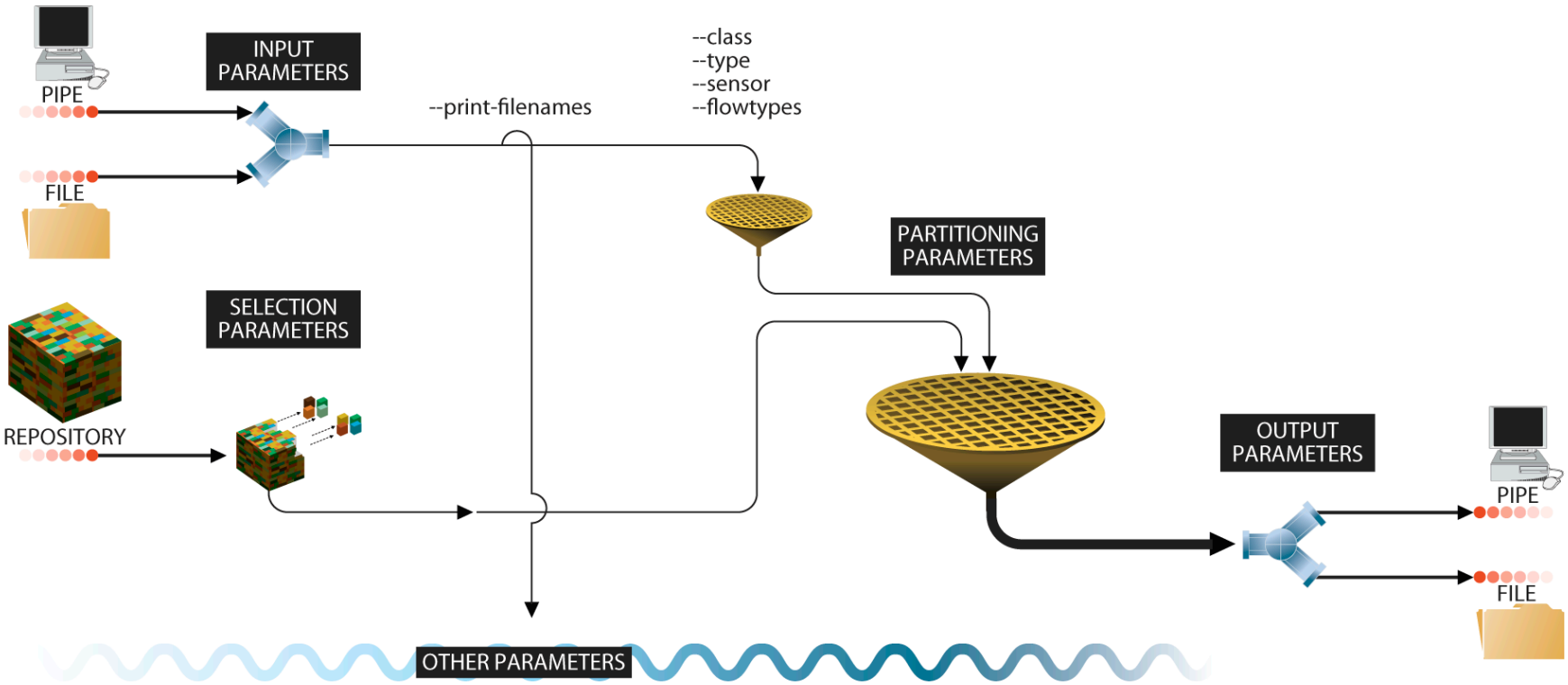
General form

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

Example call

```
rwfilter --start-date=2010/12/10:00 \
  --end-date=2010/12/10:23 --type=in \
  --protocol=0-255 --pass=all-10.raw
```

`rwfilter` Flow of Parameters

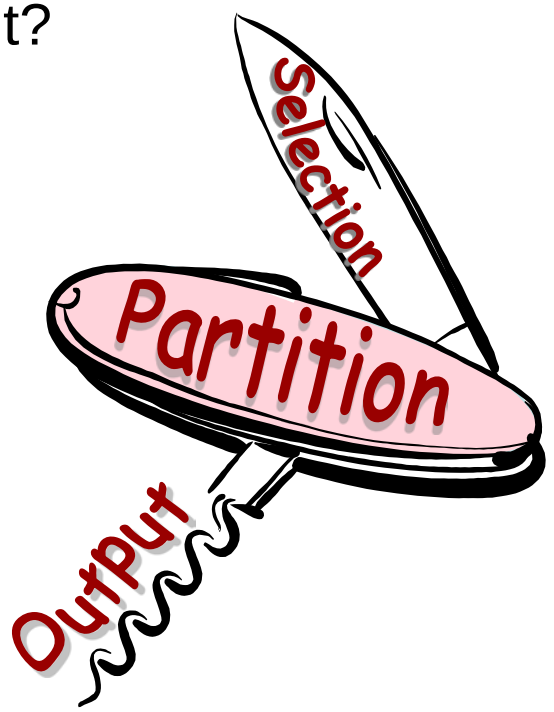


`rwfilter` Command Structure

The rwfilter command requires three basic parts:

- selection criteria or input criteria (which files are input?)
  - repository: class, sensor, type, start/end date/hour
- Partition (which records pass my criteria? Which fail?)
  - filter options: Which flows do I really want?
- output options

Partitioning is the most complex part.



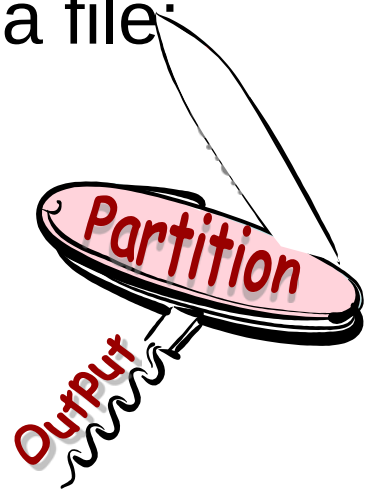
Selection Criteria

These options control access to repository files:

- **`--start-date=2007/10/03:00`**
- **`--end-date=2007/10/03T03`** (ISO format)
- **`--sensor=S0`**
- **`--class=all`**
- **`--type=in`**

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

- **`myfile.raw`**
- **`--input-pipe=stdin`**
- useful for chaining filters through stdin/stdout



--start-date and --end-date

		--start-date		
		Hour	**Day**	**None**
--end-date	**Hour**	Hours in explicit range	Ignore end-date hour. Whole days.	Error
	Day	End-hour is the same as start-hour. #hours = 1, 25, 49, …	Whole days.	Error
	None	1 hour	1 day	Current day to present time.

How Many Files are Selected?

#Files = Sensors
x Types
x Hours
– missing files

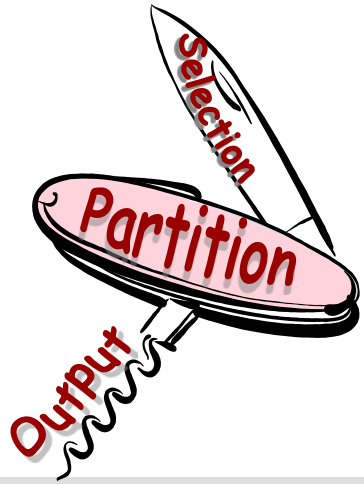
`rwfilter` Partitioning Parameters - 1

Splits flows into “pass” and “fail” groups

Lots of options

- **`saddress, daddress, any-address, not-*, next-hop-id`**
- **`sport, dport, aport`**
- **`protocol`**
- **`bytes, packets, bytes-per`**
- **`stime, etime, active-time, duration`**
- **`tcp-flags, flags-all, flags-init`**
- **`sipset, not-sipset, dipset, not-dipset`**

Frequently expanding options



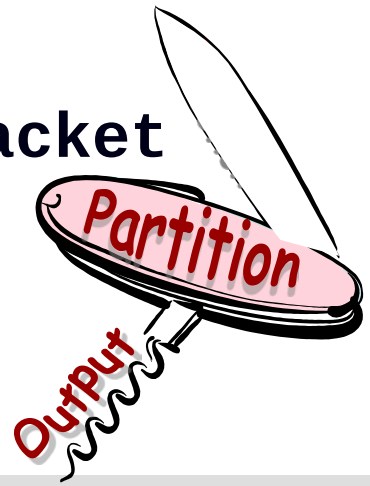
Flow Partitioning Criteria: IP Data

Pass records based on IP fields; one is required.

- **`--[not-]saddress`**, **`--[not-]daddress`**: wildcard like 12.5,7,9.2-250.x or block notation like 12.5.2.0/24
- **`--protocol`**: IP protocol
- **`--sport`**, **`--dport`**, **`--aport`**: TCP, UDP ports (caveat: ICMP)
- **`--tcp-flags=SAF; --flags-all=S/SAFR; --fin-flag=1;`** ...
- **`--icmp-type`**, **`--icmp-code`**
- **`--bytes`**, **`--packets`**, **`--bytes-per-packet`**

At least one criterion is required.

- Use **`--proto=0-`** to pass all.



What Is This? #4

```
rwfilter --type=in \
--start=2010/12/08:00 --end=2010/12/08:07 \
--daddress=71.55.0.0/16 --print-volume-stat
```

```
      |      Recs|     Packets|         Bytes| Files|
Total|  10588603|    13582511|    1756192286|     8|
 Pass|     29022|      788884|     627291737|      |
 Fail|  10559581|    12793627|    1128900549|      |
```

Rwfilter exercise

```
rwfilter --sensor=S0 --type=in \
  --start-d=2009/04/21:00 --proto=0- \
  --pass-dest=T2100.rw --max-pass=20
ls -l T2100.rw
rwfileinfo T2100.rw  # look at format(id) and
                     # at count-records
hexdump –C T2100.rw  # any readable text?
rwcut --fields=1-6 T2100.rw
```

Rwfilter exercise continued

```
rwfilter --sensor=S0 --type=in \
  --start-d=2009/04/21:00 --proto=0- \
  --pass-d=stdout --max-pass=20 \
| rwcut --fields=1-6
```

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!

- individual IP address in dotted decimal or integer
- CIDR blocks, 192.168.0.0/16
- wildcards, 10.4,6.x.2-254

Use it directly within your filter commands.

- **`--sipset`**, **`--dipset`**, **`--anyset`**

Set Tools

`rwsetbuild`: Create sets from text.

`rwset`: Create sets from binary flows.

`rwsetcat`: Print out an IP set into text.

`rwsetmember`: Test if IP is in given IP sets.

`rwsettool`: Perform set algebra (set, union, intersection) on multiple IP sets.

What Is This? - #5

```
SiLK> more MSSP.txt
171.128.2.0/24
171.128.212.0/24
SiLK> rwsetbuild MSSP.txt MSSP.set
SiLK> rwfilter --start=2010/12/8 --anyset=MSSP.set \
>  --pass=MSSP.rw --print-vol
      |        Recs|     Packets|         Bytes|Files|
Total|    30767188|    81382782|    35478407950|   48|
 Pass|    26678669|    31743084|     1464964676|     |
 Fail|     4088519|    49639698|    34013443274|     |

SiLK> rwset --sip-file=MSSPsource.set MSSP.rw
SiLK> rwsettool --intersect MSSP.set MSSPsource.set \
>   --output=activeMSSP.set
SiLK> rwsetcat --count-ips activeMSSP.set
22
```

What Is This? - #6

```
SiLK> rwfilter --type=out --start=2010/12/08 \
> --proto=0-255 --pass=stdout \
> | rwset --sip-file=outIPs.set
SiLK> rwsetcat --network-structure=24 outIPs.set
     71.55.40.0/24| 246
  149.249.114.0/24| 256
   155.208.66.0/24| 256
   177.71.129.0/24| 80
   177.249.19.0/24| 256
   177.252.24.0/24| 256
  177.252.202.0/24| 256
```

Exercise

Make a set-file of addresses of all actual inside hosts.

Should we examine incoming or outgoing traffic?

Make a set-file of all outside addresses.

Can you make both sets with one command?

Exercise solution

```
rwfilter --sensor=S0 --type=out,outweb \
    --start-d=2009/04/21 --end=2009/04/23 \
    --proto=0- --pass=stdout \
| rwset --sip-file=insidehosts.set \
         --dip-file=outsidehosts.set
```

Exercise

Examine the two set-files.

Exercise solution

```
ls –l insidehosts.set
rwfileinfo insidehosts.set
rwsetcat insidehosts.set

ls –l outsidehosts.set
rwfileinfo outsidehosts.set
rwsetcat outsidehosts.set | less
```

Exercise

Which /24 networks are on the inside?

Which /24 networks are on the outside?

Exercise solution

```
rwsetcat --network-struc=24 insidehosts.set
rwsetcat --network-struc=24 outsidehosts.set
```

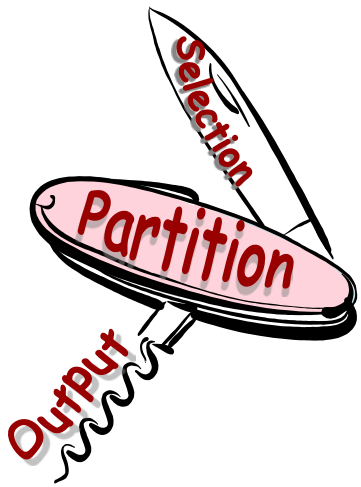
Flow Partitioning Criteria: Time

Start-date and end-date choose repository files but do not look at the actual flow records.

- **`--stime`**, **`--etime`**: choose flows which start (or end) within a time range
- **`--active-time`**: flows active in a time range
- time format: YYYY/MM/DD:HH:MM:SS.mmm
  examples: 2009/12/16:01:14:30.043 or 2009/12/16:01:14
- time range format: [Time]-[Time]

Duration

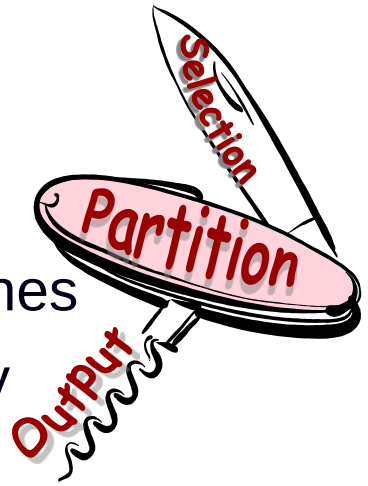
- **`--duration=1-10`**: number of seconds the flow was active



Flow Partitioning Criteria: Advanced

Tend to use these as you gain experience:

- **`--max-pass`**: limit the number of records passed
- **`--tuple-file`**: specific combinations of addr, port, proto
- **`--scc`**, **`--dcc`**: country codes
- **`--pmap`**: prefix map
- **`--python-exp`**: use an expression
- **`--python-file`**: run a script to create new switches
- **`--dynamic-library`**: dynamically loaded library



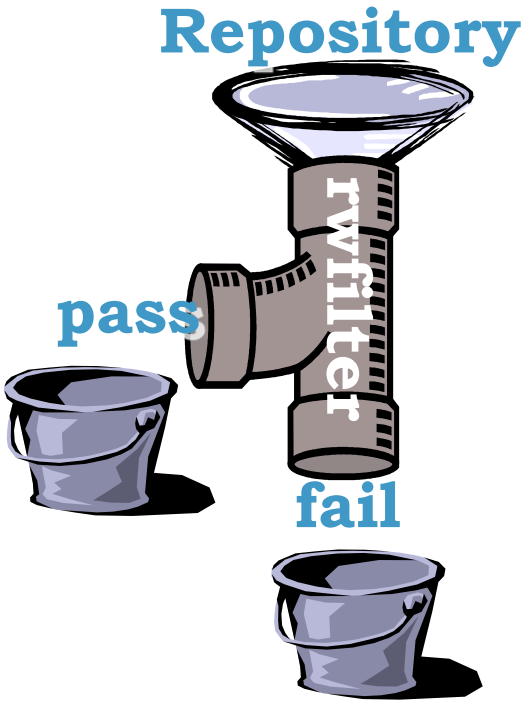
Output Criteria

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 (no screen allowed).

Other useful output

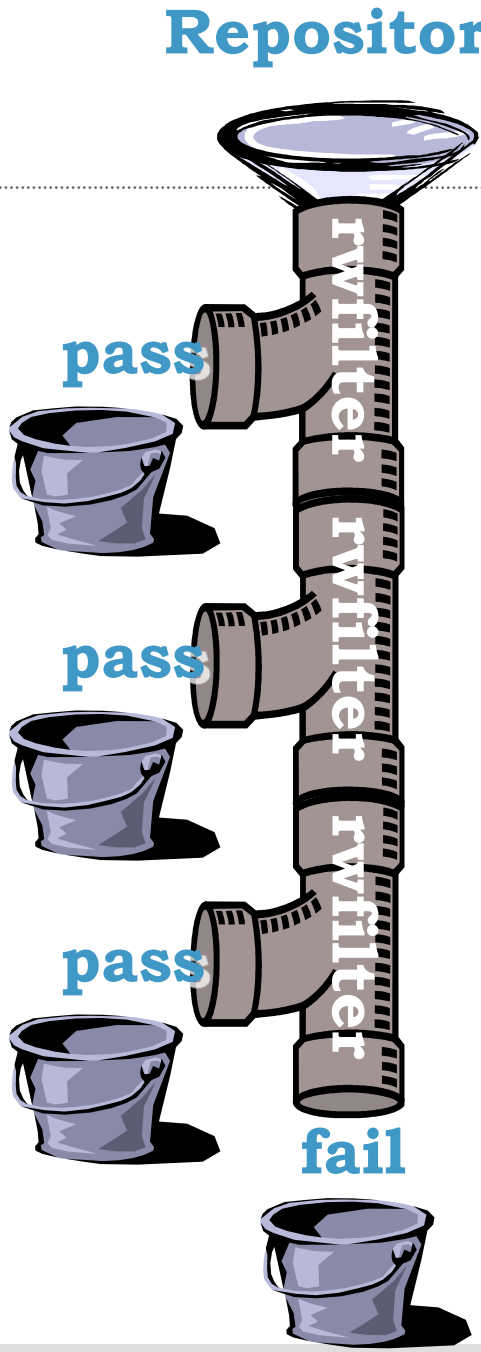
- **`--print-filenames`**, **`--print-missing-files`**
- **`--print-statistics`** or **`--print-volume-statistics`**



Chaining Filters

It is often very efficient to chain rwfilter commands together:

- Use **`--pass`** and **`--fail`** to segregate bins.
- Use **`--all`**, so you only pull from the repository once.



What Is This? #7

```
rwfilter \
      --start-date=2010/12/08 \
      --type=outweb \
      --bytes=100000- \
      --pass=stdout \
| rwfilter \
      --input-pipe=stdin \
      --duration=60- \
      --pass=long-http.rw \
      --fail=short-http.rw
```

Other `rwfilter` Parameters

`--help`: lists the available rwfilter parameters

`--dry-run`: tests the command (useful for scripting)

`--version`: tells how rwfilter is configured

`--ip-version`: filters for ipv4 or ipv6 data (if configured)

`--threads`: uses multiple threads to filter

Tips with `rwfilter`

Narrow time, type, and sensor as much as possible (fewer records to check).

Include as many partitioning parameters as possible (easy to be vague and get too much data).

Can do multiple queries and merge results

Can do further filtering to narrow results

Iterative exploration

Example Typos

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

SiLK 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.**

Basic SiLK Counting Tools: `rwcount`, `rwstats`, `rwuniq` (1)

"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 the results."
- potential scanners:
  - "Count unique dIPs by sIP and print the sources that contacted more than 100 destinations."

Basic SiLK Counting Tools: `rwcount`, `rwstats`, `rwuniq` (2)

`rwcount`: count volume across time

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

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

“Volume” = {Records, Bytes, Packets} and a few others—measure

“Key field” = SiLK fields to be measured and listed

Each tool reads raw binary flow as input.

Calling `rwcount`

- count records, bytes, and packets by time and print results
- fast, easy way of summarizing volumes as a time series
- great for simple bandwidth studies

`rwcount` Counting Options (1)

Key	Volume	Summary
`--bin-size=S`	no options	**`--skip-zeroes`**
`--load-scheme=N`	(always Records, Bytes, Packets)	**`--start-epoch`** **`--end-epoch`**

Key field is *always* time.

- Specify **`--bin-size`** in seconds.
- Use **`--load-scheme`** to select a method for counting records whose sTime, eTime straddle several bins.

Volume is *always* three columns: Records, Bytes, Packets.

`rwcount` Counting Options (2)

Key	Volume	Summary
`--bin-size=S`	no options	**`--skip-zeroes`**
`--load-scheme=N`	(always Records, Bytes, Packets)	**`--start-epoch`**
		`--end-epoch`

Limited summary options for printing output

- Include/exclude time bins with count = 0.
- Specify a minimum start and/or maximum end time.

What Is This? #8

```
SiLK> rwcount MSSP.rw --bin-size=3600
                Date|    Records|       Bytes|    Packets|
2010/12/08T00:00:00| 1351571.66| 73807086.40| 1606313.61|
2010/12/08T01:00:00| 1002012.43| 54451440.59| 1185143.62|
2010/12/08T02:00:00| 1402404.61| 77691865.26| 1675282.27|
2010/12/08T03:00:00| 1259973.65| 68575249.90| 1491393.08|
2010/12/08T04:00:00|  939313.56| 51410968.24| 1118584.81|
2010/12/08T05:00:00|  459564.75| 80862273.32| 1742058.62|
2010/12/08T06:00:00| 1280651.23| 69881126.41| 1519435.24|
...
```

Demo

Time series for all outgoing traffic on S0:

```
rwfilter --sensor=S0 --type=out,outweb \
    --start=2009/04/21 --end=2009/04/23 \
    --proto=0- --pass=stdout \
| rwcount --bin-size=3600
```

Exercise

Produce a time-series with 30-minute intervals, analyzing incoming ICMP traffic collected at sensor S1 on April 21, 2009.

Exercise solution

```
rwfilter --sensor=S0 --type=in,inicmp \
    --start=2009/04/21 --proto=1 \
    --pass=stdout \
| rwcount --bin-size=1800
```

Calling `rwstats`

`rwstats --overall-stats`

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

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

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

`rwstats` Counting Options

Key	Volume	Summary
`--fields={` **`sport, dport, icmp, protocol, sip, dip, sport, dport, …}`**	**`--value={`** **`records	bytes	packets	sip-distinct	dip-distinct}`**	**`--count=N`** **`--threshold=N`** **`--percentage=N`** **`--top`** **`--bottom`**

Use **`--top`** or **`--bottom`** to specify

top N or bottom N keys (with --count)

- volume greater or less than N (with **`--threshold`**, **`--percentage`**)

What Is This? #9

```
SiLK> rwfilter outtraffic.rw \
> --stime=2010/12/08:18:00:00-2010/12/08:18:59:59 \
> --pass=stdout \
> | rwstats --fields=sip --values=bytes --count=10
INPUT: 1085277 Records for 1104 Bins and 4224086177 Total Bytes
OUTPUT: Top 10 Bins by Bytes
                sIP|              Bytes|    %Bytes|   cumul_%|
        71.55.40.62|         1754767148| 41.541935| 41.541935|
       71.55.40.169|         1192063164| 28.220617| 69.762552|
       71.55.40.179|          331310772|  7.843372| 77.605923|
       71.55.40.204|          170966278|  4.047415| 81.653338|
     177.249.19.217|          122975880|  2.911301| 84.564639|
        71.55.40.72|          110726717|  2.621318| 87.185957|
       71.55.40.200|          101593627|  2.405103| 89.591060|
     177.71.129.255|           40166574|  0.950894| 90.541954|
        71.55.40.91|           35316554|  0.836076| 91.378030|
    149.249.114.204|           26634602|  0.630541| 92.008571|
```

Exercise

What are the top 10 incoming [IP] protocols on April 22, 2009, collected on S0?

Exercise solution

```
rwfilter --sensor=S0 --type=in,inweb \
   --start=2009/04/22 --prot=0- --pass=stdout \
| rwstats --fields=protocol --value=rec --count=10
```

Exercise 2

Top 10 inside hosts according to how many outside hosts they communicate with.

Use **--value=dip-distinct**

Exercise 2 answer

```
rwfilter --sensor=S0 --type=out,outweb --proto=0- \
    --start-d=2009/4/22 --pass=stdout \
| rwstats --fields=sip --value=dip-distinct --count=10
```

Calling `rwuniq`

`rwuniq --fields=KEYS --value=VOLUME`

- most flexible of the counting tools
  - —does not support Top-N key sorting
  - —does support multiple key queries and multiple volume summaries
- runs much faster on input sorted by key fields
  - —Use **`--presorted-input`** when this is the case.
- (standard output formatting options – see "**man rwuniq**")

`rwuniq` Counting Options

Key	Volume	Summary

`--fields=KEYS`
 	
`--bin-time=SECS`
	`--value={`
	`flows | bytes |`
	`packets |`
	`sip-distinct |`
	`dip-distinct |`
	`stime | etime}…`
		`--sort-output`
		`--VOLUME=MIN`
		`--VOLUME=MIN-MAX`

KEYS is any valid specification of SiLK fields:

- `rwuniq --fields=sIP,sPort,sTime --bin-time=60`
- `rwuniq --fields=1-5`

Choose *any* combination of volumes, or `--all-counts` for all.

Use `--sort-output` to sort by *key*, not by volume (no Top-N lists).

What Is This? #10

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

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

What Is This? #11

```
SiLK> rwuniq outtraffic.rw --fields=dip \
> --values=sip-distinct,records,bytes --sip-distinct=400- \
> --sort-output
              dIP|sIP-Distin|     Bytes| Records|
    13.220.28.183|         512|     20480|     512|
     171.128.2.27|         448|  19069280|  476732|
    171.128.2.179|         448| 139501200| 3487530|
   171.128.212.14|         448| 139467440| 3486686|
  171.128.212.124|         448| 127664480| 3191612|
  171.128.212.127|         448|  66611560| 1665289|
  171.128.212.188|         448| 139467680| 3486692|
  171.128.212.228|         448| 139393160| 3484829|
  245.225.153.120|         763|     30520|     763|
  245.238.193.102|        1339|    179480|    4487|
```

Basic SiLK Tools: `rwsort`

Why sort flow records?

- Records are recorded as received, not necessarily in time order.
- Analysis often requires finding outliers.
- You can also sort on other fields such as IP address or port to easily find scanning patterns.
- It allows analysts to find behavior such as beaconing or the start of traffic flooding.

`rwsort` Options

`--fields` (same as rwcut) is required.

input, output (stdin/stdout are defaults.)

For improved sorts, specify a buffer size.

For large sorts, specify a temporary directory.
Temporary files stored in /tmp by default

```
rwsort myfile.raw --fields=stime,sip \
   --temp-dir=. >newfile.raw

rwsort --fields=sip,sport,dport myfile.raw    \
| rwuniq --fields=sip,sport,dport --presorted \
         --dip-distinct
```

I Only Believe What I See

You'll be tempted to work with text-based records.

- It's easy to see the results and post-process with other tools (e.g., Perl, awk, sed, sort).
- It takes a lot of space, and it's ***much, much*** slower.

Guiding principle: Keep flows in binary format as long as possible.

What Is This? #12

```
rwfilter --type=out --
   start=2010/12/08 \
   --aport=22 --pass=ssh.rw

rwfilter --dport=22 ssh.rw \
   --pass=stdout | rwcut

rwfilter --sport=22 ssh.rw \
   --pass=stdout | rwcut
```

Outline

Introduction: SiLK

Network flow

Basic SiLK tools

Advanced SiLK tools

Summary

PySiLK—Using SiLK with Python

- PySiLK—an extension to Python
- Allows Python to manipulate SiLK's data files
- Uses the "silk" python module, from SEI CERT.

PySiLK example

```
#! /bin/env python
import silk
myfile = silk.SilkFile("MyFlows.rw", silk.READ)
for rec in myfile:
        if rec.sport < 2500 and rec.sport == rec.dport:
        print  ("%d %s %s %s" %
                (rec.sport, rec.stime, rec.sip, rec.dip))
```

Alternatives to PySiLK

- SiLK tools
  - Not as flexible criteria as Python.
  - Could use tuple files
    - Must be maintained
    - Aren’t self-contained with logic
    - Large tuple files run slower than Python.
- Text processing with Perl, C, or Java
  - Create text with rwcut delimited without titles
  - Convert ports back to integers
  - Dealing with dates, times, or addresses difficult

Modified example of PySilk

- Summarize the selection as a count by port
- Just keep a Python dictionary
  - Key = port number
  - Value = count

PySiLK advantages

- Speeds both programming and processing
  - Keeps data in binary, unlike Perl & C
    - No parsing text
  - Built-in conversions of objects to strings
  - Full power of Python
- Good for:
  - Stateful filters and output options
  - Integrate SiLK with other data types
  - Complex or branching filter rules
  - Custom key fields and aggregators for rwcut, rwsort

Outline

Introduction: SiLK

Network flow

Basic SiLK tools

Advanced SiLK tools

Summary

Furthering Your SiLK Analysis Skills (1)

Each tool has a **`--help`** option.

SiLK Reference Guide

SiLK Analysts' Handbook

- Both available at the SiLK tools website http://tools.netsa.cert.org

Email support

- silk-help@cert.org

Furthering Your SiLK Analysis Skills (2)

Tool tips

- SiLK Tooltips link on http://tools.netsa.cert.org

Flow analysis research and advanced techniques

- http://www.cert.org/flocon
- http://www.cert.org/netsa

Questions?

Contact Information

Ron Bandes — **rbandes@cert.org**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA

Network Security Monitoring in Minutes

Doug Burks

tcpdump -nnAi eth1 -s0 | grep -A5 "Doug Burks"

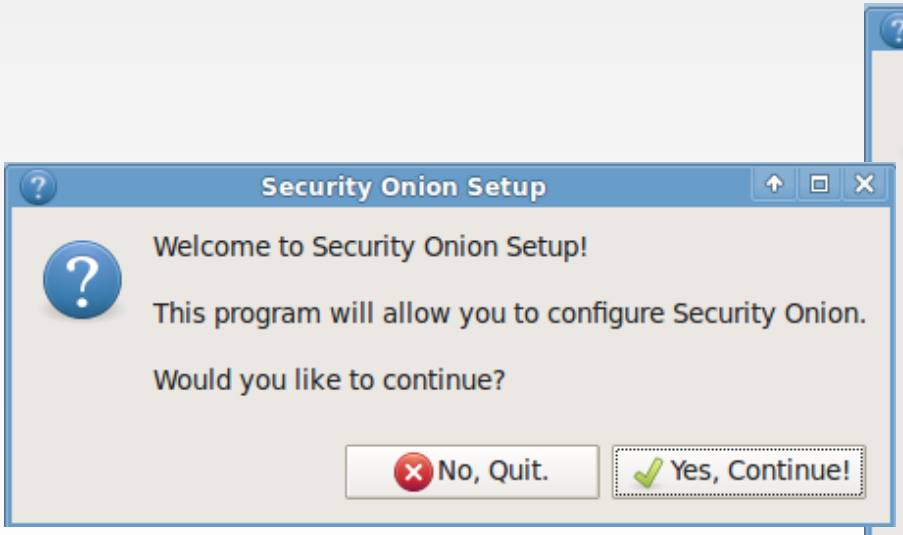
Doug Burks is:

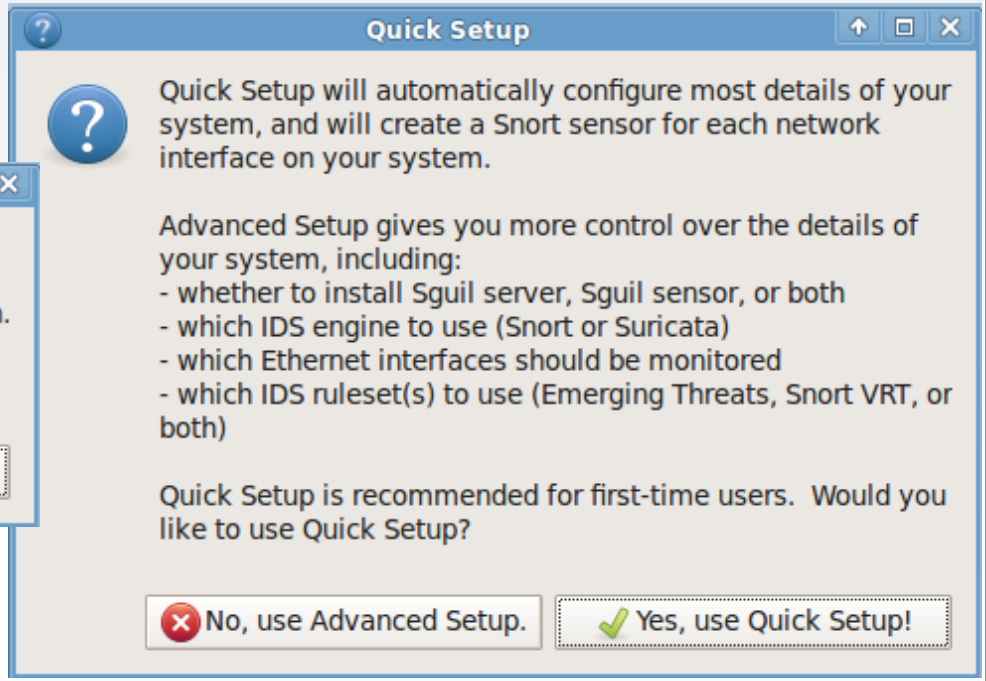
- Christian
- husband and father
- SANS GSE and Community Instructor
- Deputy CSO for Mandiant (we're hiring!)
- @dougburks #securityonion

Security Onion is a FREE Linux distro for Network Security Monitoring (NSM)

Next, Next, Finish for NSM

Setup wizard takes less than 5 minutes!





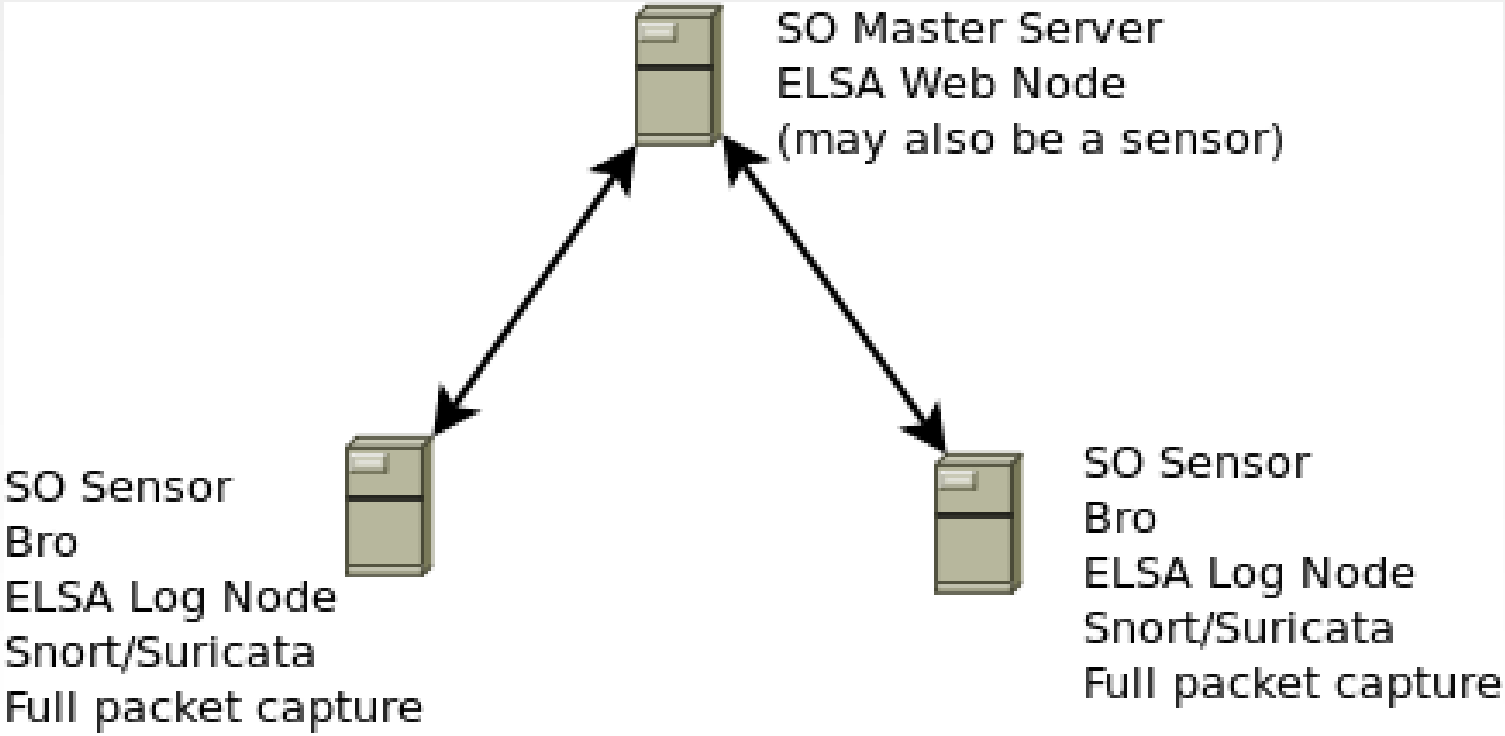
Network Security Monitoring: Data Types

- Alert data (NIDS alerts - Snort/Suricata, HIDS alerts - OSSEC)
- Asset data (Bro and PRADS)
- Session data (Argus, Bro, and PRADS)
- Transaction data (Bro protocol logs: http, ftp, dns, etc.)
- Full content data (netsniff-ng)

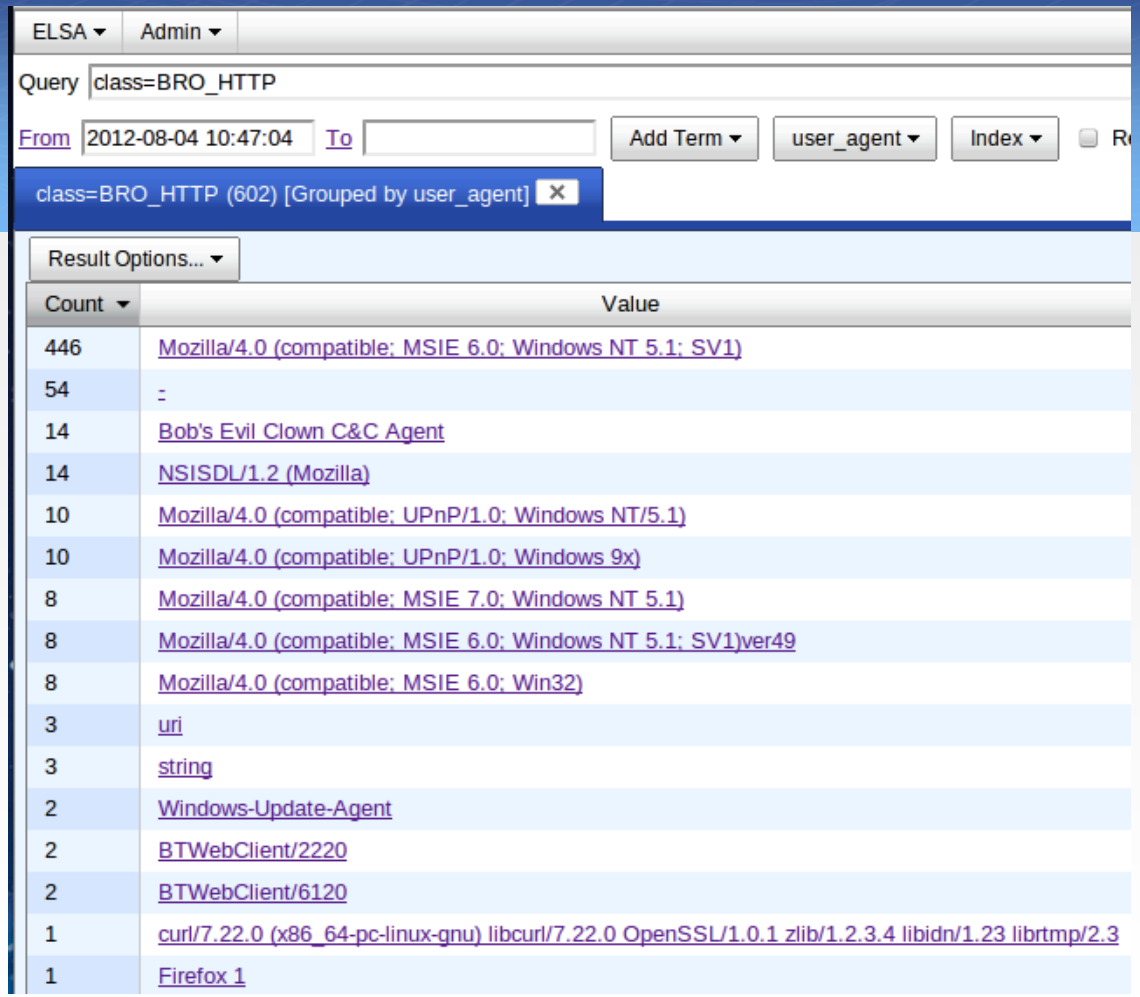
Analysis at Scale

- Download our ISO image (based on Xubuntu 12.04 **64-bit**)
  OR
  Start with your preferred flavor of Ubuntu 12.04 (Ubuntu, Kubuntu, Lubuntu, Xubuntu, or **Ubuntu Server**) 32-bit or **64-bit**, add our PPA and install our packages
- High performance:
  - Snort/Suricata/Bro running on **PF_RING**
  - Netsniff-ng uses **zero-copy** for high-speed full-packet capture
- ELSA (like a free version of Splunk) – **distributed** database with central web interface

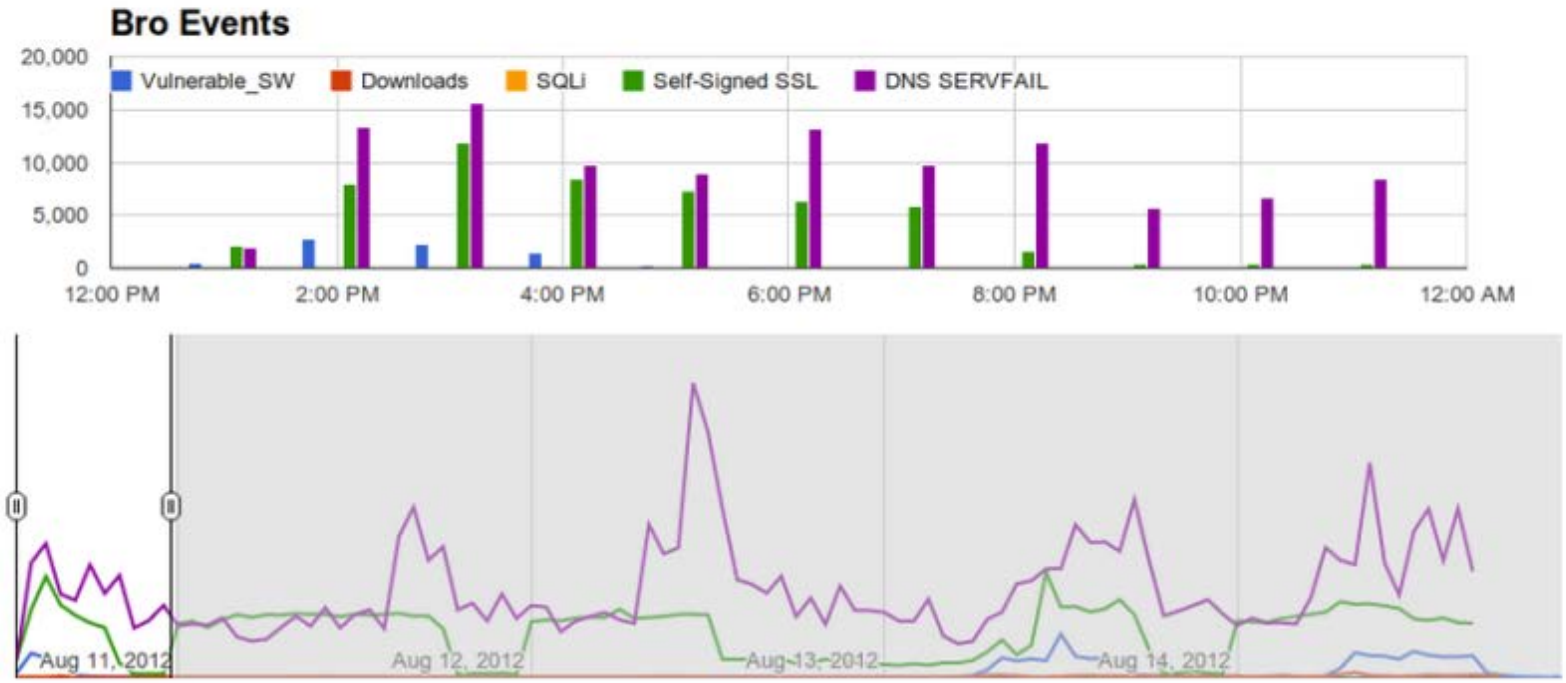
Distributed Deployment

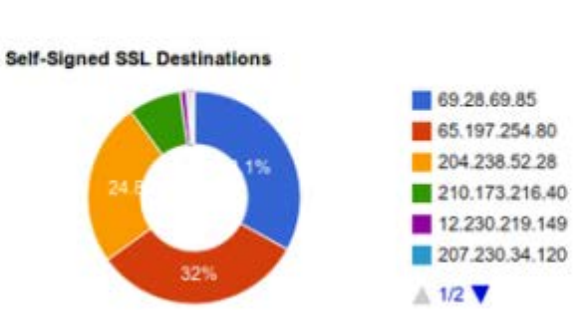


ELSA


ELSA
Admin
Query class=BRO_HTTP
From 2012-08-04 10:47:04
To
Add Term
user_agent
Index
class=BRO_HTTP (602) [Grouped by user_agent]
Result Options...
Count
Value
446 Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
54 -
14 Bob's Evil Clown C&C Agent
14 NSISDL/1.2 (Mozilla)
10 Mozilla/4.0 (compatible; UPnP/1.0; Windows NT/5.1)
10 Mozilla/4.0 (compatible; UPnP/1.0; Windows 9x)
8 Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
8 Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)ver49
8 Mozilla/4.0 (compatible; MSIE 6.0; Win32)
3 uri
3 string
2 Windows-Update-Agent
2 BTWebClient/2220
2 BTWebClient/6120
1 curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtmp/2.3
1 Firefox 1

Bro IDS





subject
emailAddress=admin@wiredsolar.net,CN=secure.wiredsolar.net,OU=IT,O=Wired Solar,L=Flagler,ST=Florida,C=US
CN=rsip.monitoredsecurity.com,OU=IT Security,O=Symantec Corporation,L=Northern
emailAddress=dhoover@centonline.com,CN=Dean Hoover,OU=Network Admin,O=Ce Berlin,ST=Wisconsin,C=US
ST=Tokyo,OU=Remote Service,O=RICOH COMPANY\,LTD.,L=Aoyama,C=JP,CN=G
CN=mcs1hkg.live.citrixonline.com,OU=Operations,O=Citrix Online LLC,L=Fort Lauder
C=CA
C=US,CN=mail.tytix.com
CN=TrustedSourceServer_IMQA01

Where do we go now?

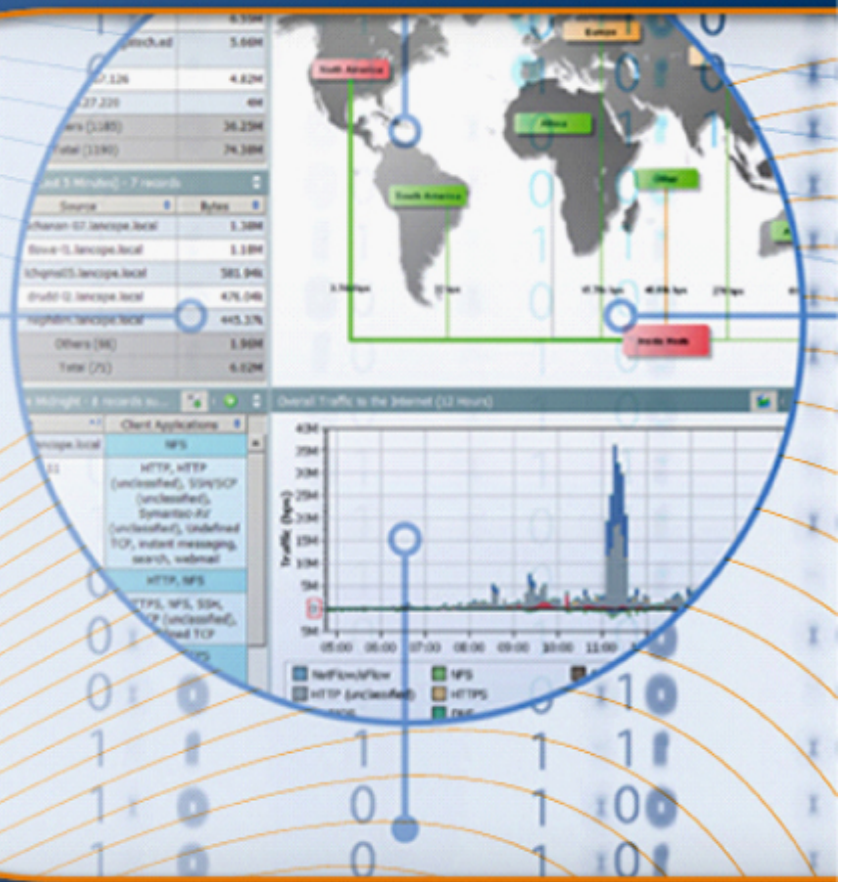
http://securityonion.blogspot.com

Updates are announced here and it also has the following links:

- Download/Install
- FAQ
- Mailing Lists
- IRC #securityonion on irc.freenode.net

Lancope
Network Performance + Security Monitoring™

KNOW YOUR NETWORK.
RUN YOUR BUSINESS.™



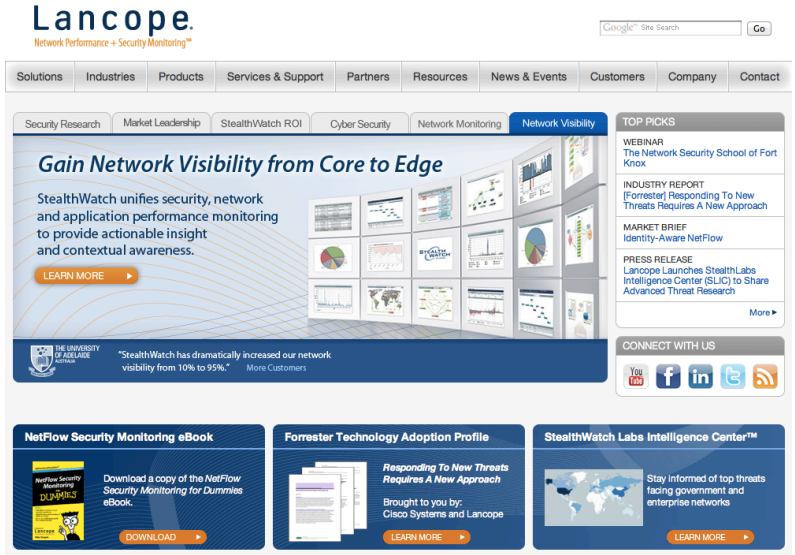
Insider Threat

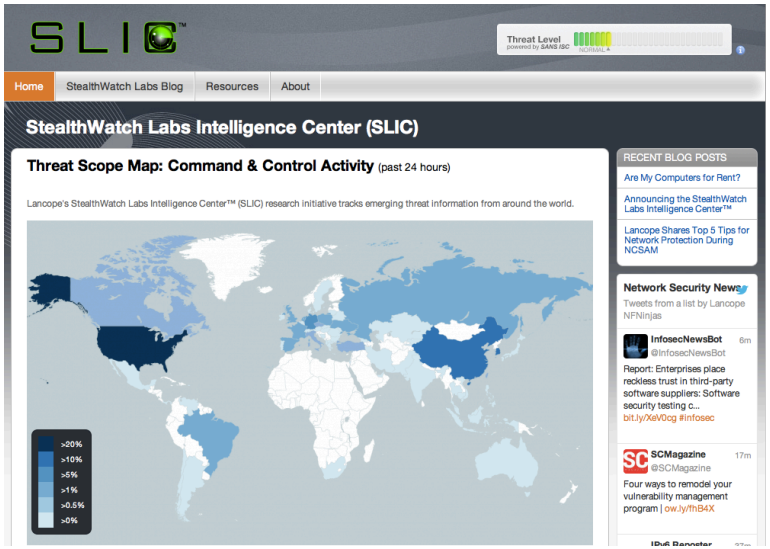
Tom Cross, Director of Security Research
tcross@lancope.com
(770) 225-6557

Lancope & StealthWatch

- Lancope specializes in Behavior-based Network Flow Analysis
- Detects attacks by baselining and analyzing network traffic patterns
- Excellent defense in depth strategy to aid in defense of critical assets
- Over 600 customers world-wide
- Founded in 2000, located in Atlanta, GA

http://www.lancope.com/SLIC/



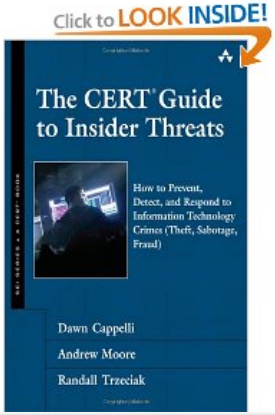


Why Insider Threat?

AlgoSec Survey of 182 IT Security Professionals

"What is the greatest risk your enterprise faces today?"

- Lack of Visibility – 28.7%
- **Insider Threat – 27.5%**
- Change Management – 24.1%
- **External Threat Actors – 19.6%**
  - Financially Motivated – 14%
  - Hacktivists – 5.6%



CERT Insider Threat Research

- 12 years of history
- Over 700 insider threat cases

- IT Sabotage
  - Average: $1.7 million
  - Median: $50,000

- IP Theft
  - Average: $13.5 million
  - Median: $337,000

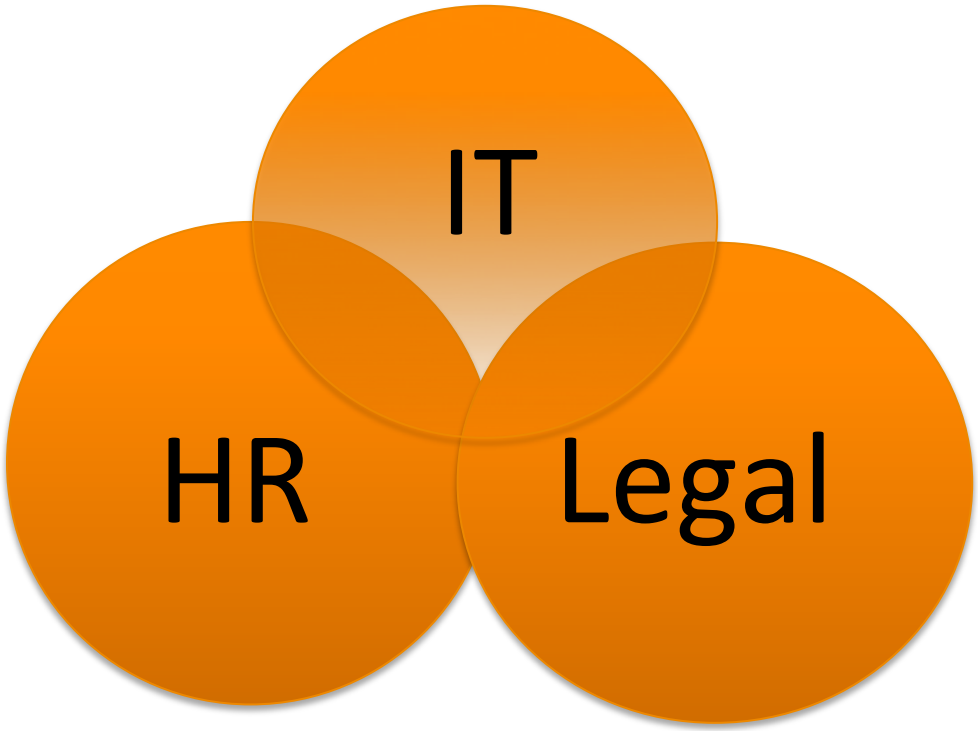
Why is Insider Threat mitigation challenging?

- The adversary may have authorized access to the systems that they intend to sabotage or the data that they intend to steal.

- Many organizations don't have control over where their sensitive data resides and from where it is and is not supposed to be accessed.

- Many organizations view the insider threat as a computer security problem, and they view computer security problems as problems for IT to resolve.

Combating Insider Threat is a multidisciplinary challenge



- IT cannot address insider threat by itself
  - People have a tendency to think that IT is solely responsible for all computer security issues.
- Legal: Are policies in place? Are they realistic? Does legal support IT practices?
- HR: Who is coming and going? Who has workplace issues? Are there soft solutions?
- IT: Is the privacy of end users adequately protected?
- What impact on workplace harmony are policies, monitoring, and enforcement having?
- Are you applying policies consistently?

Who commits insider attacks?

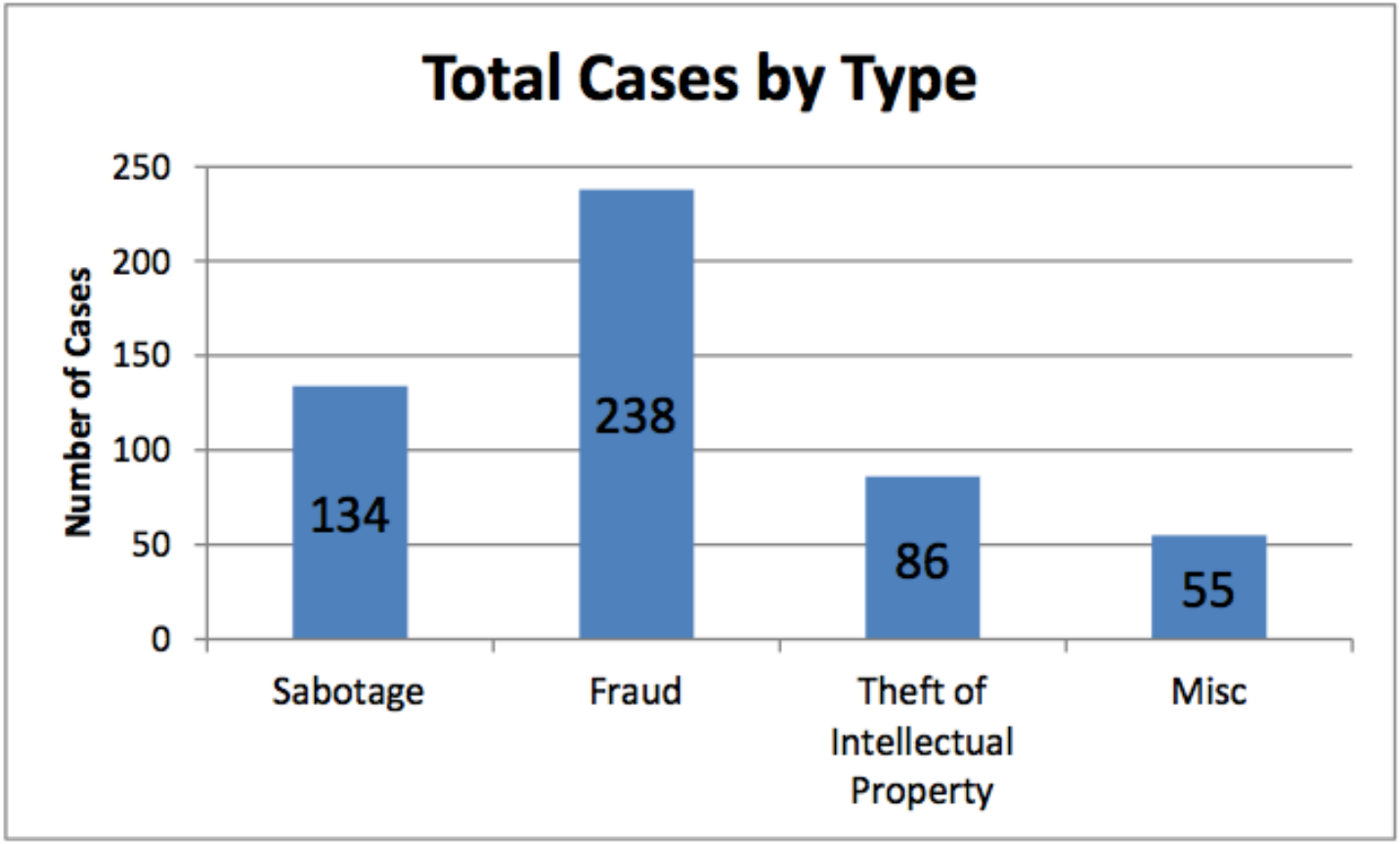


Figure 1: Number of Cases in the CERT Insider Threat Database by High-Level Category (Excluding National Security Espionage Cases)

Source: *Insider Threat Control: Using Centralized Logging to Detect Data Exfiltration Near Insider Termination* - CERT

CERT: Common Sense Guide to Prevention and Detection of Insider Threats

	IT Sabotage	Financial Gain	Business Advantage
% of cases:	45%	44%	14%
Employment:	Former	Current	Current
Position:	Technical	Data Entry & Customer Services	Technical or Sales
Authorized Access?	Rarely	75%	88%
Used their own credentials?	30%	85%	Almost always
Compromised an account?	43%	10%	Rarely
Attack was non-technical:	65%	84%	Almost always
When:	After hours	Normal hours	Normal hours
Where:	Remote	Local	Local
IDed due to:	Logs	Logs	Logs

An Observation

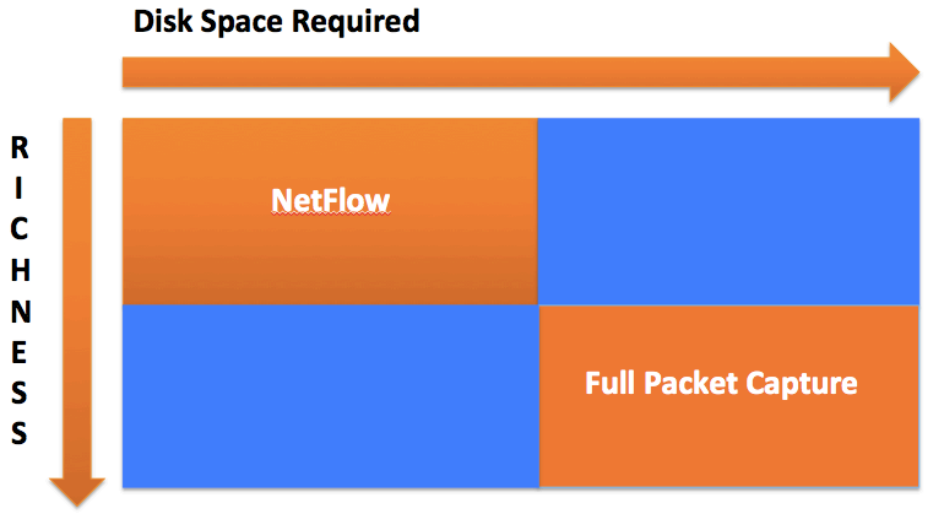
- Monitoring can deter malicious insiders
  - Common Assumption: If we can evade a security control, that control is worthless.
    - Evasions of technical controls can be automated and globally distributed.
    - Deterrence doesn't work on the Internet because attribution doesn't work on the Internet.
  - We don't apply this assumption in the world of physical security.
  - Knowledge that events are being logged and the logs are archived and monitored creates a risk for insiders unless they can modify the logs.
  - The use of fully automated analysis creates thresholds that insiders can evade.
  - A hybrid approach where automated tools help human analysts avoids creating a scenario where an attacker can know that activity won't be discovered

Sources of visibility

- Firewall logs
  - Are you logging everything or just denies?
- Internal & Host IPS systems
  - HIPS potentially has a lot of breadth
  - Can be expensive to deploy
  - Signature based
- Log Management Solutions/SIEM
  - Are you collecting everything?
  - You can only see what gets logged
- Netflow
  - Lots of breadth, less depth
  - Lower disk space requirements
- Full Packet Capture
  - Deep but not broad
  - Expensive
  - High disk space requirements

Tradeoffs:

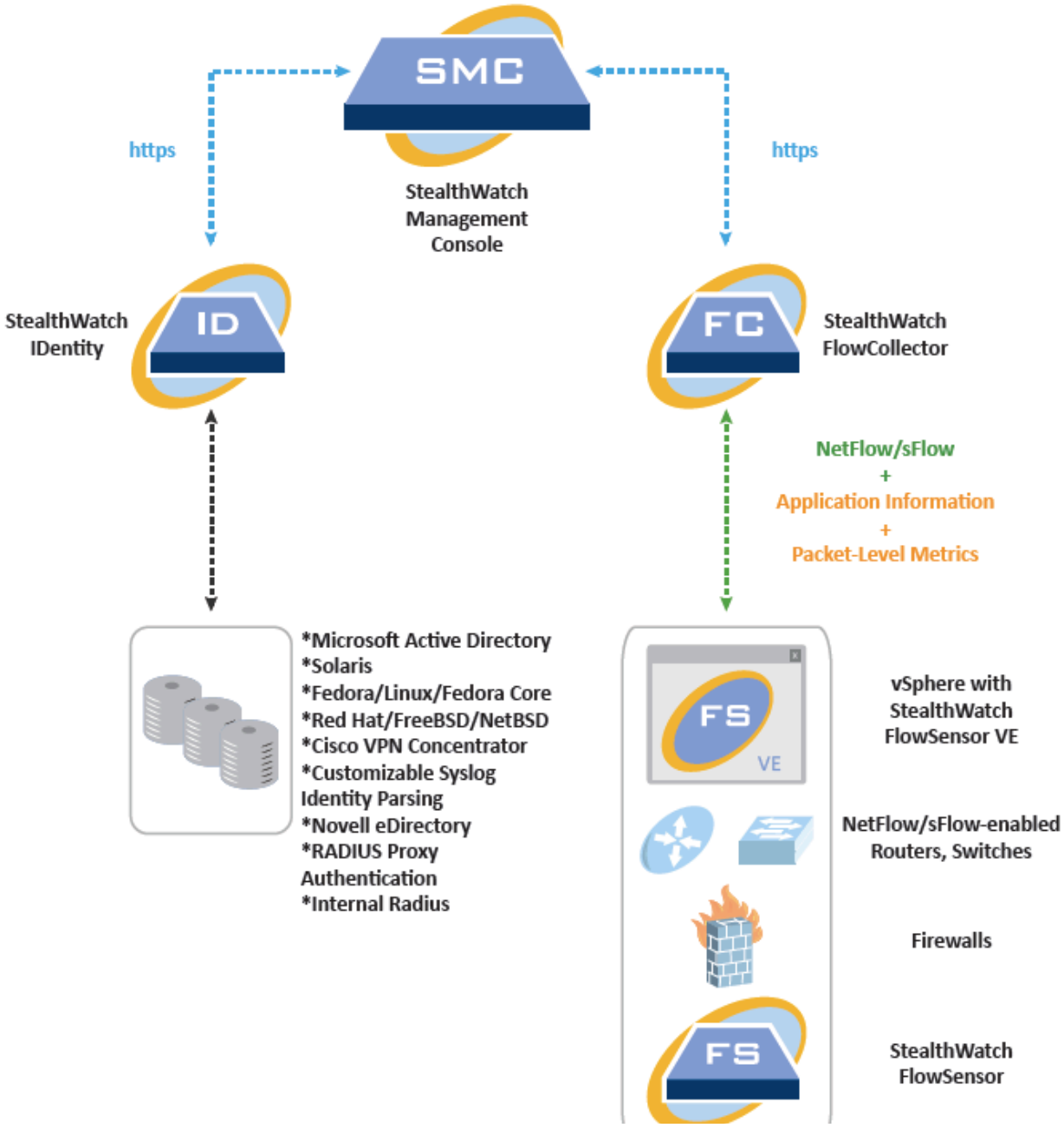
- Record everything vs only bad things
- Breadth vs Depth
- Time vs Depth
- Privacy



Utility of Netflow

- Netflow can provide a long term audit trail
  - Most malicious insiders are identified due to logs and audit trails
  - Months may pass before you realize that something has happened
- Netflow can provide economical visibility into internal network traffic at leaf nodes
  - This is critical for identifying insider activity
- Challenge: Network needs to be coupled with identity information
  - Many insider attacks can be identified with identity information alone
  - A given user's IP address changes regularly

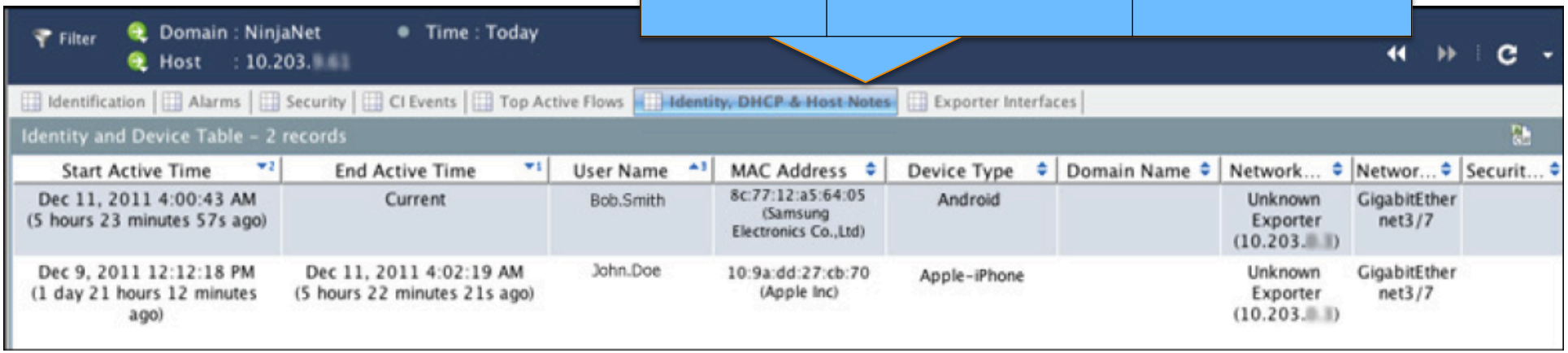
Lancope Identity 1000



Cisco Identity Services Engine (ISE)

- Cisco ISE is a context aware, policy based 802.1x authentication solution
- Detect
  - Device type, operating system and patch level
  - Time and location from which user attempting to gain access

User Name	MAC Address	Device Type
Bob.Smith	8c:77:12:a5:64:05 (Samsung Electronics Co.,Ltd)	Android
John.Doe	10:9a:dd:27:cb:70 (Apple Inc)	Apple-iPhone



Lancope

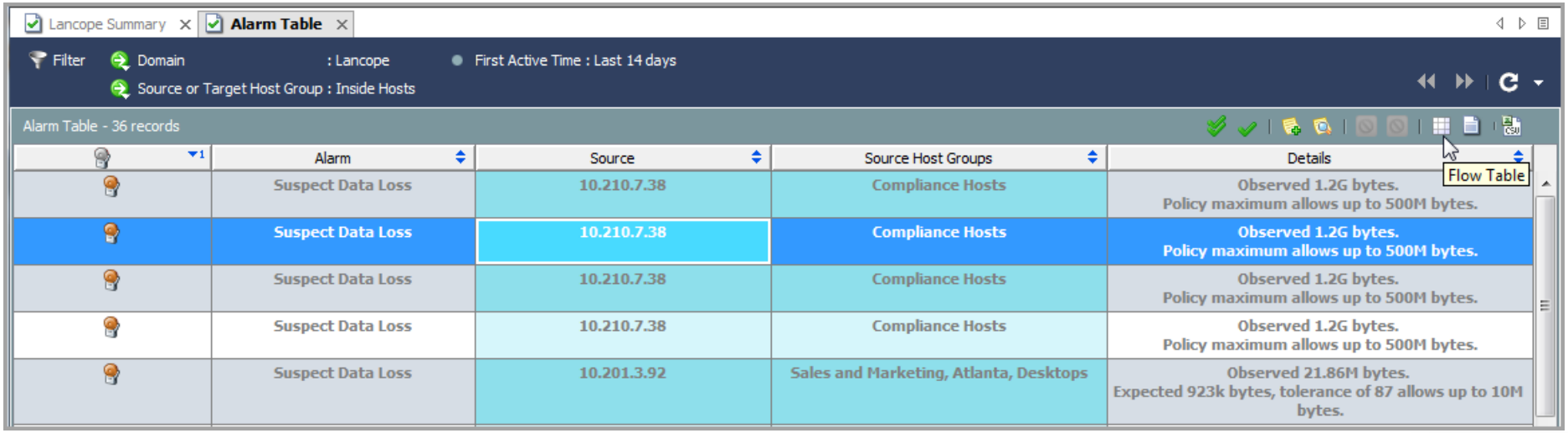
Stitching Identity Information to Netflow

- ISE gives only the endpoint address whereas ID 1000 (AD) gives all authentication accesses though out the environment
- StealthWatch associates every flow from a particular source IP address with every identity authenticated to that IP address when the flow started
  - Multi user systems become unwieldy
- With ISE every flow associated with an identity is the responsibility of the associated user.
- With ID 1000 (AD), some shared resources need to be eliminated from consideration.

Theft of Intellectual Property

- 54% of CERT's exfiltration cases occurred over the network (most email)
  - The network may still be used to collect even when it isn't used to exfiltrate
- Key window – 30 days before and after resignation/termination
- Email with large attachments to third party destinations
- Data Infiltration and Exfiltration
- Large amounts of traffic with key systems
  - Printer
  - Source code repository

Device Used	Number of Incidents
Copier	1
Fax	3
Printer	30
Scanner	2



Lancope

Fraud Detection

- Tight roll based access control
- Auditing of database access and modifications
- Having checks and balances in your processes

- Logins coming from another user's machine
  - Different user logins to different systems from the same address
  - Unusual device or source IP
- Network behavioral profiling
  - Most effective in environments where system configuration and use are static and homogeneous

IT Sabotage Detection

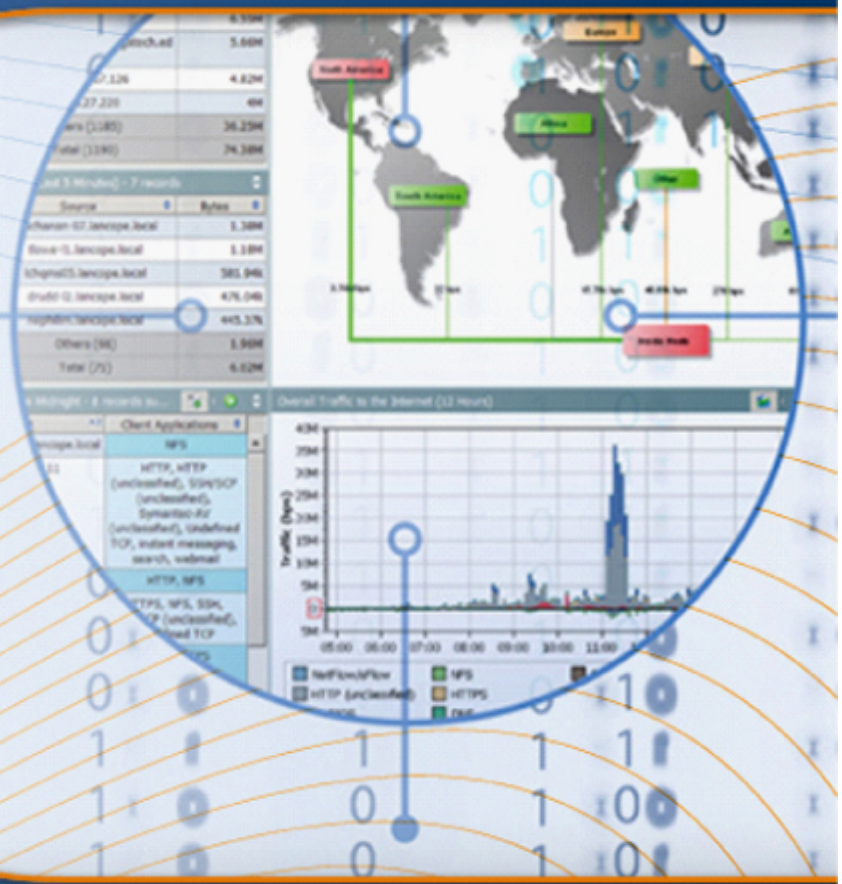
- Targeted monitoring of employees who are “on the HR radar”
  - These attackers often login from remote after hours using a system account or compromised user account
- With Identity only
  - Unusual Access Times (Could be any account)
  - Unusual Access Device or Source Address
  - Account creation near termination (who is new?)
  - Mismatch between logins on different systems
- With Netflow
  - Scanning & Brute Force Activity
  - Active sessions at time of termination
    - CERT indicated that many organizations don’t have a process for checking this
  - Access to unusual internal destination hosts (coupled with other factors)

Profiling Anomalous Insider Behavior

- Over 6 months, collect:
  - Internal system/service combinations that a user accessed
  - Bytes to/from
  - Session lengths
  - Number of accesses
  - Summarize dynamic address ranges
- Eliminate peaks from the dataset
  - Longest Session
  - Largest Transfer
- Compare with the final 30 day window:
  - Are there any new internal systems/services that were accessed?
  - Are there traffic levels or session times that are outside of the peak ranges?
  - Are there any resources that were accessed more frequently?
  - It might be useful to aggregate bytes per system/service across each month

Lancope
Network Performance + Security Monitoring™

KNOW YOUR NETWORK.
RUN YOUR BUSINESS.™



Thank You

Tom Cross
Director of Security Research
tcross@lancope.com
(770) 225-6557

ENDGAME

John Munro / jmunro@endgame.com
Jason Trost / jtrost@endgame.com

FlonCon 2013 | January 7–10 | Albuquerque, New Mexico

Introductions

ENDGAME

- John Munro (jmunro@endgame.com)
  - Network Security Researcher and Data Scientist
- Jason Trost (jtrost@endgame.com)
  - Senior Software Engineer
  - Specializes in Hadoop/Storm/BigData

Agenda

ENDGAME

- The Problem
- Our Approach
- DGA Domain Classifier
- String Statistics as Features
- Malicious Domain Classifier
- Demo
- Real-time Streaming Platform

The Problem

ENDGAME

txmxbo.info

youtube.com

yahoo.com

Ct0u2xj5dbe4.wvw—game465.com

p4.httzd5e2ufizo.3bawhfuec45dca65.401724.s1.v4.ipv6-exp.l.google.com

abulqe.com

za6.limfoklubs.com

ns3.ohio.gov

bibz01.apple.com

docs.joomla.org

Wmk41035u3751s0bgv4n91b0b7h74v.ipcheker.com

The Problem

ENDGAME

txmxbo.info

youtube.com

yahoo.com

Ct0u2xj5dbe4.wvw—game465.com

p4.httzd5e2ufizo.3bawhfuec45dca65.401724.s1.v4.ipv6-exp.l.google.com

abulqe.com

za6.limfoklubs.com

ns3.ohio.gov

bibz01.apple.com

docs.joomla.org

Wmk41035u3751s0bgv4n91b0b7h74v.ipcheker.com

The Problem

ENDGAME

- Massive Volumes
  - Some of our partners deal with TBs per day of DNS PCAPs
- Incredible Rates
  - One partner sees 13k requests/sec
  - Another closer to 100k/sec

Our Approach: Machine Learning!

ENDGAME

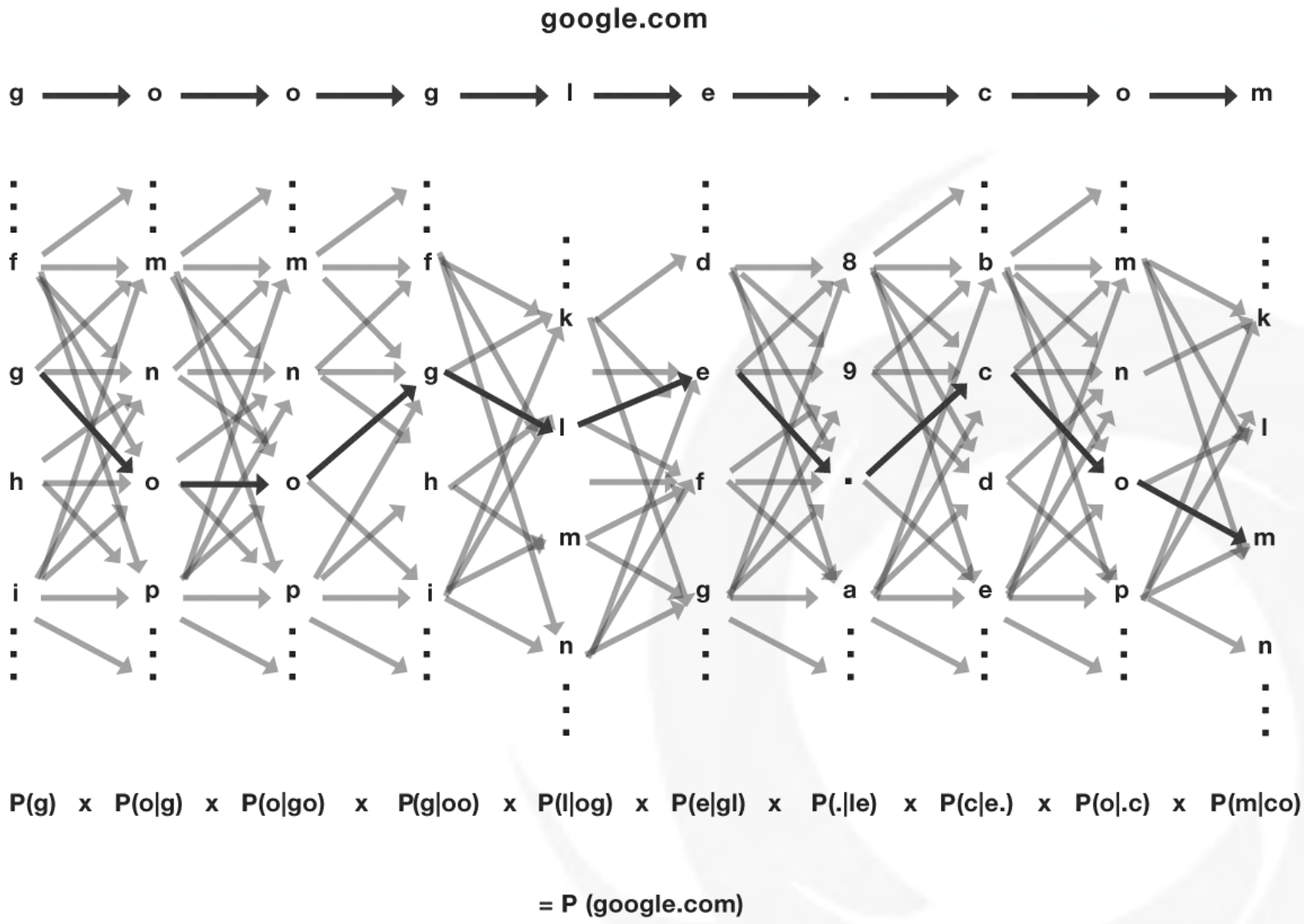
- Real-time streaming classification
  - In parallel across multiple servers
- Markov Models
  - Random Domain Generation Traffic
  - Normal Benign Traffic
- Random Forests
  - Benign vs Malicious
- Periodically retrained
  - In order to maintain accuracy

Data Sources

ENDGAME

- Benign Domains
  - Millions of popular, real domains
    - Correlated with the Alexa top 10k domains
- Malicious Domains
  - 800k domains gathered from an internal malware sandbox
  - Public blacklist domains from Conficker and Murofet Botnets

Markov Models



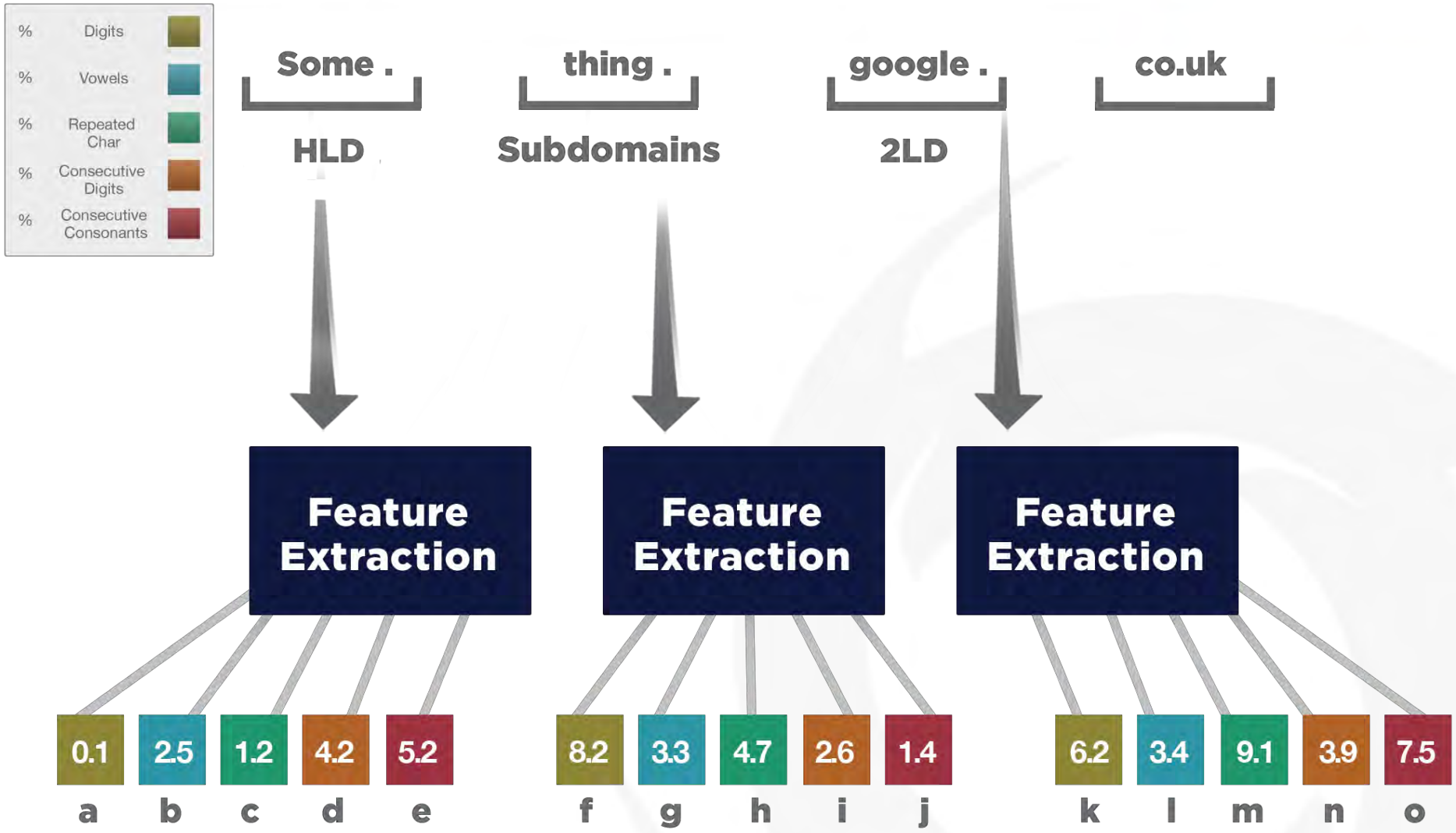
Markovian DGA Classifier

ENDGAME

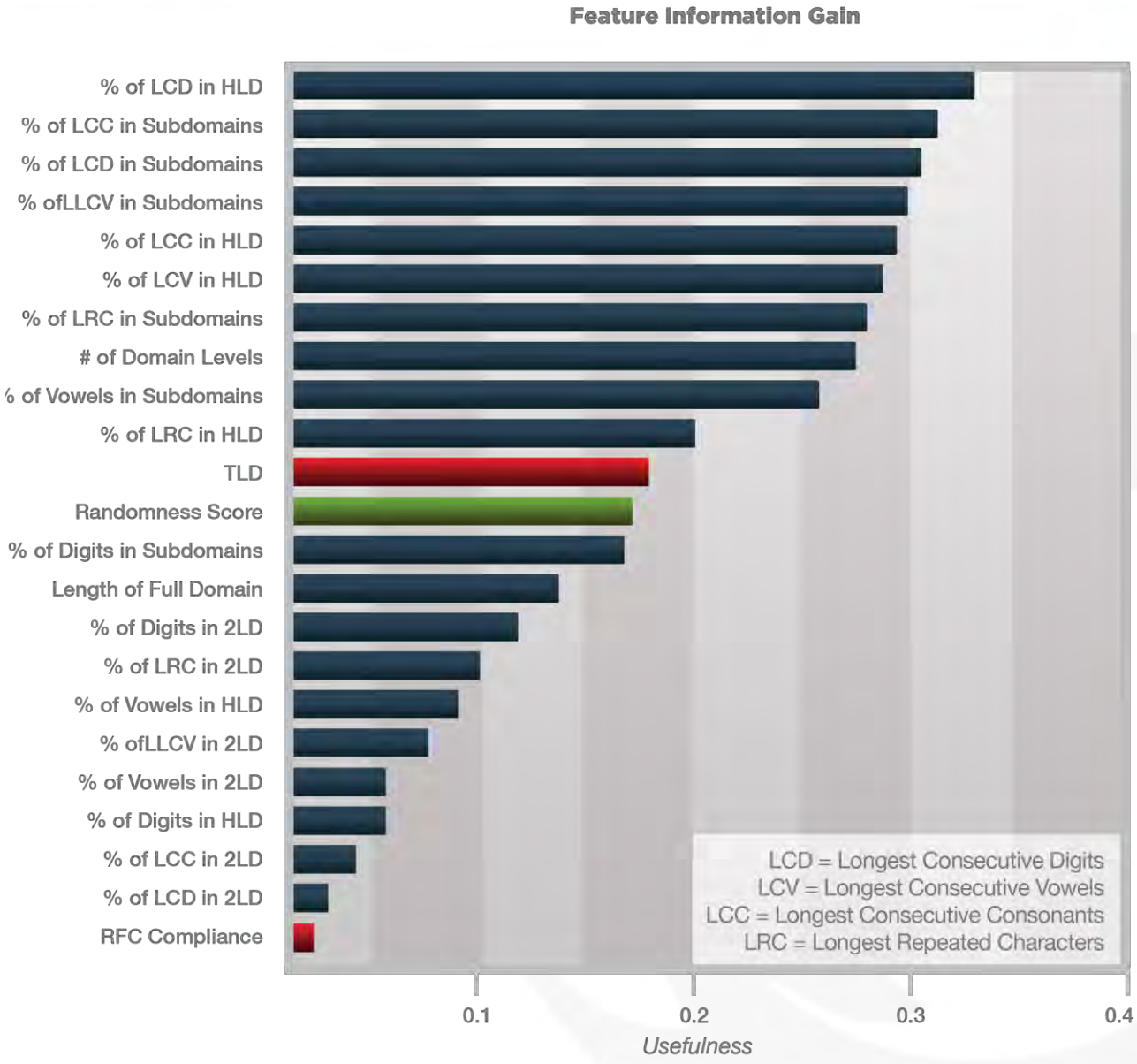
- Domain Generation Algorithm (DGA)
- Popular Domain Model
  - Trained: 258,039 domains from Day 1 of our Benign set
  - Tested: 331,359 domains from Day 2 of our Benign set
  - Accuracy: 99.40 % with 1,458 Unknown
- Randomly Generated Domain Model
  - Trained: 90,884 domains from Conficker Botnet
  - Tested: 295,306 domains from Murofet Botnet
  - Accuracy: 99.34 % with 1,923 Unknown

String Statistics as Features

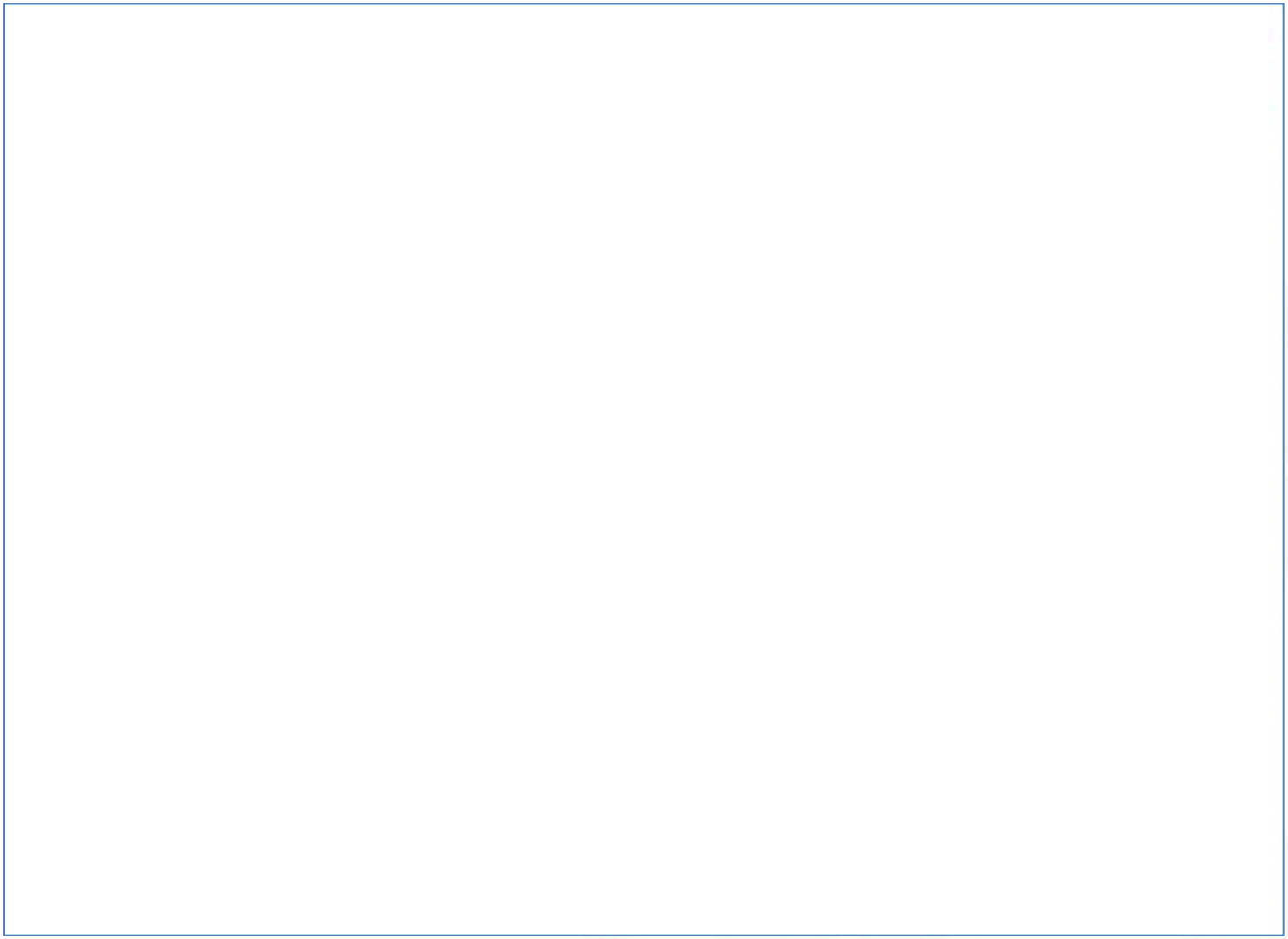
ENDGAME



Feature Usefulness



Random Forests Algorithm



Random Forests

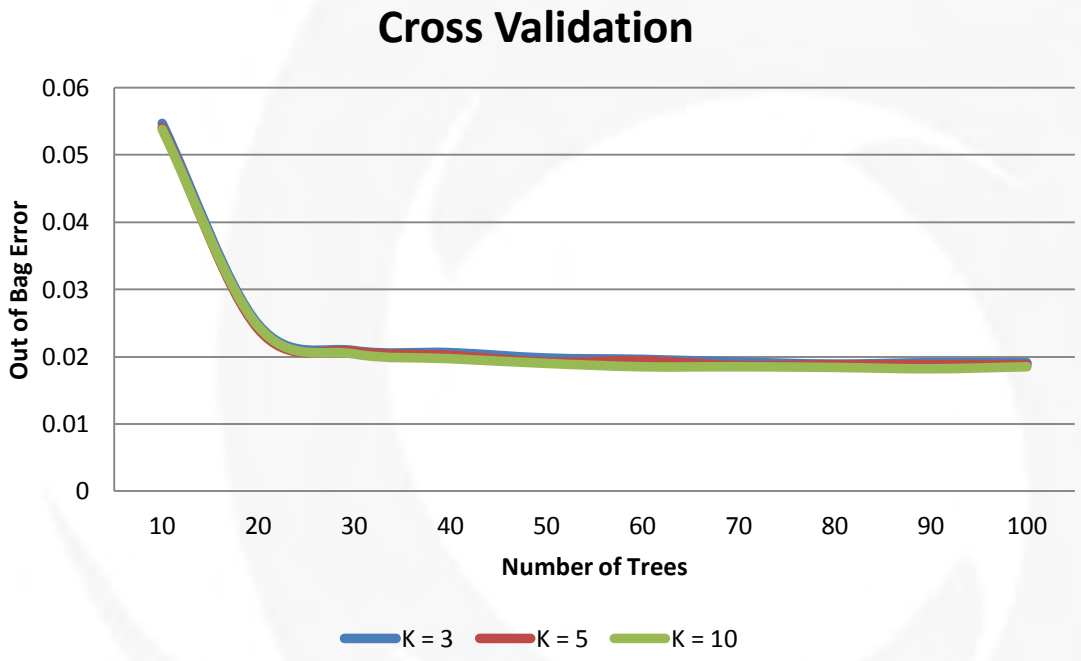
ENDGAME

- Pros:
  - Very high accuracy
  - Scalable across many nodes
  - Built-in protection from over fitting
  - Can handle very large data sets with many features
  - Robust with respect to goodness of features
  - Practical for real world use
  - Does not assume a distribution
  - Only two parameters to tune
  - Memory efficient
- Cons:
  - Not the quickest classifier, but plenty fast in practice

Malicious Domain Classifier

ENDGAME

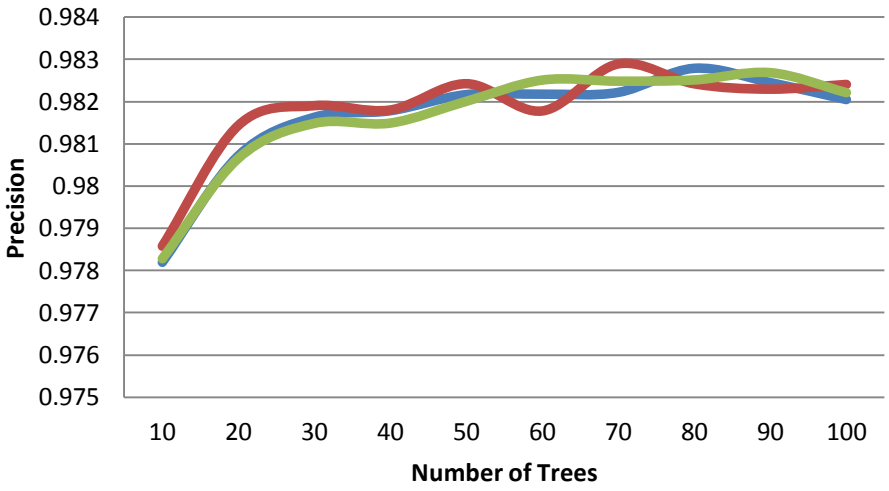
- Performance measured by 10 – fold Cross Validation
- Training Set
  - 200k Benign
  - 200k Malicious



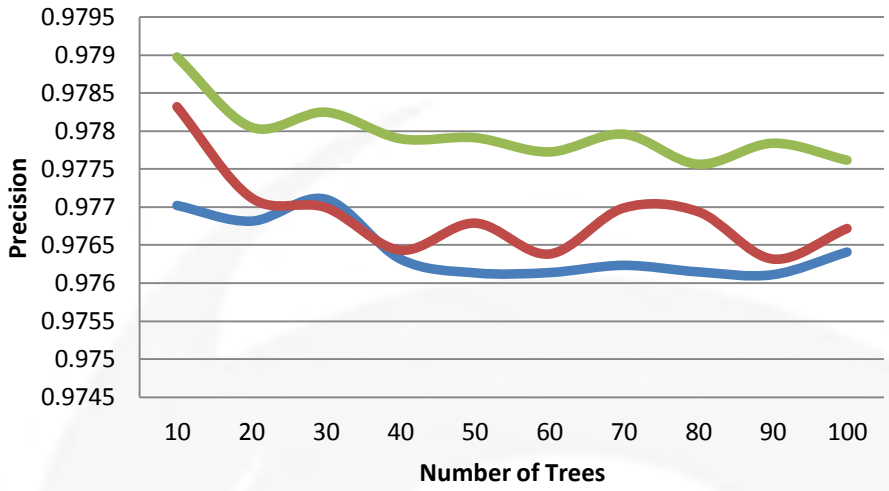
Results

ENDGAME

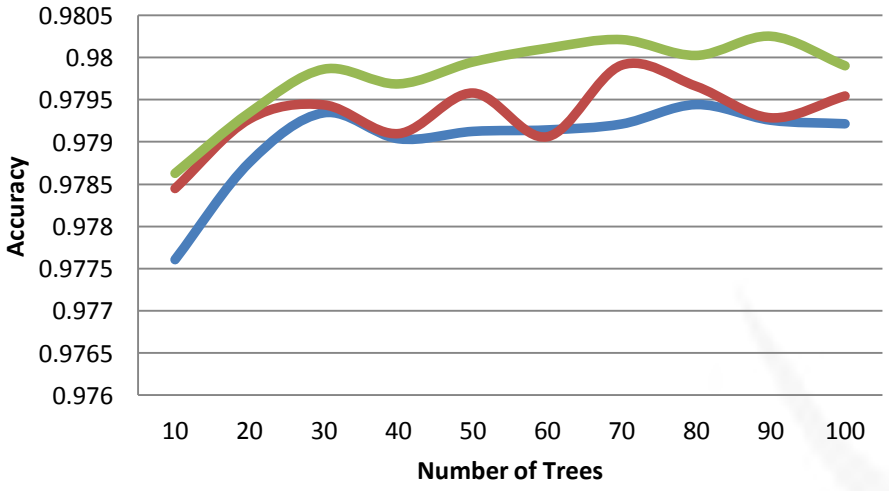
Bad Precision



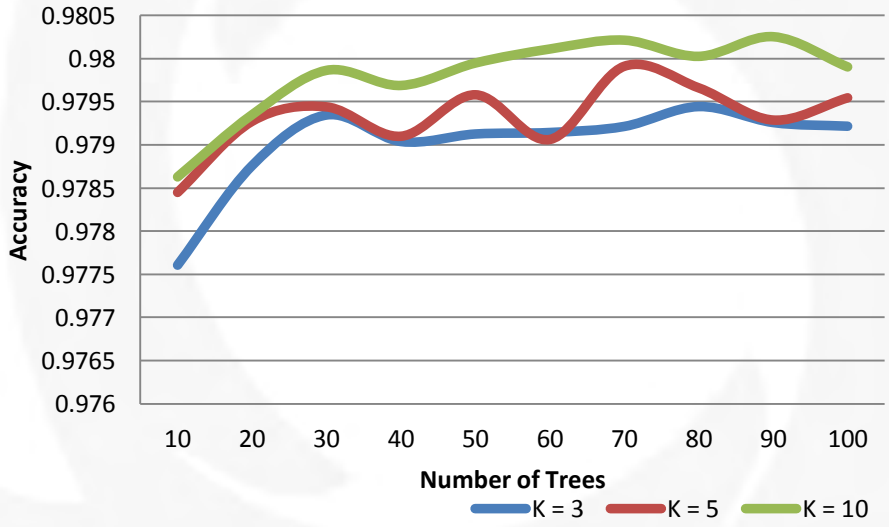
Good Precision



Bad Accuracy

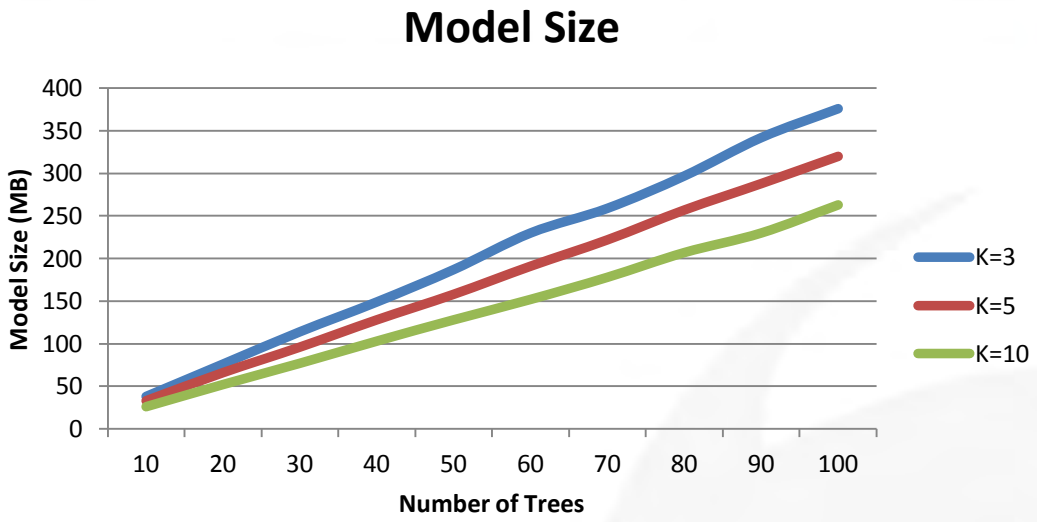


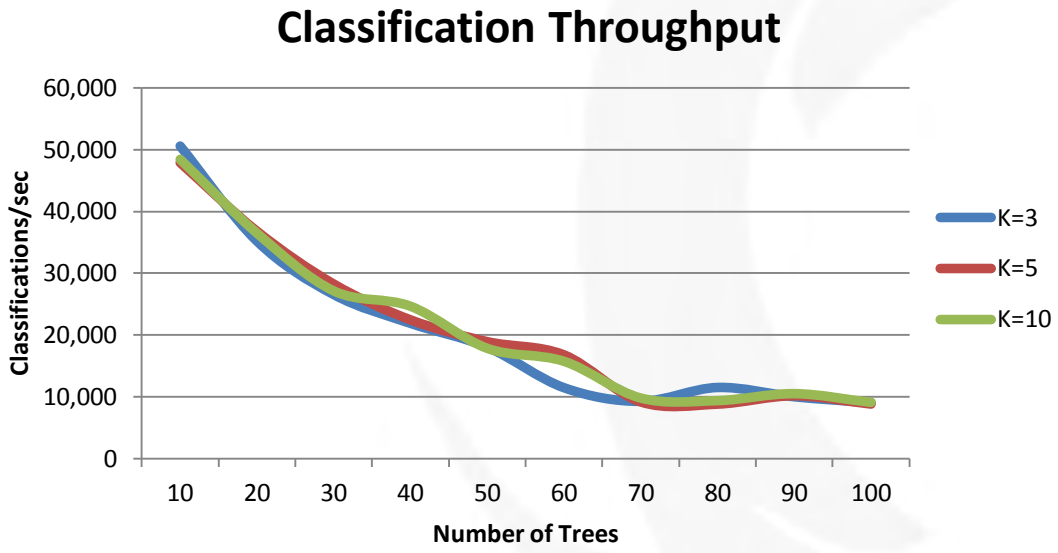
Good Accuracy



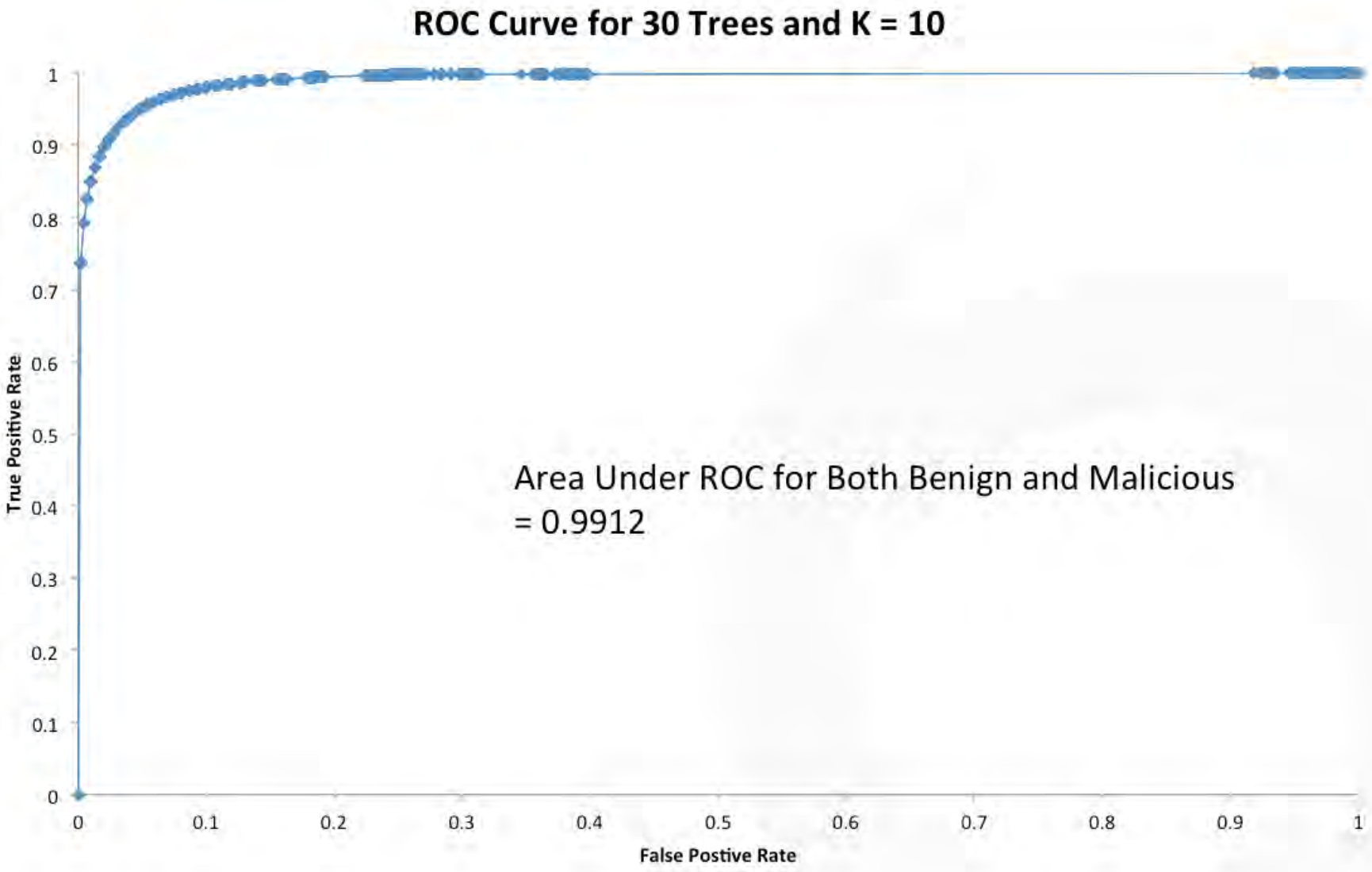
Results

ENDGAME





Results


ROC Curve for 30 Trees and K = 10
True Positive Rate
False Postive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Area Under ROC for Both Benign and Malicious
= 0.9912

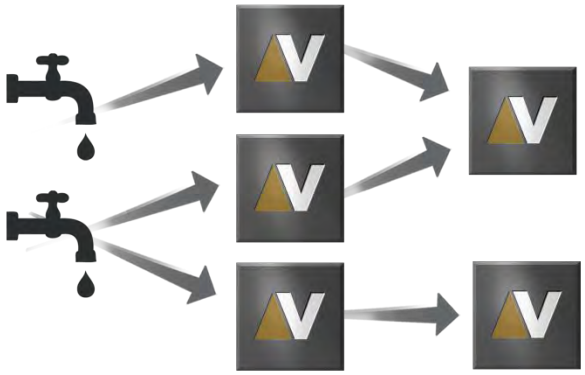
Demo

ENDGAME

Realtime Streaming Platform

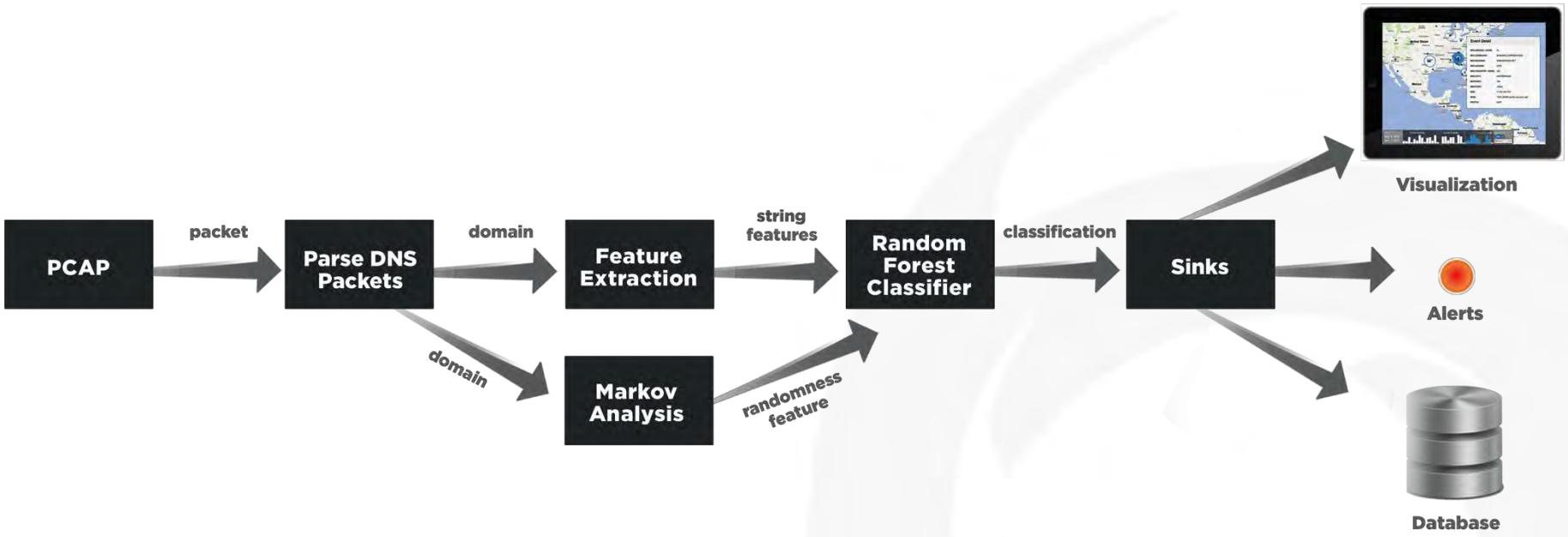
ENDGAME

- *Velocity* is a platform for processing, analyzing, and visualizing large-scale event data in real-time
- It was designed to be horizontally scalable and is built using Twitter's Storm



- It was built primarily for internal use with DNS events, IDS alerts, and netflow data, but it is in the process of being commercialized

Velocity Pipeline



Conclusion

ENDGAME

- Malicious domain classification
- DGA domain identification using Markov Models
- Summary Statistics based on domain string work well
- Random Forests are very successful at classifying domains as Benign or Malicious
- Real-time, distributed implementation

Future Work

ENDGAME

- Include more features: TTL, frequency seen, etc.
- Correlation of bad domains based on ASN, Country, Organization, etc.
- Identify subnets that are infected based on high traffic to bad domains
- Identify Content Delivery Networks
- Self Organizing Maps and other visualizations

Questions

ENDGAME

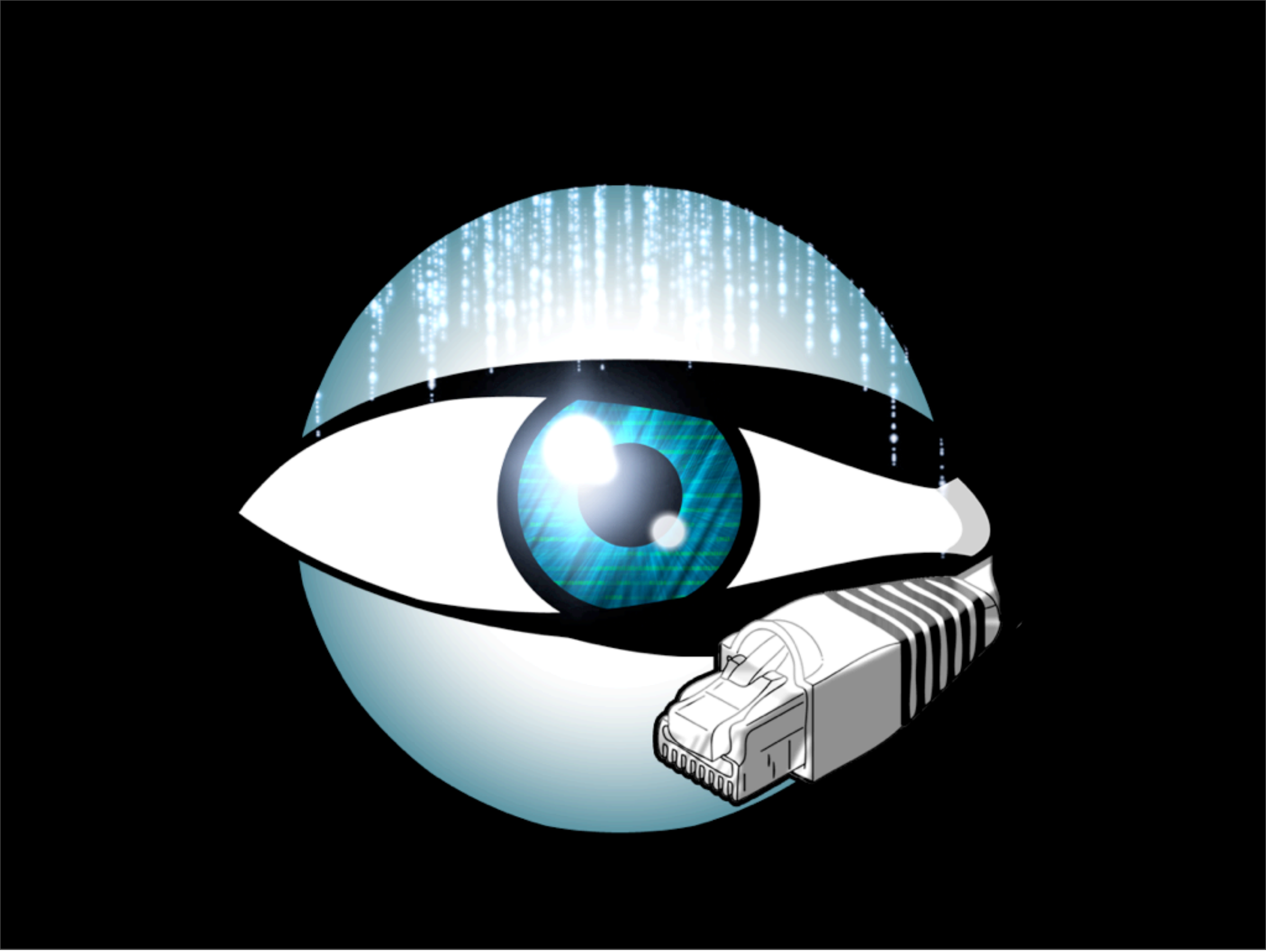
Contact Information

- John Munro
- Email: jmunro@endgame.com

- Jason Trost
- Email: jtrost@endgame.com
- Twitter: @jason_trost
- Blog: www.covert.io

Bro for Real-Time Large-Scale Understanding

Seth Hall
International Computer Science Institute



About Me

- Incident response at The Ohio State University for 7 years. 2 extra as a student.
- Spent quite a few years sitting and running flow-tools searches.
- OSU -> GE
- GE -> ICSI (International Computer Science Institue)

No NetFlow?

- NetFlow analysis served well at OSU for many years.
- Year over year degradation of detection capability with NetFlow.
- IRC Botnets were the first decline.

Bro

- Real time event analysis language and platform with protocol analysis.
- $3 million NSF grant for engineering work.
- Strong focus on usability and capability while fixing many, many bugs.
- 84th most popular programming language on GitHub.

Metrics Framework

- Return to measurement roots with new abstractions.
- Programmatic interface for measurement.

Metrics Framework Motivations

- Load balancing made previous techniques all fail. Metrics framework hides cluster abstraction.
- Better and more repeatable interface and approach for measurement and thresholding.
- Give more people the ability to write real world deployable measurement scripts.

Metrics Approach

- Discrete time slices (still investigating sliding window).
- Only streaming algorithms allowed.
- Every measurement must be merge-able for cluster support.
- Probabilistic data structures coming.

Why do any of this?

Measurement is fun!

Lavender Martini Border Gateway Protocol

Facts about Justine's TCP Sessions

I've been doing some self-experimentation this week, running tcpdump persistently and logging the results to a (huge) directory of pcap files. Here's a taste of 24 hours of the data...

Length of time in seconds:

MAX: 62182.238625 <— ssh
MIN: 1.7e-05 <— probably a bug?
MED: 2.046867
AVG: 60.9999384359015
STD. DEV: 736.853599230884

Same numbers for just HTTP/Port 80 traffic:

MAX: 5607.822474 <— seriously?
MIN: 0.003673
MED: 1.12272
AVG: 57.0826988090778
STD. DEV: 184.986012558513

Replicate those results!

```
event bro_init()
    {
    Metrics::add_filter("conns.originated",
                        [$every=1hr,
                         $measure=set(Metrics::AVG, Metrics::MAX, Metrics::MIN,
                                      Metrics::VARIANCE, Metrics::STD_DEV),
                         $period_finished(ts: time, metric_name: string, filter_name: string, data: Metrics::MetricTable) =
                            {
                            for ( index in data )
                                {
                                local val = data[index];
                                print fmt("Connection duration stats for %s", index$host);
                                print fmt("        Period from %s to %s", strftime("%F-%R", val$begin), strftime("%F-%R", val$end));
                                print fmt("        Number of conns: %d", val$num);
                                print fmt("        Max: %.2f", val$max);
                                print fmt("        Min: %.2f", val$min);
                                print fmt("        Std dev: %.2f", val$std_dev);
                                print fmt("        Average %.2f", val$avg);
                                }
                            }
                         ]);
    }

event connection_state_remove(c: connection)
    {
    if ( c$id$orig_h == 192.168.1.105 )
        Metrics::add_data("conns.originated", [$host=c$id$orig_h], [$dbl=interval_to_double(c$duration)]);
    }
```

Connection duration stats for 192.168.1.105
Period from 2009-11-18-13:34 to 2009-11-18-14:26
Number of conns: 77
Max: 3188.27
Min: 0.00
Std dev: 362.48
Average 54.79
Connection duration stats for 192.168.1.105
Period from 2009-11-18-14:34 to 2009-11-18-14:47
Number of conns: 19
Max: 1183.11
Min: 0.00
Std dev: 263.29
Average 67.83
Connection duration stats for 192.168.1.105
Period from 2009-11-18-19:00 to 2009-11-18-19:20
Number of conns: 81
Max: 30.11
Min: 0.00
Std dev: 7.08
Average 3.58

Other Detections using the Metrics Framework

200.29.31.26 had 349 failed logins on 2 FTP servers in 14m47s

92.253.122.14 scanned at least 29 unique hosts on port 445/tcp in 1m4s

88.124.212.10 scanned at least 41 unique hosts on port 445/tcp in 1m13s

212.55.8.177 scanned at least 75 unique hosts on port 5900/tcp in 0m36s

200.30.130.101 scanned at least 66 unique hosts on port 445/tcp in 1m20s

107.22.92.186 scanned at least 64 unique hosts on port 443/tcp in 0m1s

5.254.140.123 scanned at least 29 unique hosts on port 102/tcp in 4m1s

122.211.164.196 scanned 15 unique ports of host 75.89.37.60 in 0m5s

Coming in Bro 2.2

Thanks!

- seth@icir.org
- @remor on Twitter
- http://www.bro-ids.org
- info@bro-ids.org
- @Bro_IDS on Twitter

Flow Analysis Using MapReduce

Strengths and Limitations

Markus De Shon
Sr. Security Engineer

Agenda

Google

- MapReduce
  What is it?
- Case Study
  Entropy Timeseries
- Scaling MapReduces
- Other thoughts, Conclusions

MapReduce: What is it?

Google

A parallel computational method

3 stages

- Map: Apply function(s) to each record, compute a sharding key
- Shuffle: Group data by sharding key
- Reduce: Apply function(s) to records for each key

Optimal for trivially parallelizable problems

- Our problems sometimes are, sometimes not...

Shuffle phase

Google

This is where the magic happens...

Transport

- Locally: localhost sockets
- Different host: RPC of protocol buffer over TCP socket

There is no free lunch (e.g. count distinct)

- How is data distributed among input shards?
- Ideally, key by input shard (e.g. input filename), but any non-trivial shuffle will defeat this
- Try to optimize (number of keys * number of emits per key)

Case study: Entropy timeseries

Google

Normalized Shannon Entropy:

$$E = -\frac{1}{\ln(N)} \sum_i p_i \ln(p_i)$$

p_i = probability of each bin (count in bin i/N)
N = total count

Single pass version (after binning):

$$\begin{aligned} L &= \sum_i c_i \ln(c_i) \\ S &= \sum_i c_i \\ E &= -\frac{L - S \ln S}{S \ln N} \end{aligned}$$

"logsum" L, "sum" S, "entropy" E
c_i = count in each bin

Case study: Entropy: High-level design

Google

Map

- Only calculate partial sums

Shuffle

- Deliver data for each key to the shard handling that key

Reduce

- Calculate the final sums (L and S)
- Calculate the entropy

Case study: Entropy: Details

Google

Map

- Calculate the key (e.g. [source ASN, time bin])
- For each key, emit e.g. { source IP, packet count } tuples

Shuffle

- Reorganize data by the [source ASN, time bin] key
- A particular shard receives all the tuples for a particular [source ASN, time bin] key

Reduce

- Iterate through the data calculating a map[source IP] of packet counts
- Finally, iterate through the map and perform the one-pass entropy calculation

Case study: Entropy: Optimization

Google

Typically, you would be generating multiple such entropy time series

- source IP, dest IP, source port, dest port

perhaps multiple weightings

- by packet count
- by byte count

Optmize by emitting once for each chunk of input records

- data type = enum { sIP, dIP, sPort, dPort }
- e.g. per [ASN, time bin] key do a single emit for a list of all your { data type, packet count, byte count } tuples
  - Advantage: Fewer RPCs
  - Danger: RPC too large

Scaling MapReduces

Google

Map

- How many unique input sources?
  - Log files processed simultaneously
  - HBase rows
- How is data distributed by sharding key?
  - More grouping is better

Reduce

- How many unique sharding keys?
  - More than that many shards is pointless
- Memory/CPU allocation per shard

"Real time" flow analysis

Google

Frequent, small MapReduces over recently arrived data

Time windowing vs. latency are critical considerations (cursors)

Need good bookmarking of input files

Other thoughts:

Google

SiLK http://tools.netsa.cert.org/silk
Can SiLK-like analyses be done using MapReduce? Sort of...

rwfilter
- Yes! Just matching, boolean forward or not on per-record basis
- Hard: doing ipsets, tuples efficiently per shard

rwsort
- Done automatically by sharding key, subkeys (depending on output method)

rwcount, rwuniq, rwbag
- Yes, but need to optimize for scalability

rwstats
- Yes, rwuniq plus sorting by value

rwset
- Yes, sort of. Not easy, not optimized to IPv4
- rwsettool: not really, not as elegantly

Quick, iterative analysis: Not really, unless... (cf. SQL/MR)

Conclusions

Google

Strengths

- Commodity computing platform
- Strong scalability for many problems of interest to us
- Good for ongoing, repeated analyses of large amounts of data
- "Real time" analyses feasible (not as much of a commodity)

Limitations

- Inherent overhead in shuffling phase
  - Irreducible anyway? Remember: no free lunch
- Not so good for iterative, *ad hoc* analysis (except SQL/MR)

Presenting Mongoose
A New Approach to Traffic Capture

(patent pending)

presented by

Ron McLeod and Ashraf Abu Sharekh

January 2013

Outline

- Genesis - why we built it, where and when did the idea begin
- Issues – requirements
- What we built and how it works (mostly)
- Recent and current challenges
- Our biggest challenges and ongoing Work

Genesis - Why?

- Network administrators require situational awareness to detect:
  - Scanning, Intrusion, Exfiltration, Policy Violations and System Performance Issues
  - Most organizations that we have encountered in our security practice are not monitoring or logging their network activity beyond bandwidth usage.
  - The exceptions in our experience being large government entities and Universities with significant IT staff.
  - So we started asking Why not? …More on this later.

Genesis - When

- It all came together during a “Walk in the Desert” and the statement “We would like to monitor the activity of the network from outside the network. And (maybe) without the users knowing that it is happening.”
- For security reasons I can’t say what desert or who made the statement or who I was walking with.
- There were other pre-existing sparks but this was a watershed moment.

Issues - Privacy

Privacy was of significant importance.

- Network owners did not want external monitors to know or leak information about network structure.
- IP addresses may be interpreted as personal private information in some jurisdictions.
- External monitors must not be able to tie traffic to a machine or a user. Only internal Network admins should be able to do this.
- Users (and administrators) are nervous about payload capture, until there is a problem.
- Communication to and from the network must be secure.

Taken together these issues meant that we would have to be able to modify captured traffic.

Issues - Network Architecture

Independence

- Must work regardless of physical layer components (wireless, wireline)
- Must not require knowledge of sub-netting and NAT'ting within the network.
- Must not rely on the presence of firewalls or services such as active directory.
- Must be able to continue to monitor and control a device that moves throughout the network and beyond.
- Must not be blinded by the use of VPN's

Issues - Visibility

- Must not be noticeable to the end user through:
  - performance (CPU, Memory, bandwidth)
  - or as a running application.

Issues - Speed to Deployment

- **To understand this issue, we must first describe our typical incident response experience.**
  - An organization suspects a data breach or is performing an audit.
  - Q & A with the network administrator:
    - **Can you draw me a diagram of your network structure so I can decide where to put the taps?**
      - **No. I didn't build it.**
    - **Can you tell me which of your routers are capable of producing flow or which switches have port mirroring?**
      - **What's flow?**
    - **When do you need this?**
      - **Today.**
- **Alternatively : in covert deployment speed may be of the essence.**

Issues - Control

- Must be capable of remote interdiction and modification.
  - if a machine is doing a bad thing I need to be able to stop it immediately regardless of my network infrastructure, while maintaining the operating state for forensic analysis.
- Interdiction should not obviously be an interdiction unless I want it to be.
  - i.e. if someone is stealing data from a machine I need to stop the theft but I don't want them to run away before the authorities get there.

Simple – Right?

Two Years later….

Mongoose is a host based traffic collection system that:

- installs in a few minutes as a downloadable kernel patch and service.
- captures inbound and outbound traffic at the host.
- Builds a proprietary representation of each packet and places it in a “dump file”.
- dump files containing (initially) 20,000 packets (1.5 meg) are shipped approximately every 2 minutes to a cloud server farm via a secure SSL connection.

At the server farm:

- Dump files are processed to produce a proprietary flow representation and stored in a client database.
- Alert and classification systems constantly scan the flow data (ongoing development)

Through a web interface:

- Network administrators can log in from anywhere and get a near real time picture of their network activity.

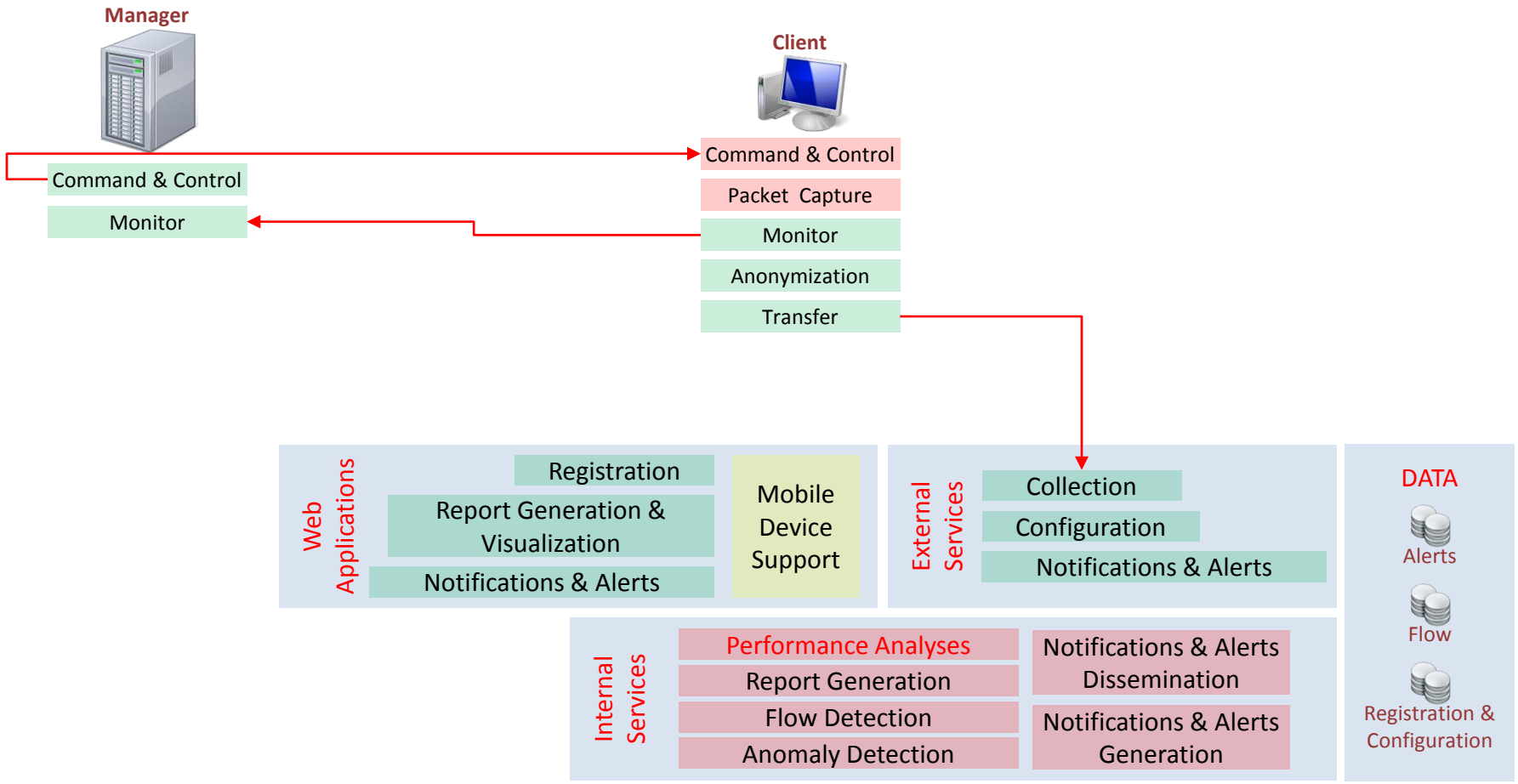
Through a software “Manager” Console:

- Network Administrators can exercise remote control over certain aspects of the client machine.

Manager Functionality
(the parts we can talk about)

- Remotely:
  - retrieve machine information (cpu, memory, processor, operating system etc) for diagnostics or inventory.
  - **start and stop network access**.
  - adjust size of payload capture up and down in real time by any number of bytes to full payload. (**Hybrid Capture**)
  - adjust the number of packets per dump file.
  - start and stop various components in the Mongoose system.
- Other functionality under development – since we are in the kernel you can let your imagination go wild.

Mongoose Architecture



Beta Testing and Experimentation

- Approximately eight months of Beta testing in up to five live production sites operating under confidentiality agreements (geographically distributed).
- 20 - 50 client machines per site reporting to a single collection and processing site.
- Real implementations now have limited shared access to a single collection server and multiple processing nodes, one per customer.

Beta Testing and Experimentation

Excluded traffic

- initially captured everything.
- 90 – 98% of all traffic was local broadcast and link layer traffic for address resolution, name services etc.
- much of this was never meant to exit the local link, but we sent data on all of it out of the edge router….and quickly impacted the bandwidth.
- We currently exclude much of this traffic but may give the network admin the ability to sample it for brief periods.

Beta Testing and Experimentation Environment

- Testing and Development Environment
  - Multiple Servers (VM's) located in both Quebec and Alberta.
  - All beta clients located in Nova Scotia.
- Commercial environment
  - Multiple Servers (VM's) located in Nova Scotia.

Current and Recent Challenges

- Choosing a platform
  - Windows 7 family (Vista, Win 7, Win 8, Server 2008 and so on..)
    - Will not work with XP, server 2003 etc..
  - Android development in the near future

Current and Recent Challenges

- first challenge: capture and modify traffic.
  - We do it in the kernel. We don't use pcap. The rest is secret sauce.
- second challenge: process and ship the packets in a way that does not affect computer performance and is not easily visible to the user.
  - processing dump file on the client causes a cpu spike of < 1 sec/file.
  - Shipping files causes a much smaller cpu spike that does not exceed the normal operating range of other running applications.
    - i.e. if CPU is operating at 30% – 50% then shipping spike is within this range.
- Experiments involved changing the processing algorithm, dump file size and shipping frequency until an acceptable performance level was achieved.
- Dump file size is configurable through the Manager

Current and Recent Challenges

- Secure communication
  - Each Mongoose client contains a unique certificate for use in SSL communication with the collection servers.

Current and Recent Challenges

- Constructing Flows
  - Originally less than 100 lines of C code.
  - 20,000 packet representations are processed in less than 1 sec.
  - Experiments with map/reduce and Hadoop clusters have not yet proved beneficial over our current implementation given the current number of clients (dump files) collected
  - This is due largely to the overhead associated with the Hadoop approach.

Current and Recent Challenges
Some Hadoop Results

- Hadoop with one name node and two data nodes vs existing processing.

Existing: < 1 sec per dump file

Files	Hadoop processing time in secs/file
20	2.45
40	1.28
53	0.96

Using six data nodes we processed 750 dump files at a rate of 0.11 secs/file (best result).

Current and Recent Challenges
Alerts

- Currently four alert categories
  - blacklist of external ips
  - sensitive ports
  - exception reporting on specific machines
  - behavioral classification (neural classifiers)
- Near real time Alert conditions remains our biggest challenge.
  - currently experimenting with algorithmic and system modifications to improve alert performance.

Current and Recent Challenges
Behavioral Alert Classification

Interesting results from neural classifiers for user/machine pairing

- training with 72 hours of real flow data from the population of a beta client
- using flow data statistics similar to that described in my presentation at FloCon 2006.
- multilayer feed forward network with back propagation of error.
- neural network maintains 100% discrimination accuracy for a small sample set of (3) machines for one month without re-training. Not tested beyond this point.
- challenges include the incorporation of the neural classifier into the alert processor and scaling of test population. One is limiting the other. Would like to have the ability to dynamically expand and contract the number of machines we are classifying to test the scalability.

Interesting areas of experimentation and development

- User signatures - isolate an individual based on network traffic. For use in insider masquerade attacks and for covert surveillance.
- Device signatures – isolate a device based on traffic signature. For use in authentication and surveillance.
- Application signatures – classify an application.

Some Unresolved Questions for our Beta Clients

- How long do you want to maintain your flow database? 30 days?
- How long do you need full payload capture to be running? 1 minute per sample?

Our biggest challenges and ongoing work

- performance on the client.
- secure remote communication.
- server infrastructure that is sustainable in the business model.
- provisioning and decommissioning customers and clients.
- Near real time alerts and classification.

Thank You!

Questions?

NetFlow LOGIC

Taming Big Flow Data

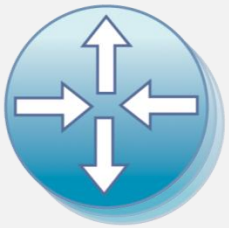
Intelligent Approach to Integrating Flow data with Mainstream Event Management Systems

Igor Balabine, CTO
Sasha Velednitsky, Co-Founder

NetFlow Logic
www.netflowlogic.com

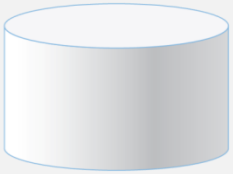
January 2013

Problem



400K recs / sec

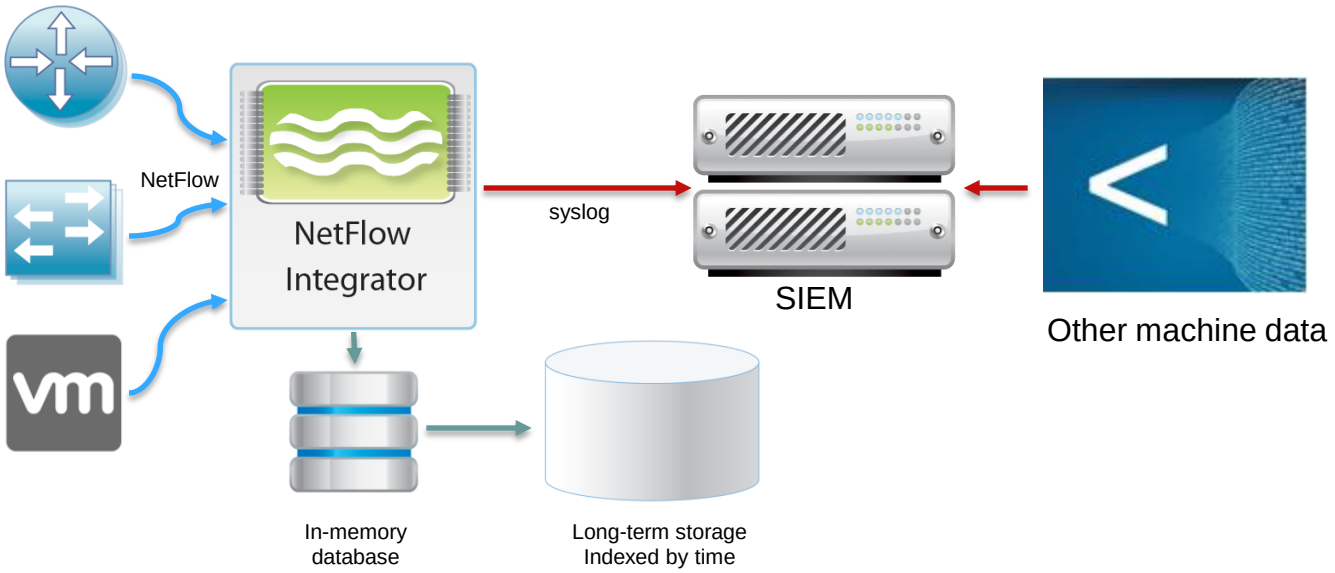
1.6TB per day! =



$$$

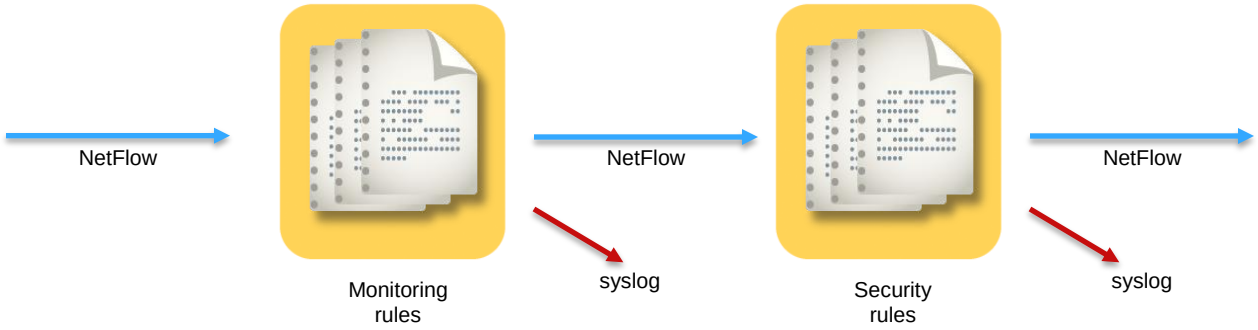
- Modern network devices can create 400K flows / sec. (1.6TB/day of NetFlow data from a single device)
- NetFlow collectors are incapable of processing that much data at reasonable cost
- This problem requires a drastically new approach
- NetFlow collectors / analyzers often are isolated from other log management

Proposed Solution



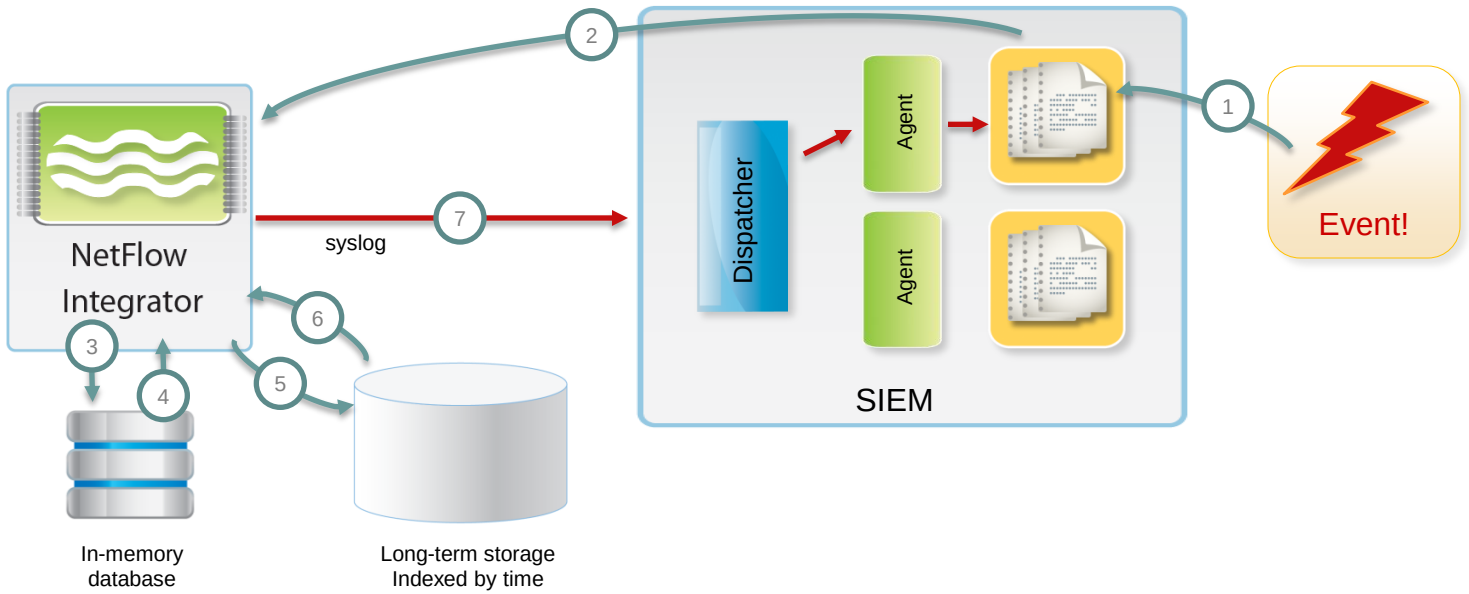
- Consolidated flow information is sent to SIEM in syslog format
- SIEM may request to provide detailed NetFlow data in Δ t around interesting events

Flow Consolidated Information



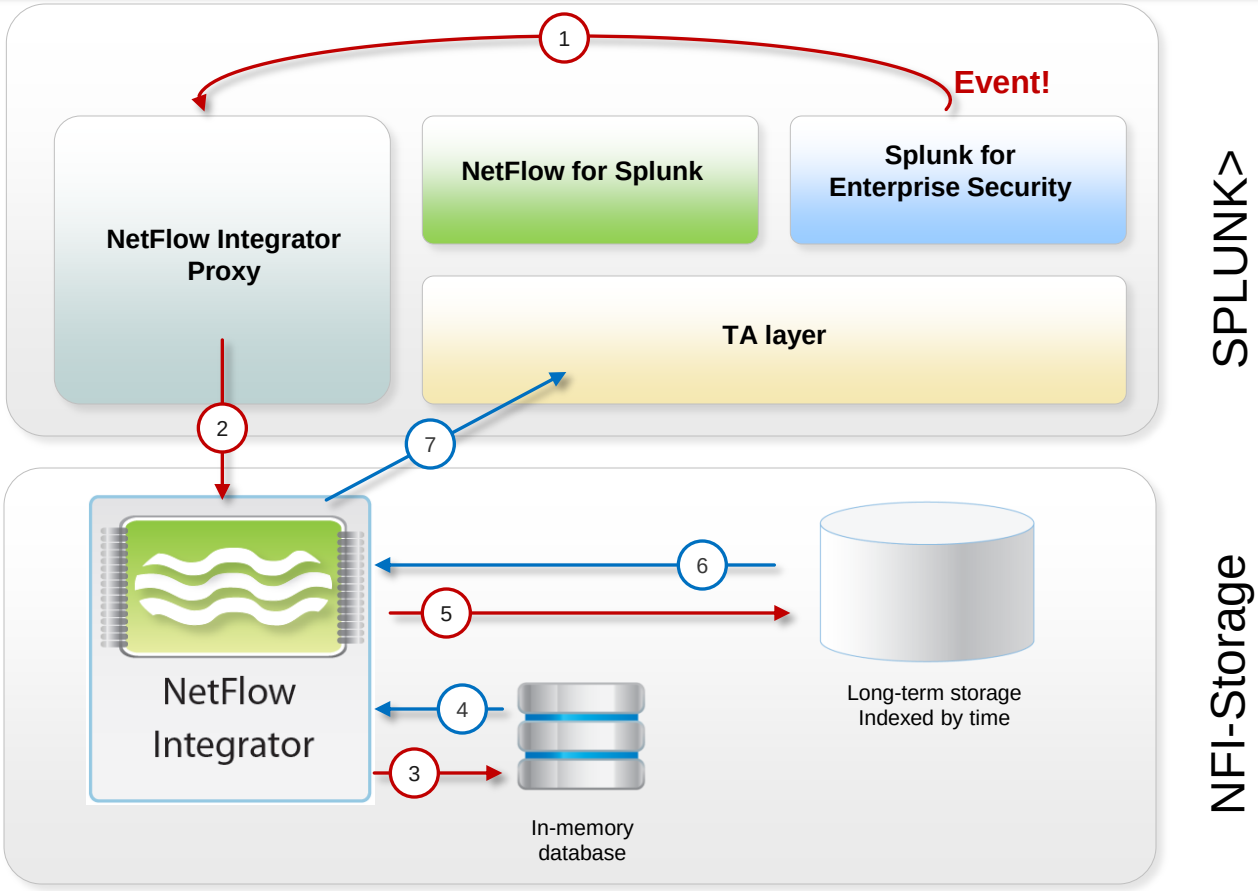
- Traffic Summary
- The number of network policy violations, such as ACL, exceeds a certain threshold
- A host on internal network generates unusual traffic volume
- A host on internal network generates unusual number of connections
- Events based on host reputation
- And so on… just add rules to NetFlow Integrator

Qualifying Events Reported to SIEM



- Event: configuration change
- Was the user who made the change associated with the network flow of the source IP address assigned to that user?
- Request is sent to NetFlow Integrator: provide network traffic detailed for Δ t around the event
- If the user who made the change was not associated with the network traffic – we discovered an imposter!

Example: NFI-Storage + Splunk



- Splunk App for Enterprise Security detects security event and requests the underlying flow information through NetFlow Integrator Proxy
- NetFlow Integrator retrieves flow records for Δ t around the event and sends them in syslog format via Splunk Technology Add-on

Identifying Network Users Using Flow-Based Behavioral Fingerprinting

Barsamian, Berk, Murphy

Presented to FloCon 2013

What Is A User Fingerprint?

- Users settle into unique patterns of behavior according to their tasks and interests

- If a particular behavior seems to be unique to one user…
  … and that behavior is observed…
    … can we assume that the original user was observed?

- Affected by population size, organization mission, and the people themselves

Why Fingerprint?

- Basic Research
- Policy Violations and Advanced Security Warning
- Automated Census and Classification

Why Fingerprint?

- Basic Research
  - Change Detection
  - Population Analysis
- Policy Violations and Advance Warning
  - Preliminary heads-up of botnet activity
  - Identify misuse of credentials
- Automated Census and Classification
  - Passive network inventory
  - User count estimation (despite multiple devices)
  - Determination of roles

Background

- Passive and active static fingerprints
  - Operating system identification
    - p0f/NetworkMiner, Nmap
  - Signature-based detection of worms and intrusions
- Dynamic fingerprints
  - Hardware identification
  - Unauthorized device detection[1]
  - Browser fingerprinting[2]
- Increasingly important part of security systems[3]
  - Reinforcing authentication
  - Identifying policy violations

[1] Bratus, et al "Active Behavioral Fingerprinting of Wireless Devices", 2008
[2] http://panopticlick.eff.org
[3] François, et al "Enforcing Security with Behavioral Fingerprinting", 2011

But…

- Difficult to implement, requiring significant expertise not available to many IT departments
- Require unusual or unavailable data
  - Data collection incurs overhead; easier to justify if data is useful for multiple purposes
    - No unitaskers in my shop!
  - Protocol analysis needed
    - Computationally expensive
    - Impinges user privacy
    - Increasingly defeated by encrypted channels and tunnels

Challenge

Make active, adaptive fingerprinting available to the widest possible set of network administrators

- Data requirements
  - Common data source, common data fields
- Processing requirements
  - Can't require major computing resources to create and handle
- Ease of implementation
  - Not just technology, but policy
  - Could search emails and web forms for personally-identifying statistically improbable phrases, but would never fly at most institutions

Why NetFlow Fingerprints?

- NetFlow has very attractive properties to an analyst…
  - Privacy
    - Unintrusive to end users
    - Not affected by encrypted channels
  - Speed
    - Easily-parsed datagrams with fixed fields
    - Bulk of processing taken care of by specialty equipment
  - Scalability
    - Less affected by volume than protocol analyzers
- … but is it up to the task?
  - (Spoiler alert: yes)

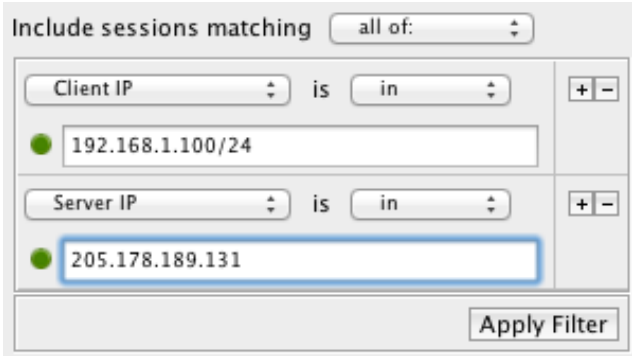
Methodology

After multiple revisions, arrived at the following:

1. Define your parameters
2. Get a list of all the outgoing sessions from that subnet
    1. List of sessions for which client IP is in CIDR block of interest
    2. From that list, extract the destination addresses
3. For each of those destination addresses, do a 'ip-pair' query: (CLNIP==classC && SRVIP=dest).
    1. Count the unique local addresses for each destination
4. Eliminate all of the external addresses that get contacted by more than 1 local address
5. Result is a set of external addresses that are only contacted by ONE client

(CLNIP==*classC*)

(CLNIP==*classC* && SRVIP=*dest*)



FLOWTRAQ

Example Fingerprints

- Individual fingerprints for a user (when that user has one) contain a list of IP addresses that user (and only that user) contacted within the time period
  - One-time connections not included here
- Using the Class C block for the server would compress fingerprints like User B's
  - In this case, would still be unique

User A	8475 total sessions
aaa.93.185.143	38
bbb.175.78.11	44
ccc.22.176.46	42
ddd.28.187.143	37

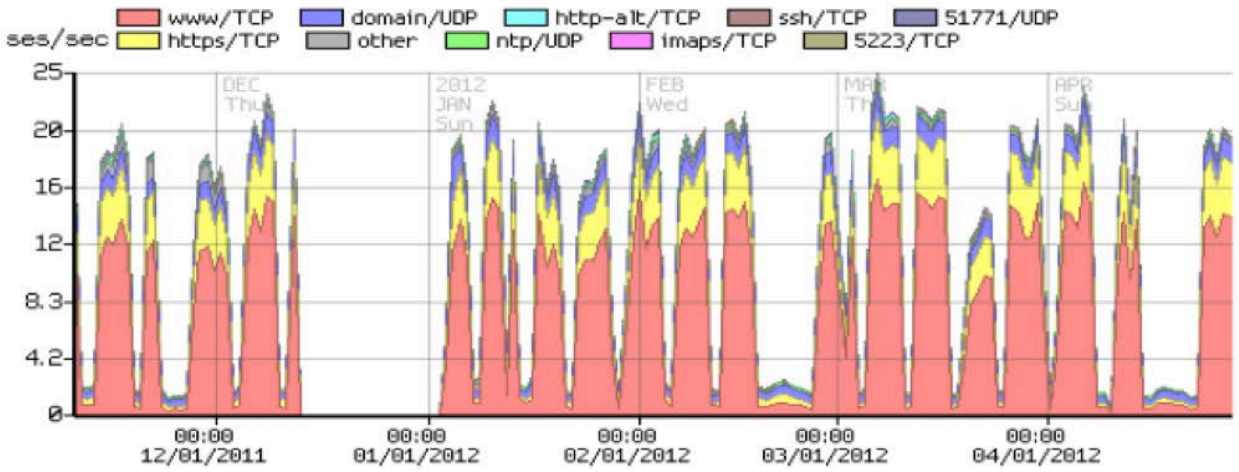
User B	661 total sessions
eee.87.169.51	93
eee.87.160.30	34
eee.87.169.50	37

Parameters

- Definition of local network
  - Select the smallest network of interest
  - May be worth fingerprinting wired and wireless networks separately, to account for users with both desktops and wireless devices
- Time frame
  - Shorter-term profiles faster to create
  - Longer-term profiles less transitory
- Destination subnet
  - When filtering on each destination, using a slightly wider subnet can reduce the computing impact of content distribution networks
- Top N vs. All
  - Cutting off the list of servers with very few sessions improves scalability
  - Potential reduced fingerprint list

Data Source Characterization

- Knowing your source helps determine optimal parameters
- Educational environment with a mix of wireless and wired infrastructure
- Inherent “life spans” to fingerprints
  - Large turnover each year
  - “Mission” changes every term
  - Gaps in data (scheduled breaks) confound ability to detect gradual change

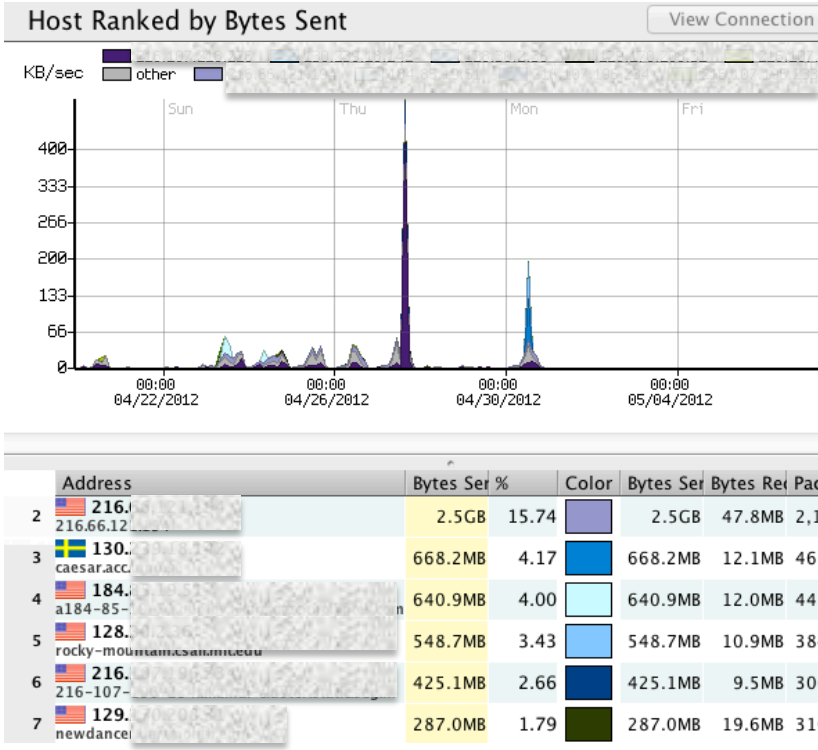


FLOWTRAQ

Select Outbound Requests

- Get a list of top servers by destination

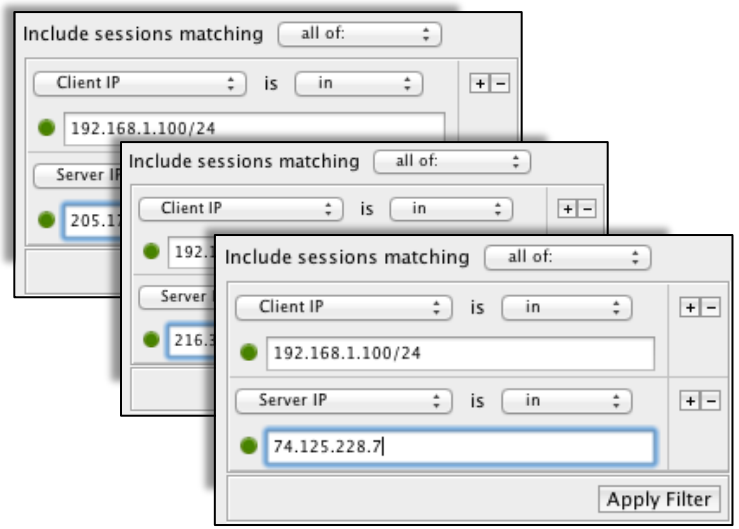
- How do you define “outbound” and why?
  - Anything outside examined subnet? Outside organization?
  - Presumption that use of internal resources not identifying?
    - Mostly true, but what about private servers?



FLOWTRAQ

Select Pairs

- For each server in Top N list, get the list of clients that contacted it
- Filter to reduce computation?
  - Select only ports of interest (HTTP)
    - Avoiding BitTorrent makes for stronger profiles
  - Filter out known-common networks (Akamai, Google)
  - Include only servers with more than some minimum number of sessions



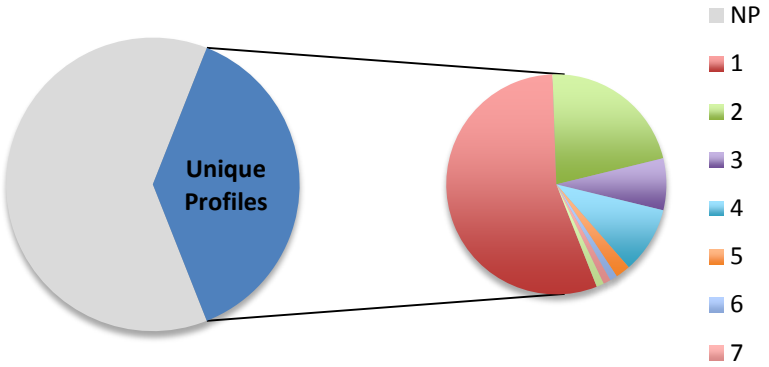
FlowTraq

Compile Fingerprints

- At this stage we have a list of those servers that have only been contacted by one client
  - Potentially pre-filtered for significance (e.g. minimum number of sessions, removed trivial connects such as BitTorrent, etc)
- Create for each client a list of servers
  - Optionally: ranked by percent of client's total traffic (requires second query for each client, increasing total fingerprint time, but providing context and significance measure)
- Each list is a basic but functional fingerprint of that client
  - Sessions to one of those servers in future traffic indicates likely link to that fingerprinted user
    - Primary: that user generated that traffic (on the original device or not)
    - Secondary: that user is connected directly to the user who generated that traffic

Initial Results

- Of ~250 users, profiles could be created representing
  - 38% of users
  - 53% of total traffic
- Breakdown by profile length (# servers in profile):
  1. 51 users (55.4% of profiles)
  2. 20 users (21.7%)
  3. 7 users (7.6%)
  4. 9 users (9.8%)
  5. 2 users (2.2%)
  6. 1 users (1.1%)
  7. 1 users (1.1%)
  8. 1 users (1.1%)

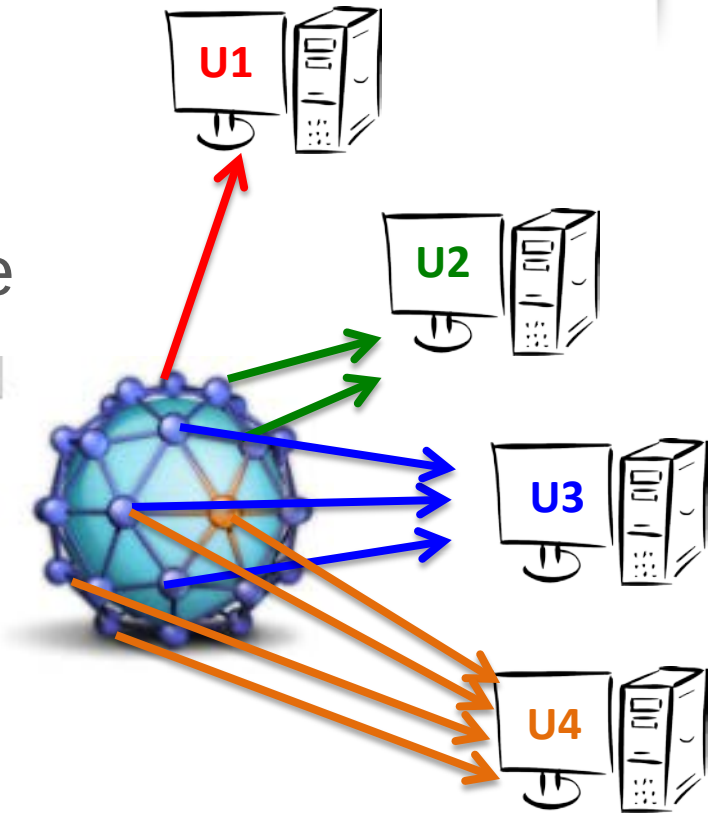


(i.e. 51 users each contacted 1 host unique to them, and one user contacted 8 hosts that nobody else did)

FLOWTRAQ

Uniqueness Levels

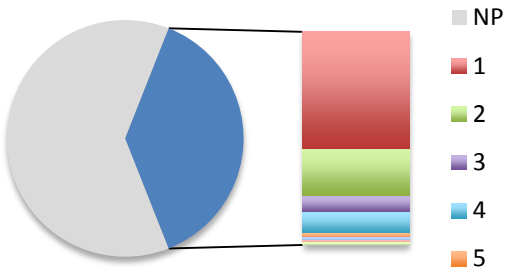
- By relaxing uniqueness requirement, more users can be fingerprinted
  - Tradeoff: Certainty vs. breadth
- Nomenclature
  - The more clients that share a host, the higher the U number



- What is lost in ability to pinpoint users, is gained in insight into shared task/interest
- Some profiles non-unique
  - Same user at different IP addresses?

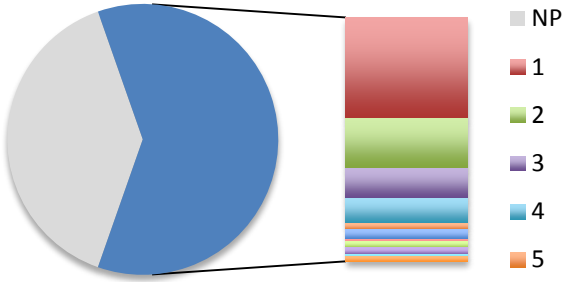
U1-U4 Profile Lists

U1 Profiles



38% of users, 53% of traffic

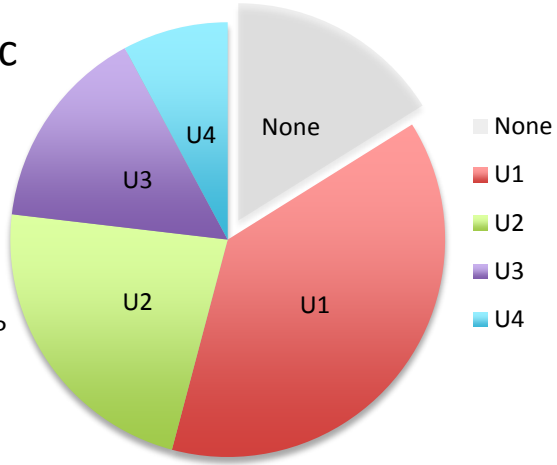
U2 Profiles



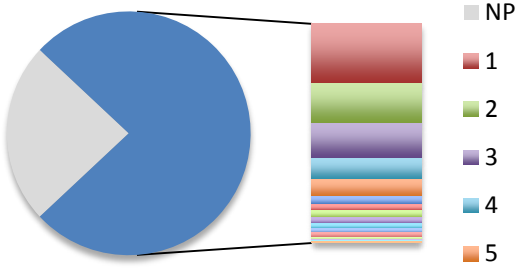
60% of users, 78% of traffic

12 non-unique users

Membership



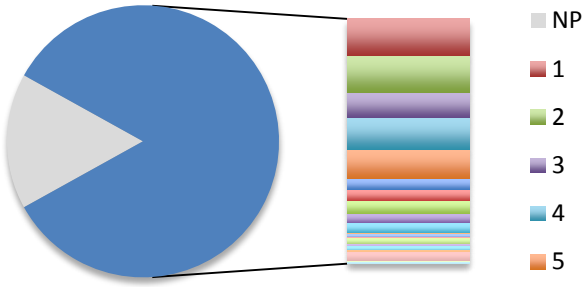
U3 Profiles



75% of users, 89% of traffic

10 non-unique users

U4 Profiles



83% of users, 93% of traffic

10 non-unique users

Variance Over Time

- Variability from month to month is observed
- Month 1

Uniqueness	% of users	% of traffic
U1	38%	53%
U2	60%	78%
U3	75%	89%
U4	83%	93%

- Month 2

Uniqueness	% of users	% of traffic
U1	46%	80%
U2	60%	92%
U3	69%	96%
U4	75%	98%

Results and Lessons Learned

- This represents a first step toward making simple flexible fingerprinting widely available
  - NetFlow is an ideal data source
- Able to fingerprint users comprising majority of network traffic in relatively unrestricted environment
- Uniqueness Levels
  - U1 profiles are more significant
  - U4 profiles cover far more of the population
  - Keeping track of them in parallel allows us the best of both worlds

Take-Home

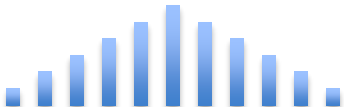
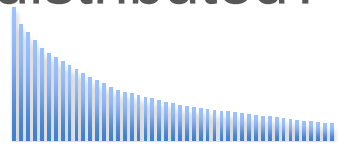
- NetFlow, with its benefits to privacy, ease, and scalability, can be used to produce simple user fingerprints
  - Several types are possible; we went with the simplest plausible type
- Unique site accesses represent one such fingerprint type
  - Intuitive and easy to grasp
  - Adjustable to the level of desired uniqueness
- More sophisticated fingerprints are expected to be more useful still

Next Steps, Short-Term

- Room to grow within NetFlow collection regime:
  - Refine by port/protocol
  - Aggregate content distribution networks
- Make better use of ground truth
  - Newer version of software allows searching on MAC address, to quickly check when fingerprint appears to change or duplicate
  - Determine whether there are substantive differences between wireless and wired networks
    - Number of individuals with identifiable fingerprints
    - Fingerprint stability

Next Steps, Long-Term

- Learning Period Estimation
  - What constitutes a baseline?
- Long-Term Stability
  - How much do these fingerprints change over time?
  - What can be learned from those changes?
  - How are fingerprint lives distributed?


vs


- Autonomous Operation
  - Can fingerprint creation and tuning be automated?

    … to the point of using them for auto-remediation?

For Additional Information…

- For a copy of these slides and the whitepaper, or to evaluate the fingerprinting tool, visit us at:
  - http://www.flowtraq.com/research/FloCon2012.html
- We would be happy to address any questions or comments
  - abarsam@flowtraq.com
  - vberk@flowtraq.com
  - jmurphy@flowtraq.com

MS-ISAC CERT

Capabilities

- Incident Response
- Malware Analysis
- Computer Forensics
- Network Forensics
- Log Analysis
- Statistical Data Analysis
- Netflow Monitoring / The Albert
- Rapid Deployment

Malware Analysis

- Static and Dynamic Analysis
- Reverse Engineering
- Can analyze around 1000 malware samples daily
  - Albert integration is underway

Computer and Network Forensics

- Certified and experience staff
- Performed as part of incident response or as a separate case
- Chain of custody is always maintains
- Also assisting FBI, USSS and HIS on their forensic cases

The Albert

- Currently monitoring 16 states and 1 territory
  - 5 additional states are in the process of being added to the service
- Near real-time alerts are verified and sent to states
- Anomaly detection capabilities are implemented
- DNS mining results in identifying new malicious domains

Teaching Hospital

- Cyber residency program for students
- Malware analysis
- Forensics
- Vulnerability Assessment
- Incident Response

Near Real-Time Multi-Source Flow Data Correlation

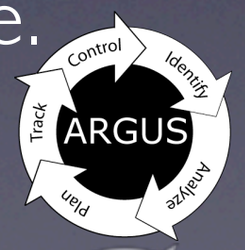
Carter Bullard
QoSient, LLC

carter@qosient.com

FloCon 2013
Albuquerque, New Mexico
Jan 7-10, 2013

Problem Statement

- Cyber incident attribution and forensics, is a complex process.
- To assist in security incident response, recognizable hostile activity needs to be associated with other information system behavior in order to understand the complete cyber security incident life cycle
    - Within a complex internal spoofed stepping stone attack, using a Wiki vulnerability, a machine with an Antartican source address sends a message, that runs a rogue program that sends a command control message to a botnet style agent on an other machine that exfiltrates data back to Antartica.
- For most existing protection strategies, this isn't detectable.

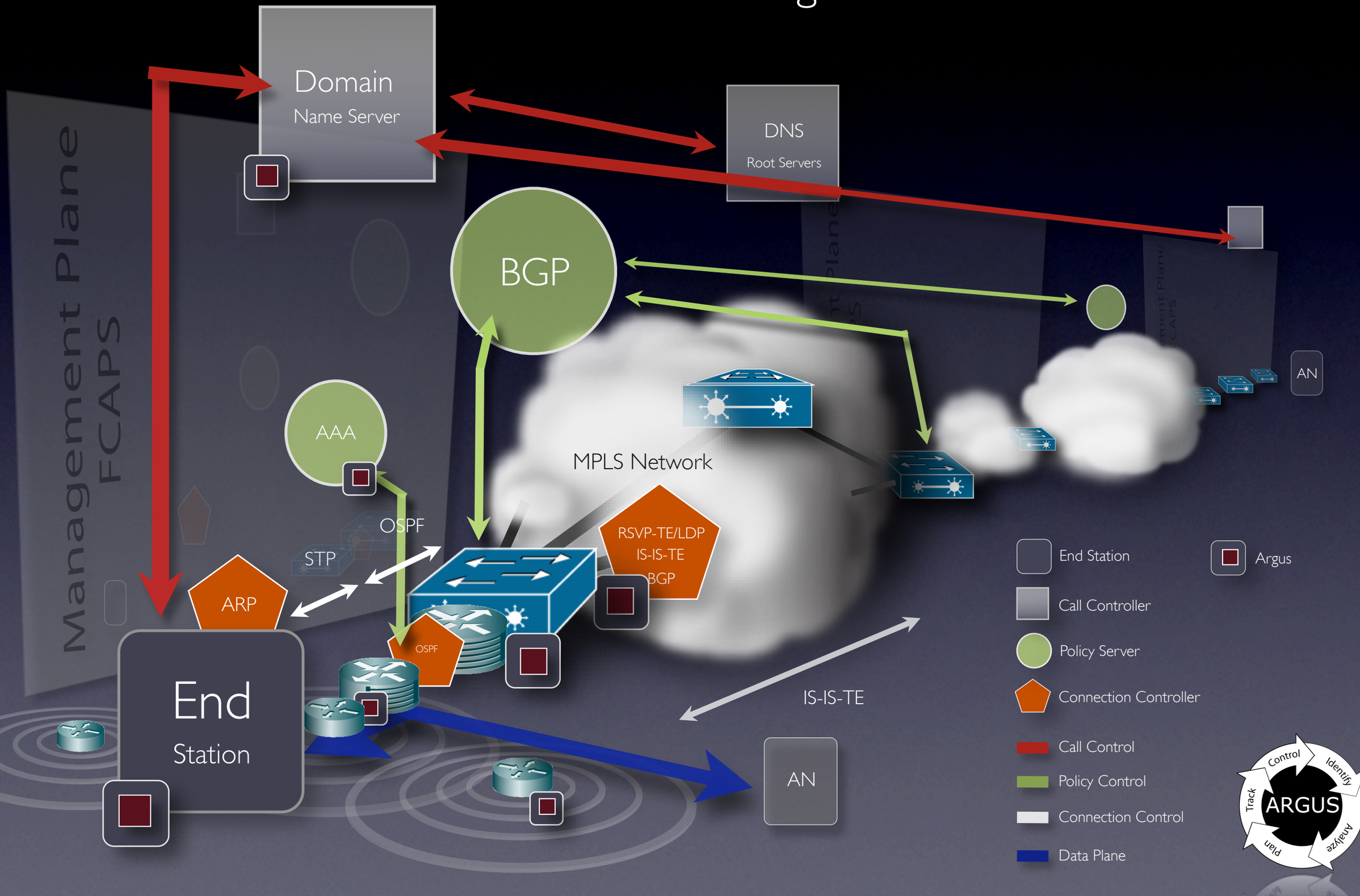


Flow data is an important component

- Exfiltration should be detectable from sensors on the external border, or from a sensor in Antartica - But in this case, nada
- The machine that sent the data should be able to report to something that it sent data to Antarctica - But there aren't any logs that contain that transaction
    - Having some form of audit for the network activity of key hosts, is important.
    - Having a means to associate that transfer with the program that actually sent the data is critical, here.
        - Realizing that that program was run by a program, not by the current user of the system, is important.
- The machine that was accessed by the Antarctican machine, like most internal machines, provide inadequate access control, protection or auditing to track.
    - Associating that program with the stimulating / initiating message from Antarctica is critical
- Realizing that the machine isn't really in Antartica, but its down the hall, is going to be a challenging problem.

Comprehensive Enterprise Awareness

Dealing with the Insider Threat



How to approach this

- Establishing a strategy that can help attribution and forensics analysis for the internal attack
  - Establish formal attribution / non-repudiation systems
- Improve audit so that the basic information is available, reliable, and relevant
  - At least each host should maintain a network activity log
- Improve methods and techniques so that correlation can be used to make the end-to-end attribution possible.
- Currently, for many sites, its really luck, rather than engineering, that makes this stuff work

How to deal with host issues?

- We need to modify system audit strategies to approach this really important problem
- In the absence of direct support, what to do.
  - We can install flow monitors on hosts
    - That will provide the network audit
    - argus is a good candidate
  - We need user and program bindings to flow data to make the back chaining possible to deal with our scenario.
    - Socket audits are possible in some systems
    - Demonstrate using lsof() to provide that info.

Argus Strategy

- In argus we have integrated into the basic argus data generation, collection, processing, storage and analytics, the ability to correlate flow and non-flow data.
- Argus has a facility, Argus Events, that can be used to generate, structure and transport metadata.
- Argus-3.0.6+ supports the collection of many non-flow data sources, including /etc/proc, vm_stat, SNMP data, and lsof() output.
- We've implemented the ability to correlate lsof() data with cached flow data, as a simple example, in all ra* programs

End-to-End Situational Awareness

Network Optimization - Black Core Mesh

SNMP RMON Element Statistics/Traps
Interface Status / Transitional Events
Bulk Link Statistics

System Layer 2-7 Flow Data
System Communication Efficiency
Connectivity / Availability
Offered Load / Loss / Jitter
One-Way Delay (GPS synchronization)
Round Trip Delay

Comprehensive Layer 2-7 Flow Data
Site Communication Efficiency
Enterprise Communication Efficiency
Site Offered Load / Loss / Jitter
Network Transit Times

Comprehensive Layer 2-7 Flow Data
End-to-End Communication Efficiency
Reachability / Connectivity
Received Load / Loss / Jitter
Network Transit Times

Comprehensive Layer 2-4 Flow Data
ISP Communication Efficiency
Ingress Available Capacity / Loss / Jitter
One-Way Delay (GPS Synchronization)
Network Path Status

Comprehensive Layer 2-4 Flow Data
Network Path Assurance / Status
Reachability / Availability Assessment
One-Way Delay (GPS Synchronization)

Core Service Provider Management Domain

Enterprise Management Domain

Comprehensive Flow Monitor

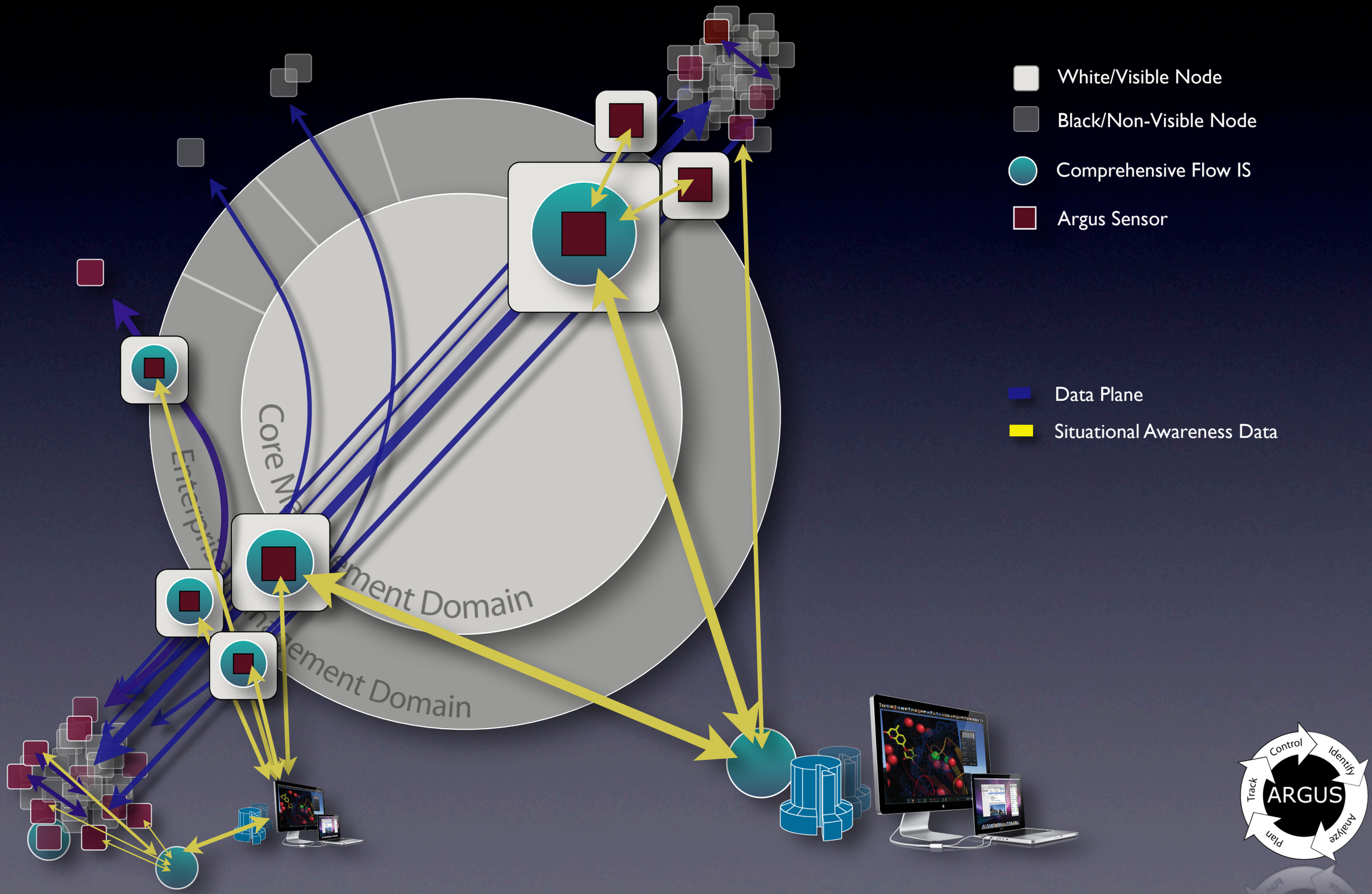
SNMP RMOM Style Monitor

Information System Repository

ARGUS
Control
Identify
Analyze
Plan
Track

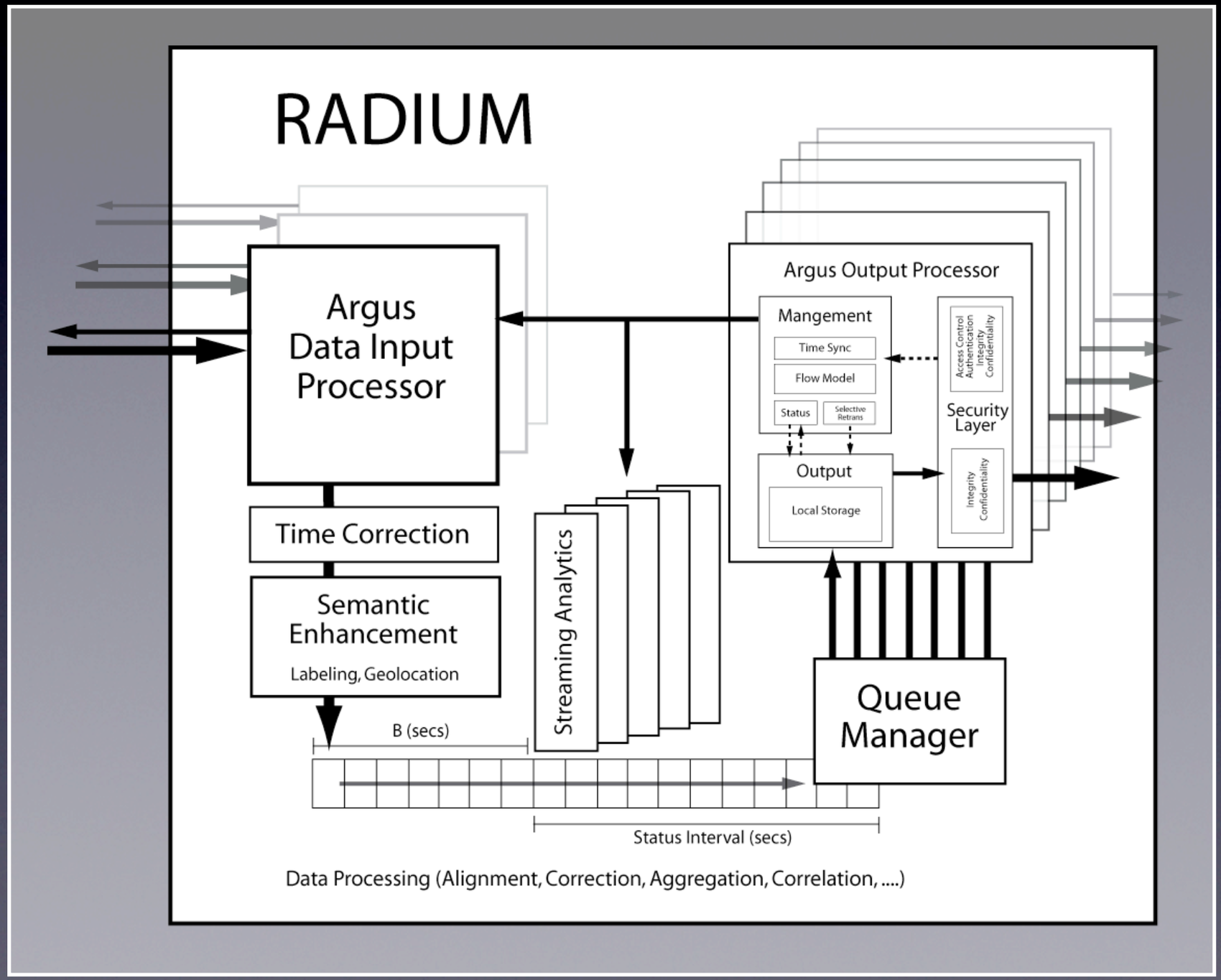
Complex Comprehensive Awareness

Local and Remote Strategies



Radium

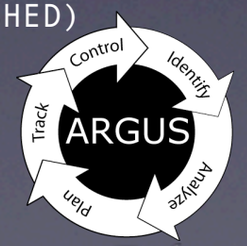
Data Flow Design



Argus Events

- Argus event type specific format for a particular collection, using a generic XML free form strategy.

```
event[49241]=  2013/01/04.12:47:16.733468:srcid=192.168.0.68:prog:/usr/local/bin/argus-lsof
<ArgusEvent>
  <ArgusEventData Type = "Program: /usr/sbin/lsof -i -n -P">
    COMMAND     PID           USER   FD   TYPE     DEVICE SIZE/OFF   NODE NAME
    mDNSRespo    53 _mdnsresponder   56u  IPv4 0xbb72da10      0t0    UDP *:50451
    awacsd       69             root  241u  IPv4 0xbb72da10      0t0    TCP 192.168.0.68:57367->17.172.208.94:443 (CLOSED)
    apsd         71             root   10u  IPv4 0xbb72da10      0t0    TCP 192.168.0.68:53556->17.149.32.65:443 (ESTABLISHED)
    blued        72             root    4u  IPv4 0xbb72da10      0t0    UDP *:*
    ntpd         75             root   20u  IPv4 0xbb72da10      0t0    UDP *:123
    radium      110             root   10u  IPv4 0xbb72da10      0t0    TCP 192.168.0.68:49166->192.168.0.68:561 (ESTABLISHED)
    radium      110             root   11u  IPv6 0xbb72da10      0t0    TCP [::1]:562->[::1]:49171 (ESTABLISHED)
    [snip]
    Keynote   68546           carter    8u  IPv4 0xbb72da10      0t0    TCP *:49901 (LISTEN)
    raevent   69821           carter    5u  IPv6 0xbb72da10      0t0    TCP [::1]:51255->[::1]:562 (ESTABLISHED)
    perl5.12  69824             root    4u  IPv6 0xbb72da10      0t0    TCP *:561 (LISTEN)
    perl5.12  69824             root    6u  IPv4 0xbb72da10      0t0    UDP *:*
    perl5.12  69824             root    8u  IPv6 0xbb72da10      0t0    TCP 192.168.0.68:561->192.168.0.68:49166 (ESTABLISHED)
    perl5.12  69824             root    9u  IPv6 0xbb72da10      0t0    TCP [::1]:561->[::1]:58040 (ESTABLISHED)
  </ArgusEventData>
</ArgusEvent>
```

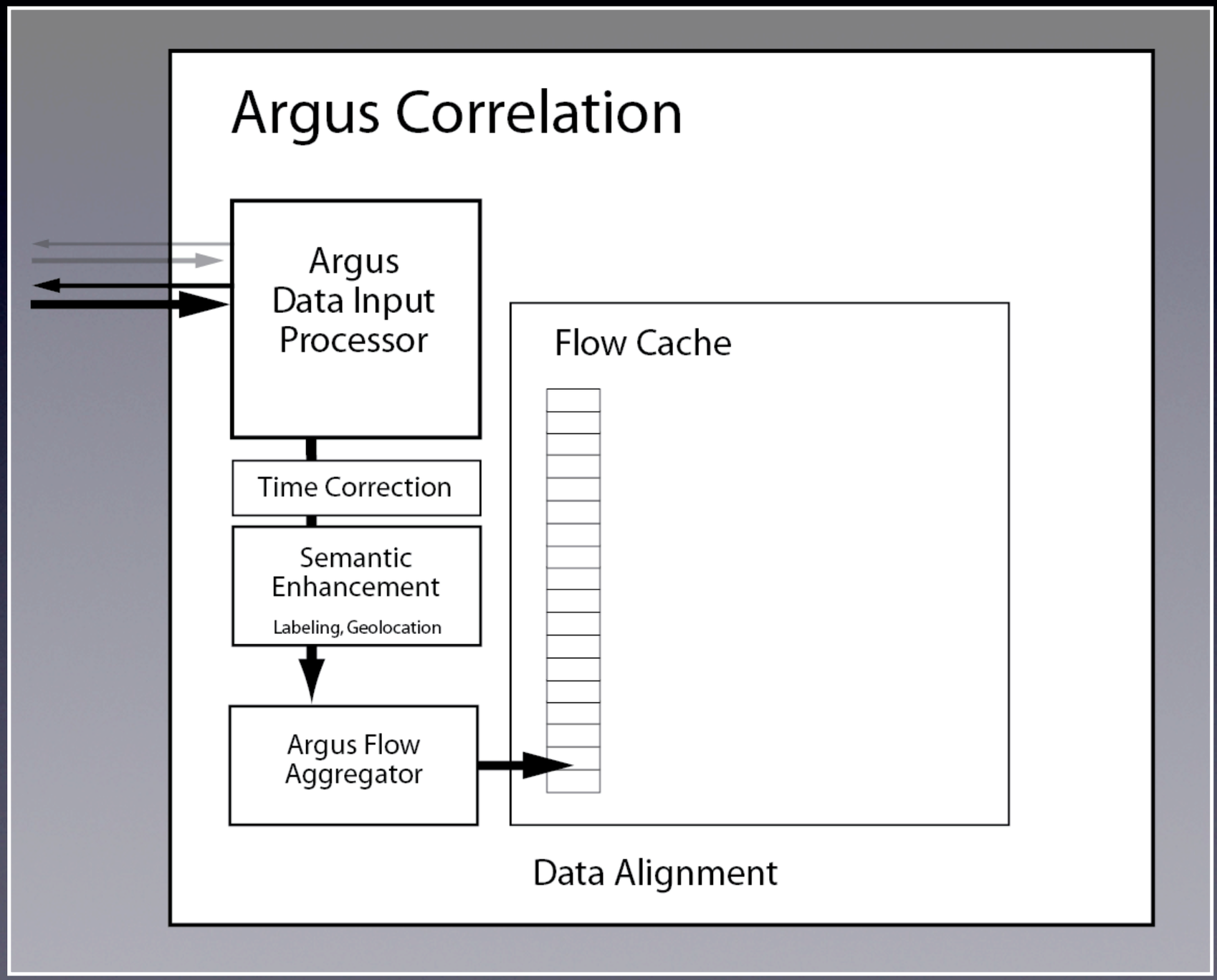


Argus Events Configuration

```
# Argus.conf Argus Event management configuration syntax is:
#       Syntax is: "method:path|prog:interval[:postproc]"
#           Where:  method = [ "file" | "prog" ]
#                   pathname | program = "%s"
#                   interval = %d[smhd] [ zero means run once ]
#                   postproc = [ "compress" | "compress2" ]
#
#ARGUS_EVENT_DATA="prog:/usr/local/bin/ravms:20s:compress"
#ARGUS_EVENT_DATA="prog:/usr/local/bin/rasnmp:1m:compress"
#ARGUS_EVENT_DATA="file:/proc/vmstat:30s:compress"
ARGUS_EVENT_DATA="prog:/usr/local/bin/argus-lsof:30s:compress"
```

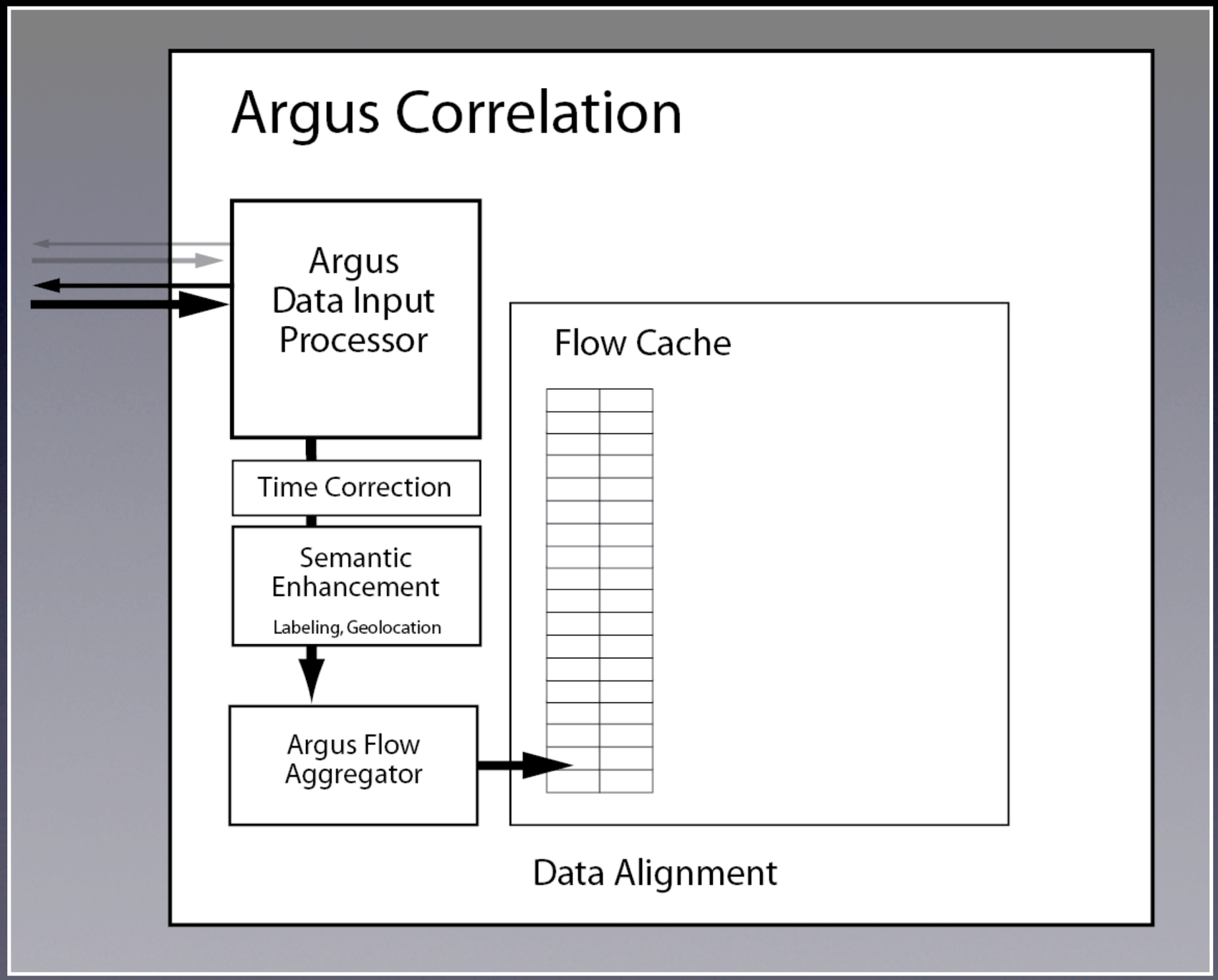
Argus Correlation Design

Radium Process



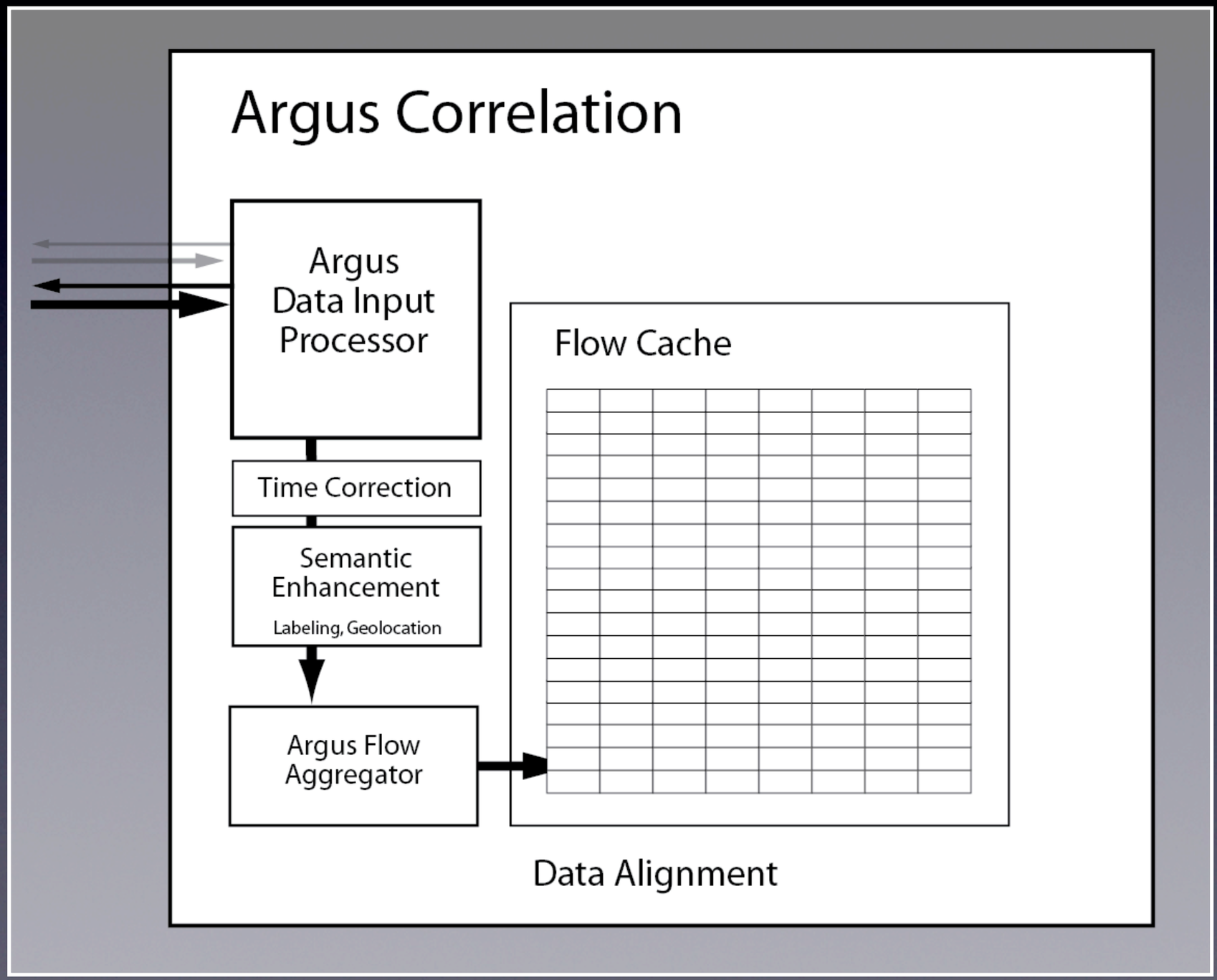
Argus Correlation Design

Radium Process



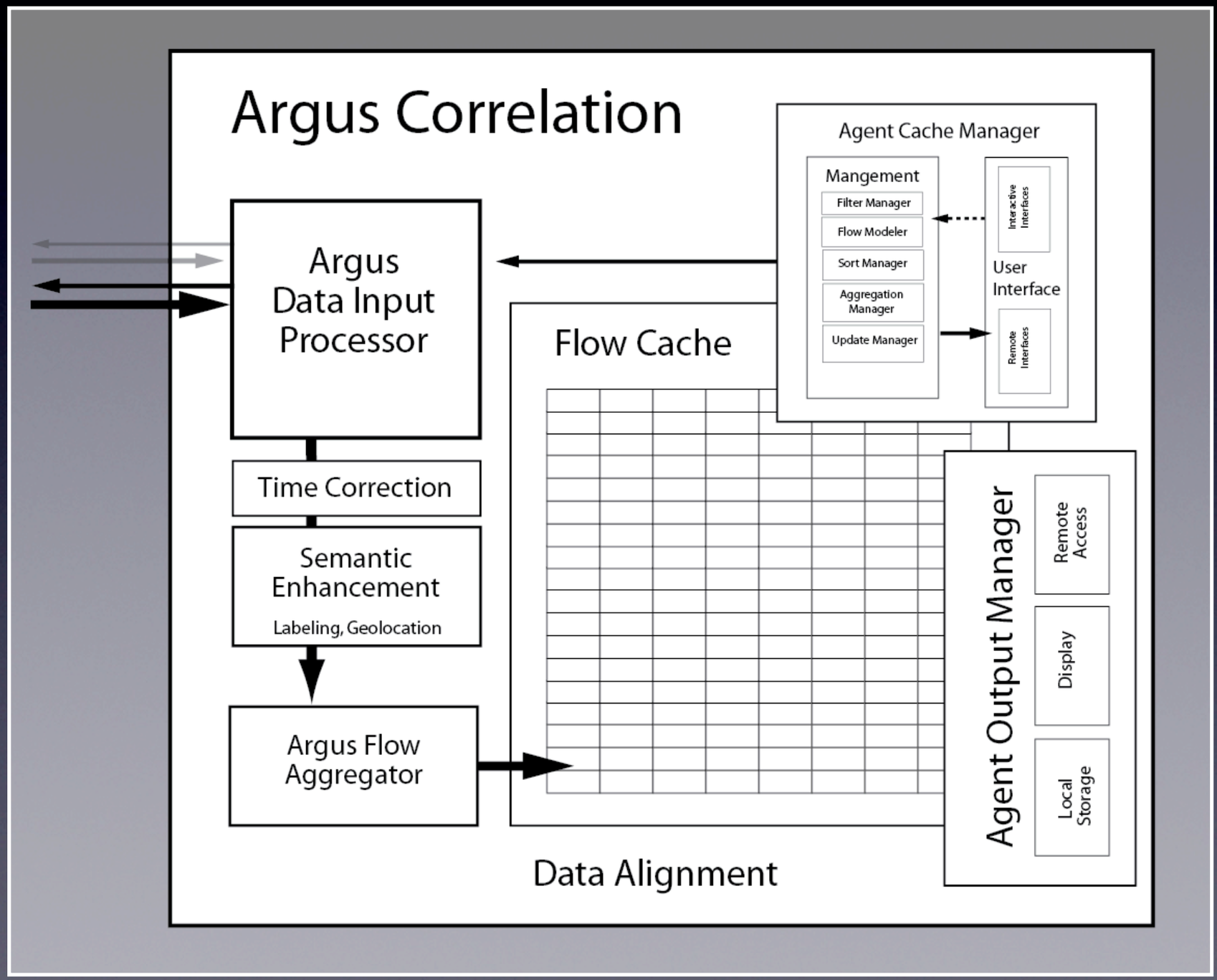
Argus Correlation Design

Radium Process



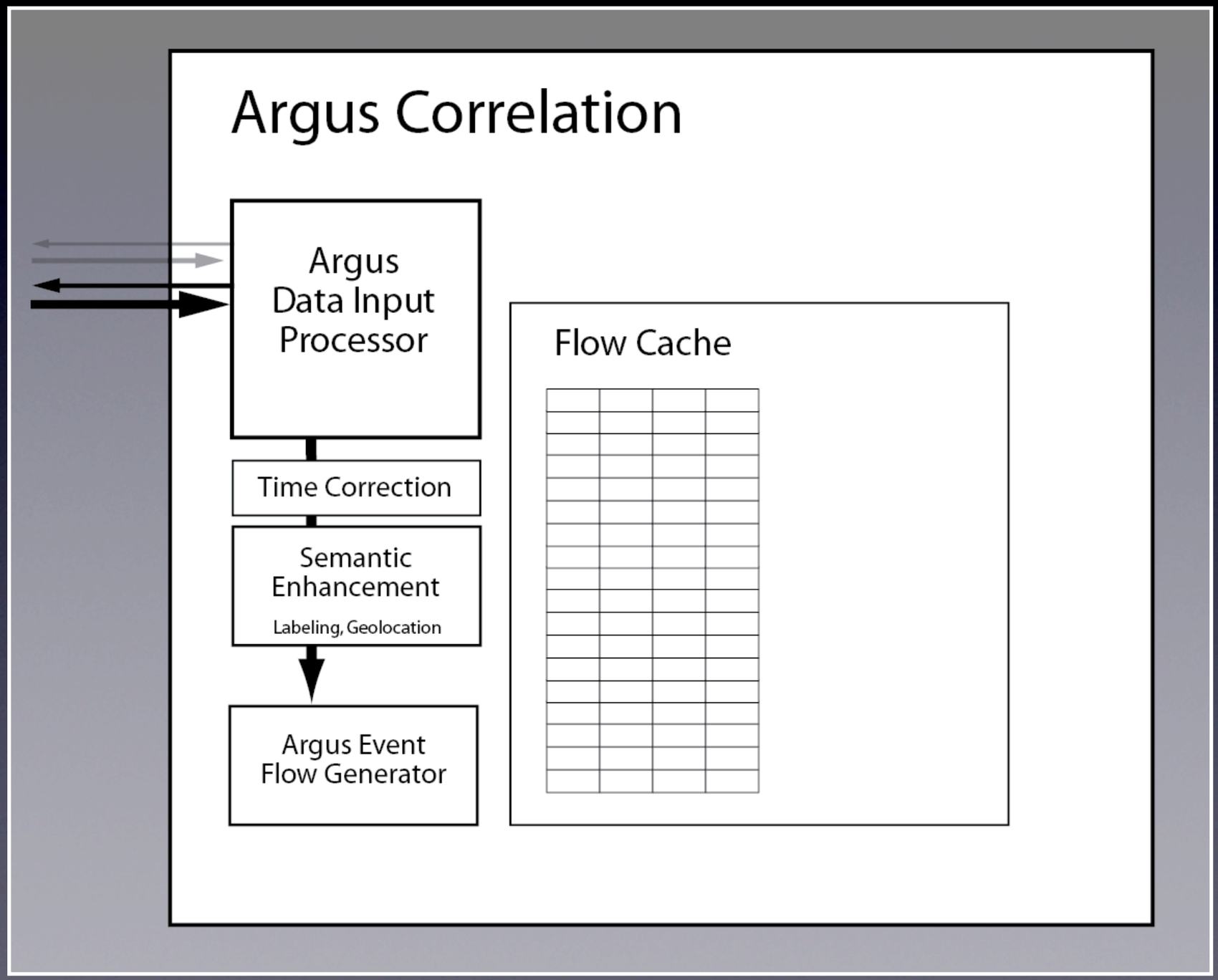
Argus Correlation Design

Radium Process



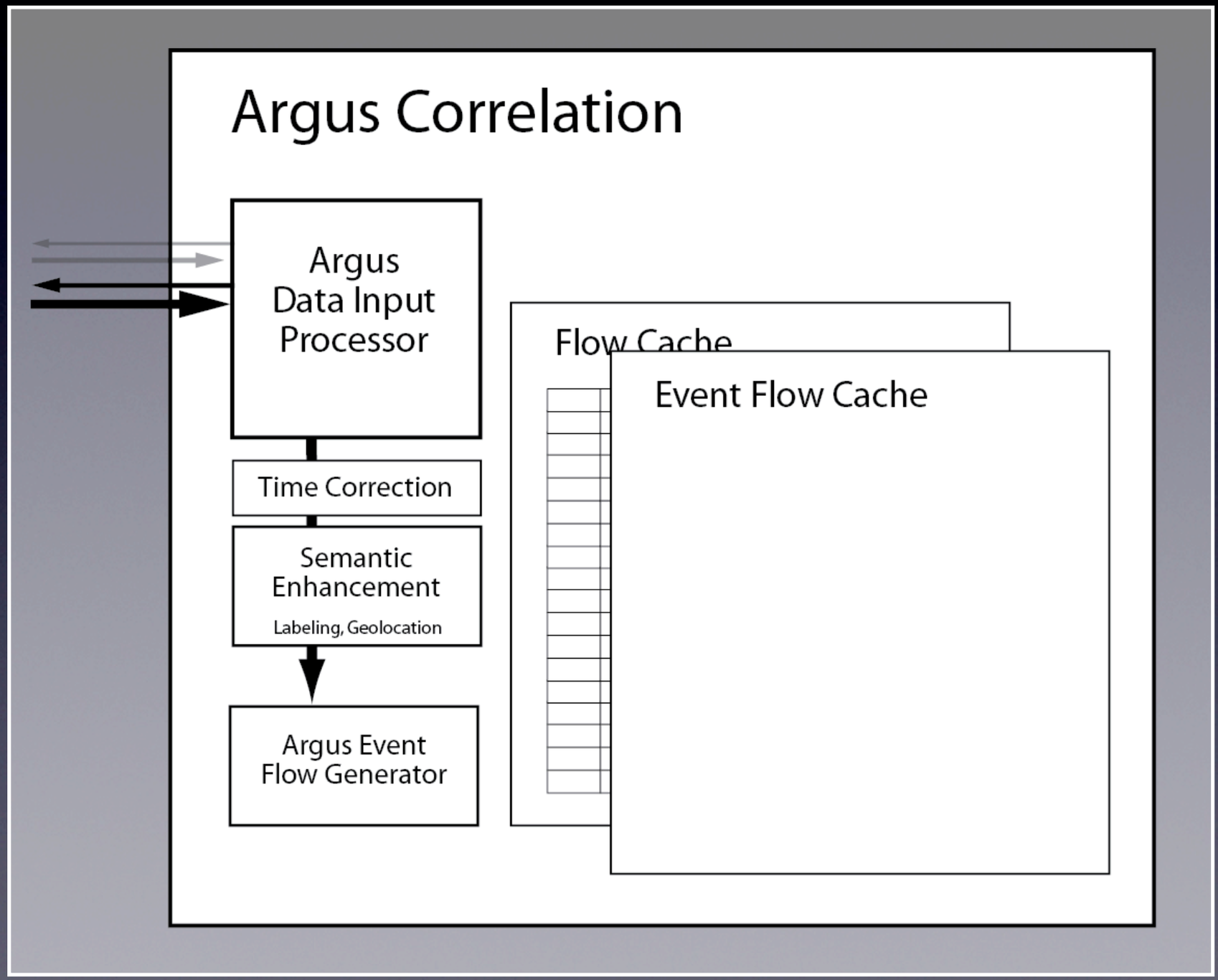
Argus Correlation Design

Radium Process



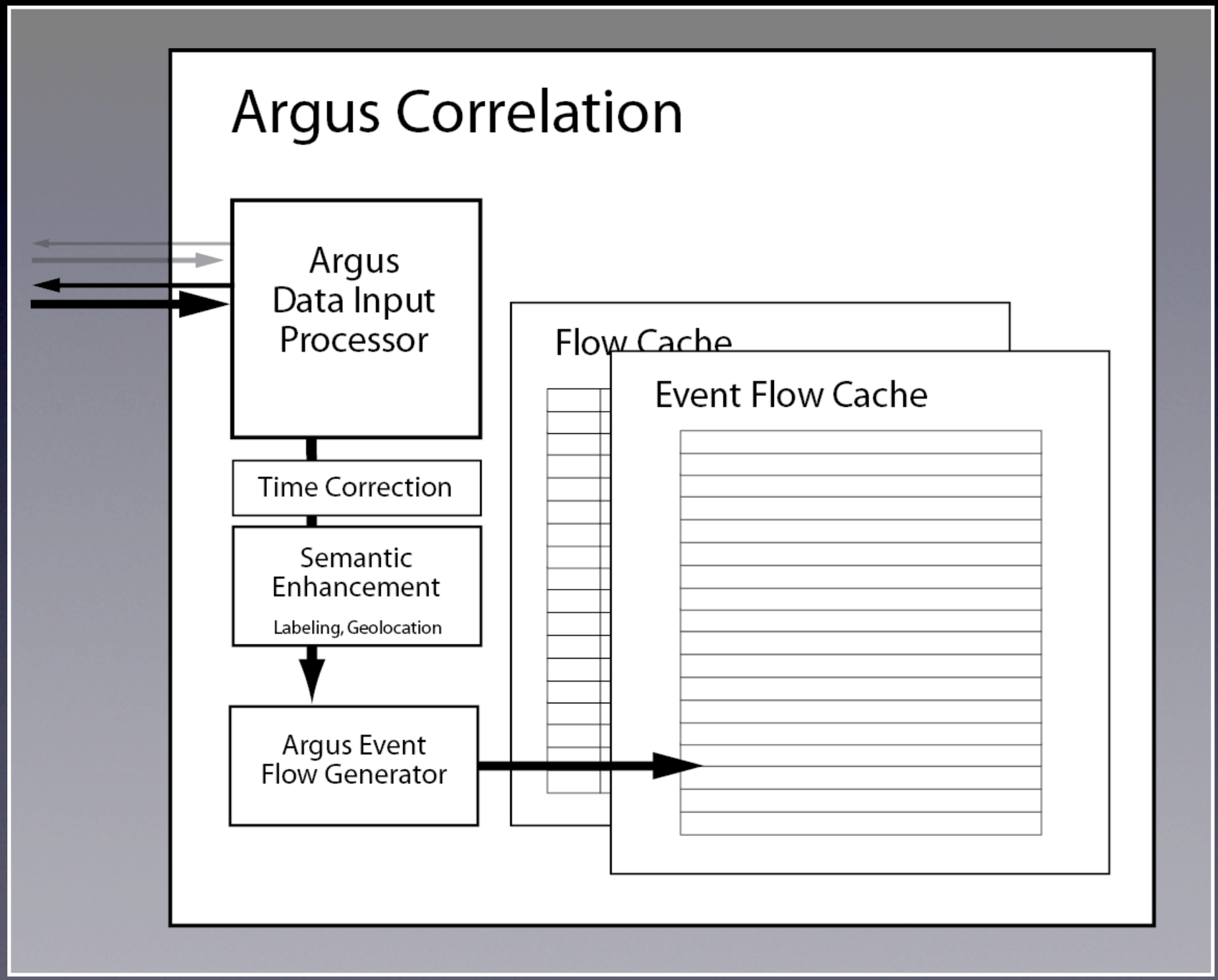
Argus Correlation Design

Radium Process



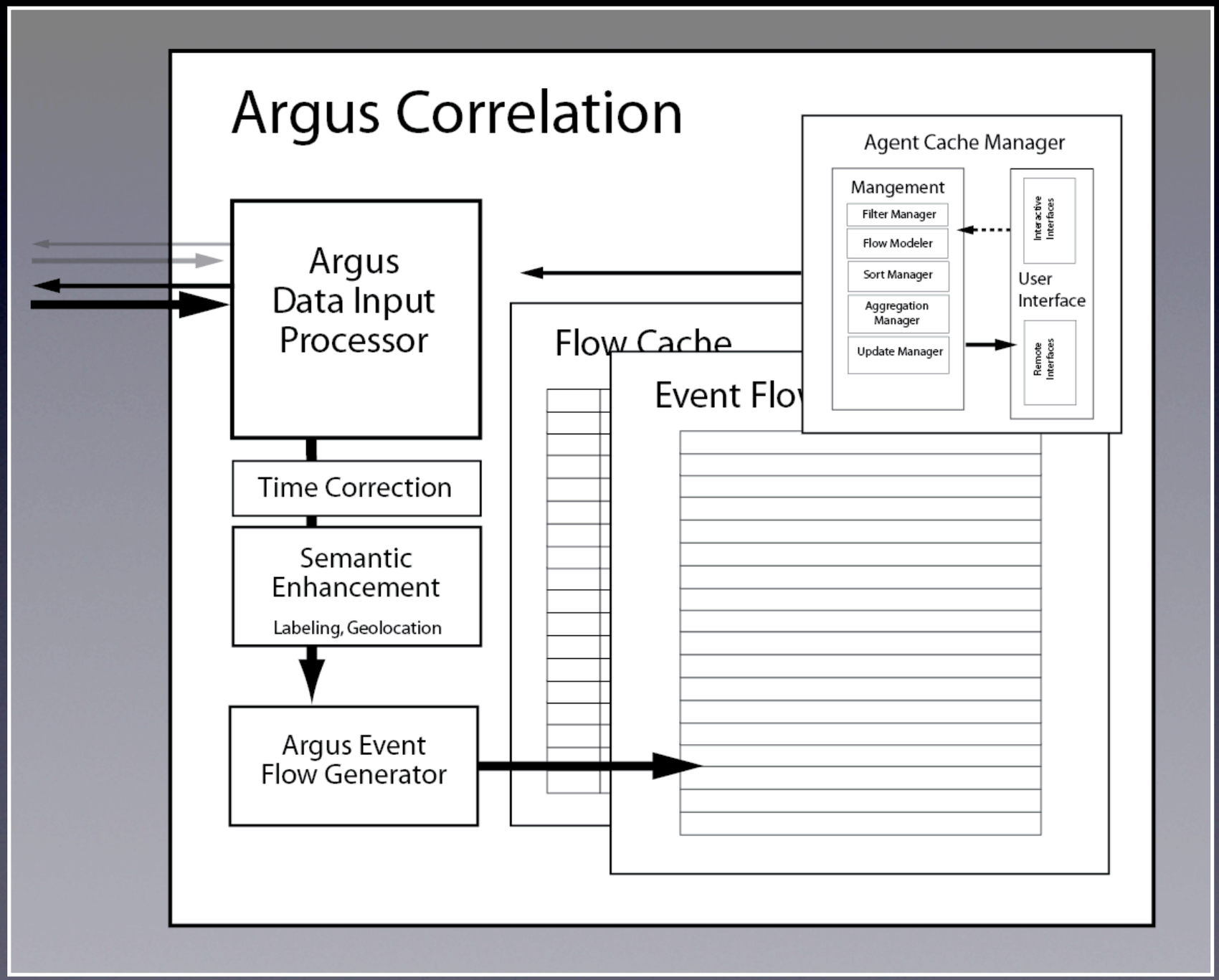
Argus Correlation Design

Radium Process



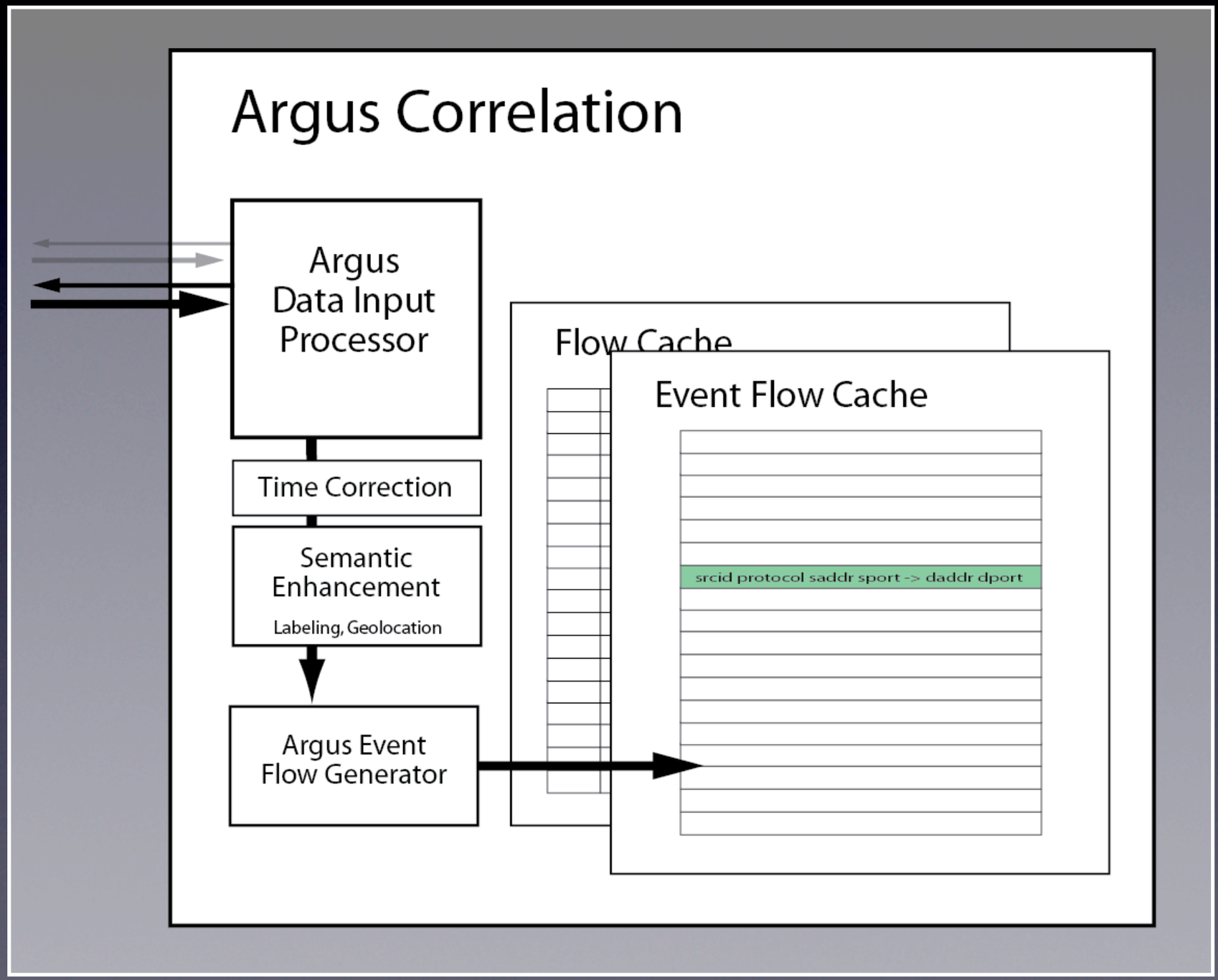
Argus Correlation Design

Radium Process



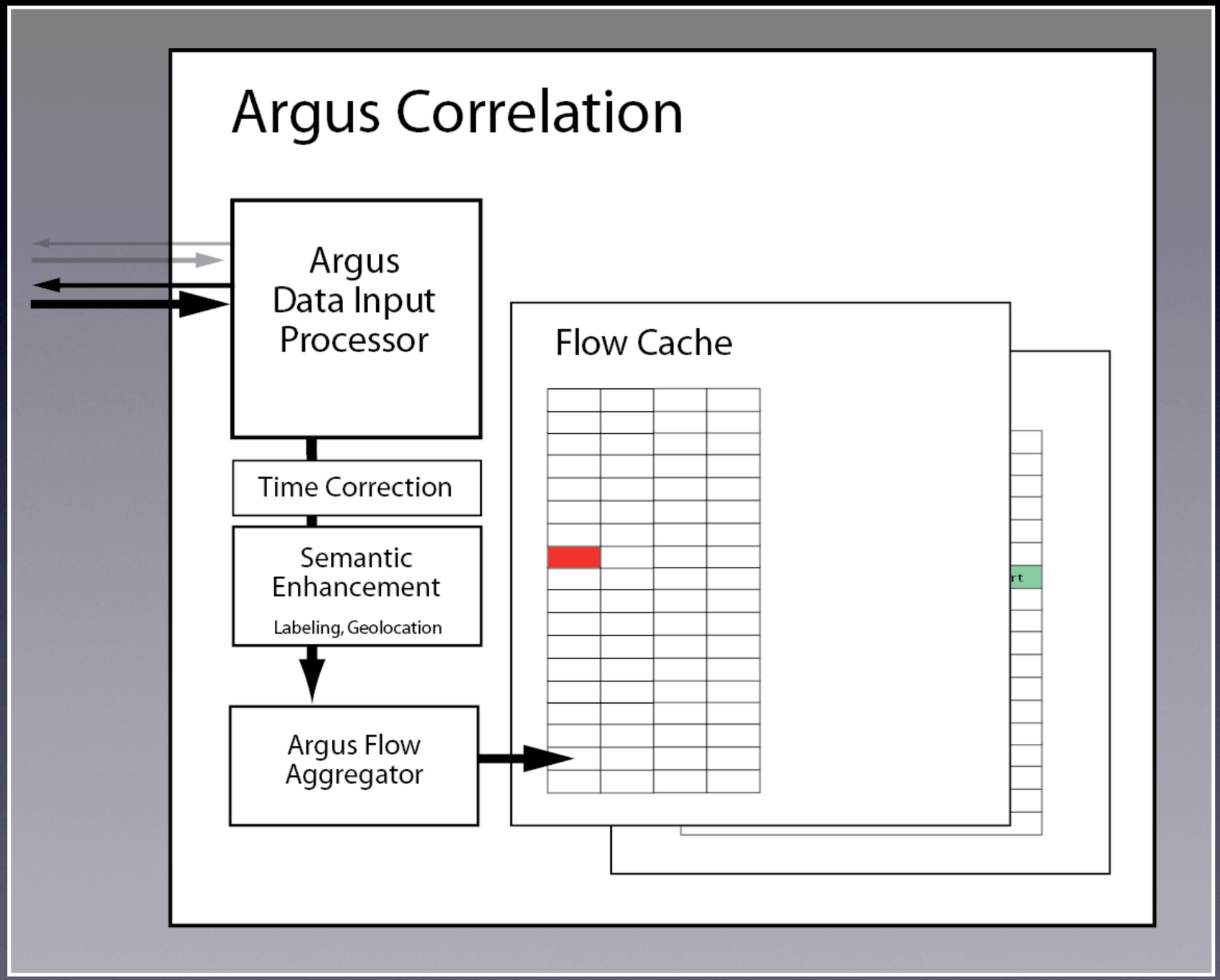
Argus Correlation Design

Radium Process



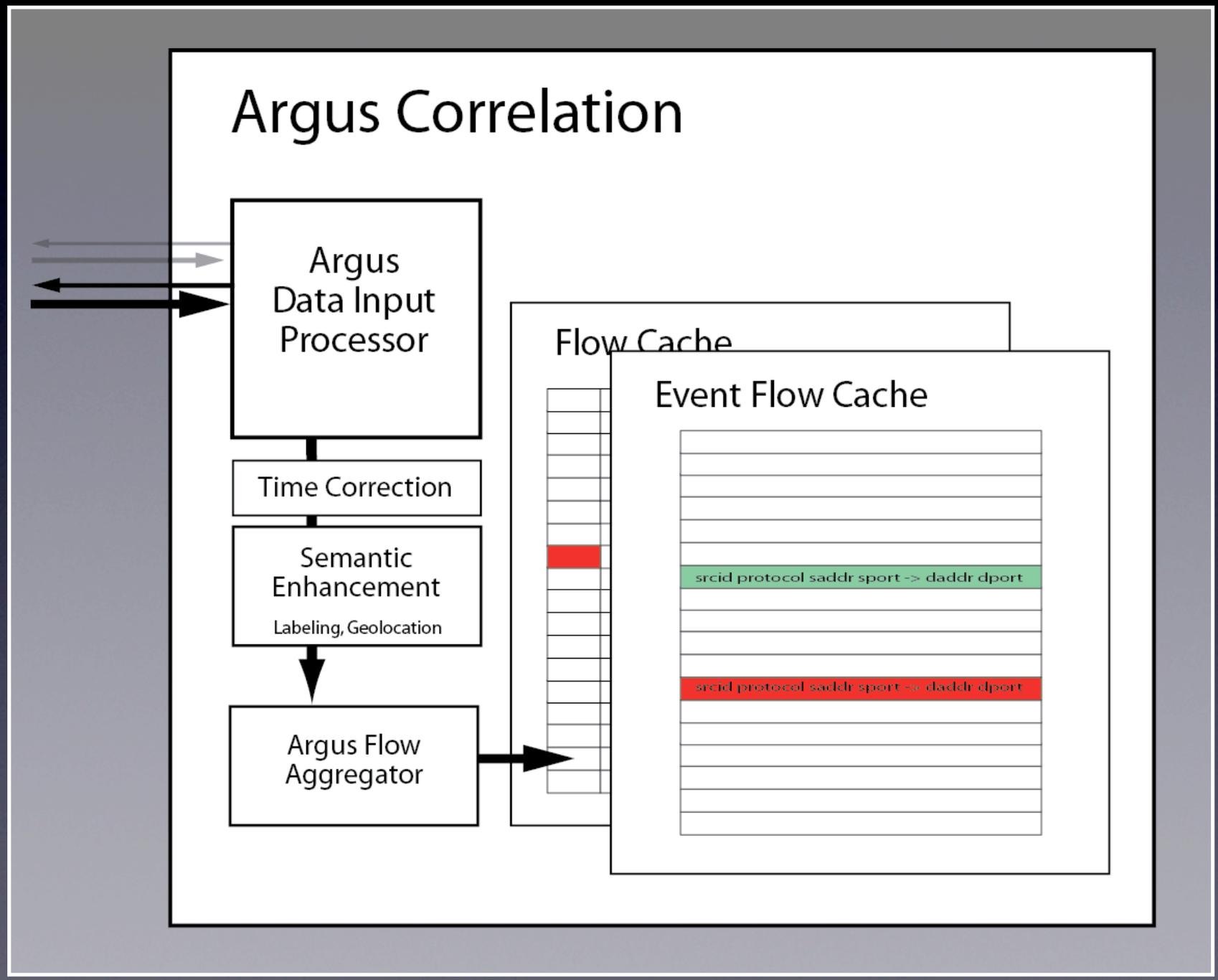
Argus Correlation Design

Radium Process



Argus Correlation Design

Radium Process



Argus Strategy

- Argus events processing generates flow descriptions and annotation labels that contain the user and the program
- We append these labels to the record.
- And then process like any other flow record

- Lot of rules on how argus labels work.

- Argus Metadata Tutorial has a lot of stuff on this topic.

Live Demonstration

from Presentation Laptop

ra and ratop screens showing live traffic as observed from the laptop

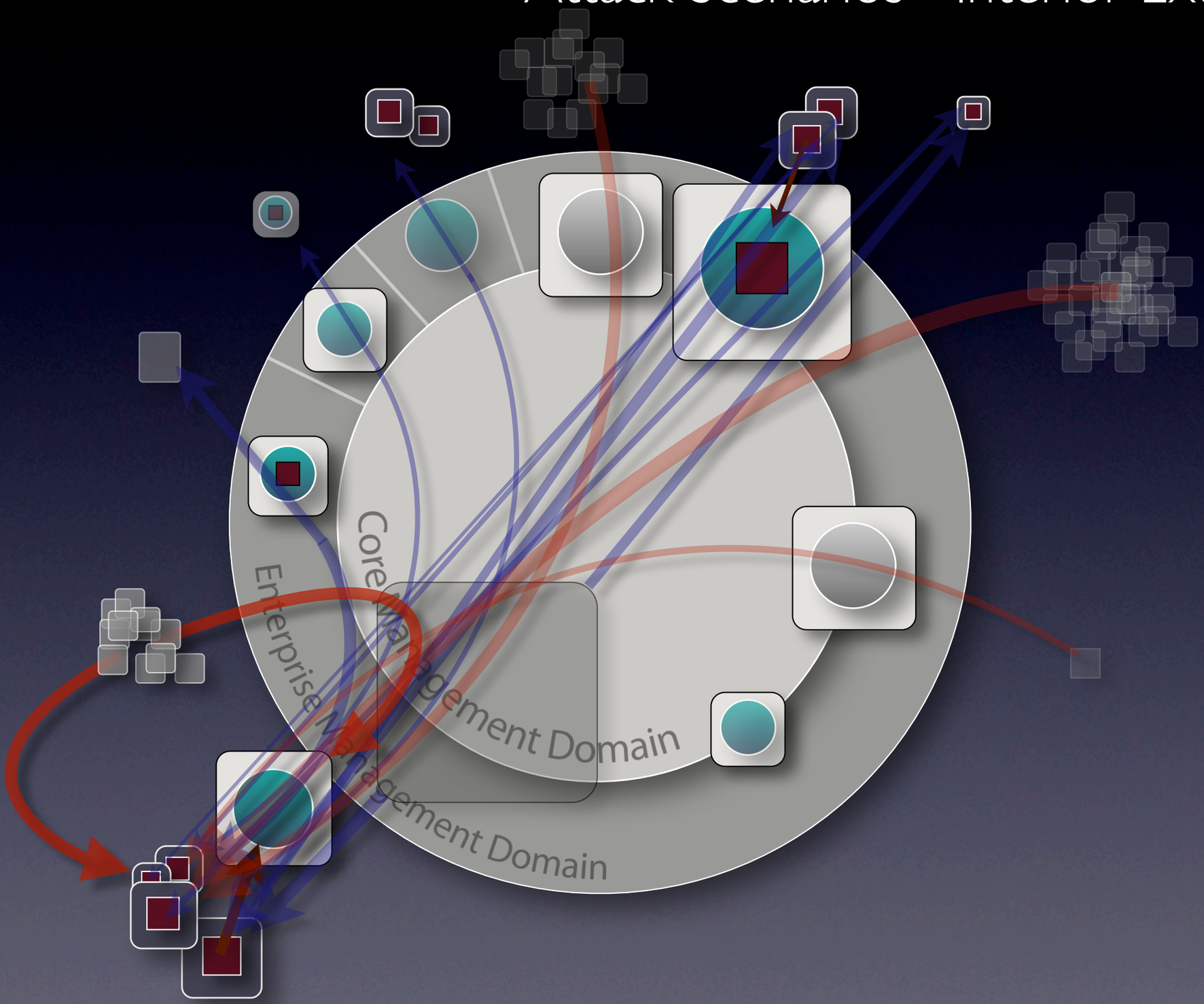
and realtime labeling of user, pid, program name

inserted into the flow record itself.

Supporting Slides

Distributed Situational Awareness

Attack Scenarios - Interior Exterior Spoofing



White/Visible Node

Black/Non-Visible Node

Data Plane

Command and Control
Attack Traffic

Spoof Correlation

- Simple multi-domain flow correlation
- However, with NAT, encryption, tunneling, traditional flow correlation is not possible.
  - No applicable flow identifiers for matching
  - Flow granularity mismatch
- Need flow metadata to make assessment
  - Content
  - Time
  - Packet dynamics (PD).
- Absence of correlation is the key
  - Statistical systems are unusable

Identifying Network Traffic Activity Via Flow Sizes

REDJACK

REDJACK

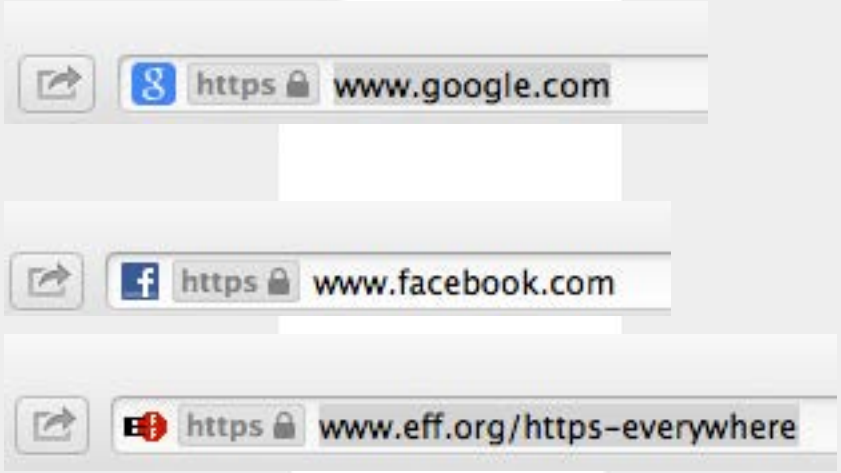
Overview

- Motivation – identifying activity via payload
- Theory behind the idea
- Measuring NetFlow
- Measuring DNS traffic captures
- Implications and future work

REDJACK

Motivation

- Users don't have the common decency to send plaintext all over the place anymore
- HTTPS prevalence
- OTR encryption for IM
- SSL for email



REDJACK

This Expands on Previous Work

- 2007 Paper on BitTorrent detection that focused on multiple behaviors – fumbling, file transfers, &c
- Now doing in depth study of control messages to see what we can find
  - Advantage – this time, have payload
- Questions:
  - Size of control messages
  - Distribution of control messages
  - Frequency of combinations?

REDJACK

Identifying Protocols Via Flow Sizes

- Hypothesis: traffic consists of three families of data
  - "Chatter"
    - Short (< MTU) , roughly symmetric packets of variable size
    - SSH, Telnet, IRC, ICQ, AIM
  - Transfer
    - MTU packets, met by payload-zero packets
    - FTP, Mail, HTTP
  - Control
    - < MTU packets, fixed sizes "fill in the blank" templates
    - All protocols

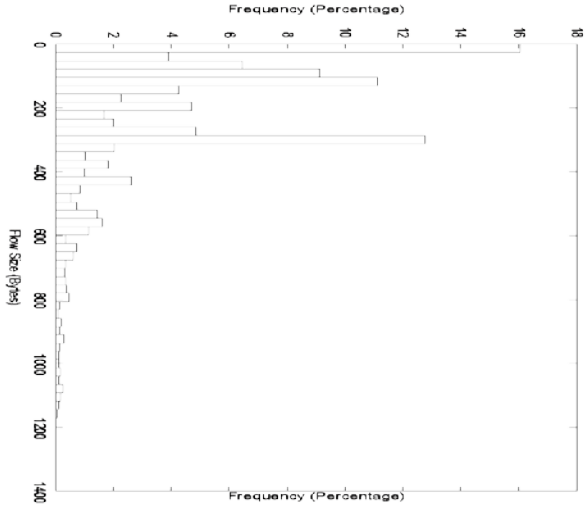
REDJACK

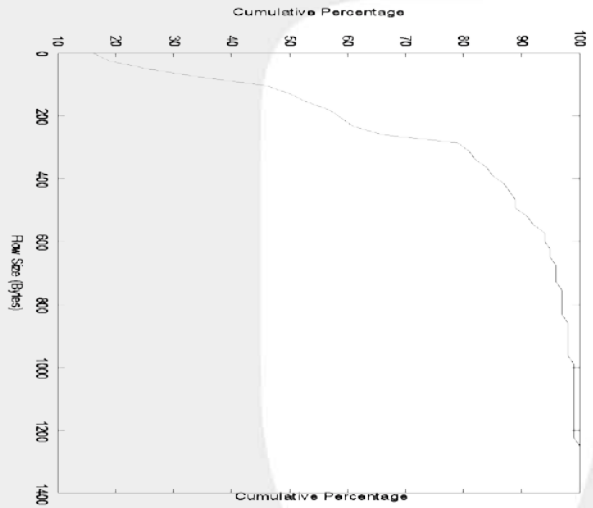
Differentiate Via Control Message Sizes

Histogram

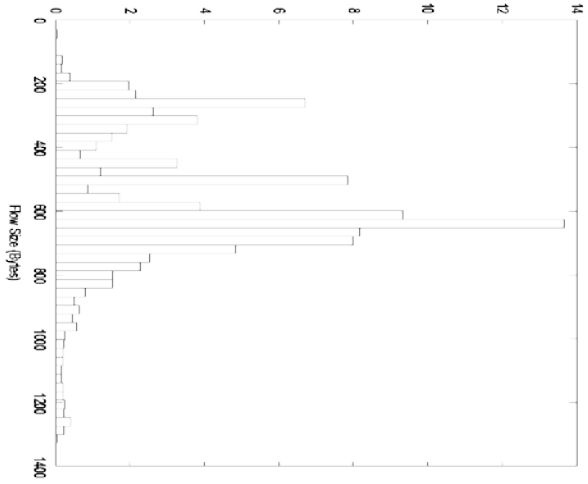
CDF

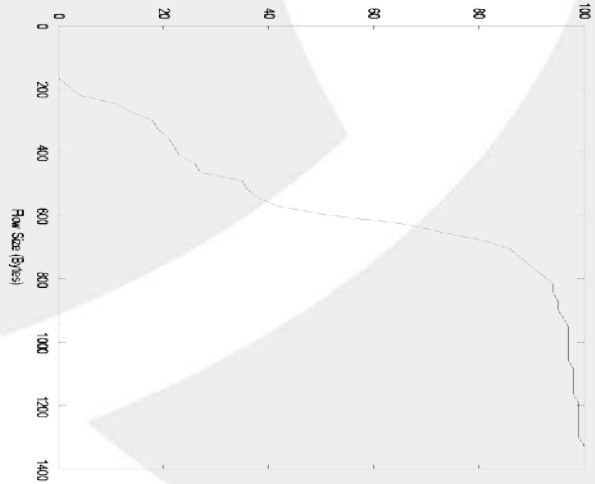
SMTP





HTTP





REDJACK

Done Some of This Already

- 2007 paper on p2p identification showed that you could find BitTorrent by looking for specific behaviors
  - Control packet sizes were one particular behavior
- However…
  - What are the actual packets?
  - What are the sizes
- Didn't have ground truth in previous work
  - Now have access to it via DNS records

REDJACK

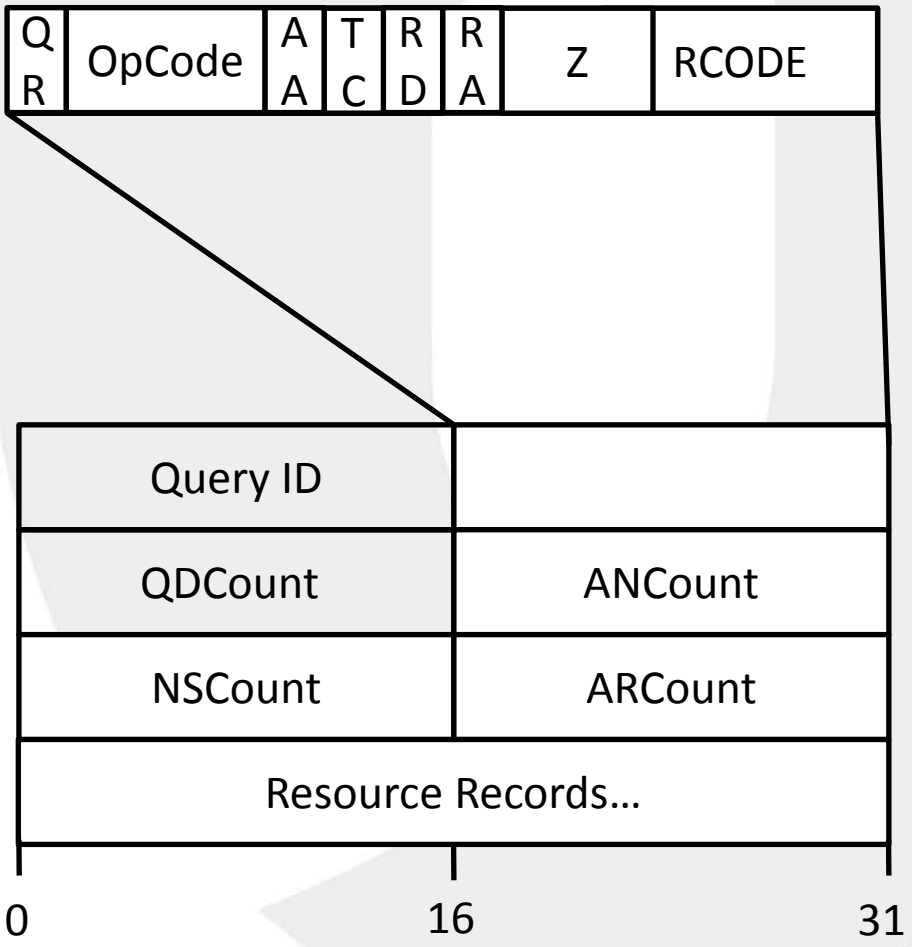
DNS Analysis

- Using DNS data, we can compare the exact messages sent against packet sizes
- See what messages produce what packet sizes
- Determine if we can predict messages via sizes
- Can't predict *content*, but we can guess what the user was looking for

REDJACK

The DNS Datagram

- State is maintained by Query ID
- Other flags set various info – authoritative, recursive, &c
- Response is sent in one or more RR's (resource records)



REDJACK

Resource Records and DNS Information

- DNS handles a *lot* of information
  - Name lookup
  - Name *ownership*
  - Authentication
  - Redirection
  - Email

REDJACK

Ripping Apart DNS Message Contents

- A DNS message contains 1 or more RR's (resource records)
  - Different RR's serve different purposes
  - Each RR has a different format, although most contain at least one variable length domain name
- Multiple different RR's may be sent to comprise a single message
- There's no requirement that the RR's actually be related to the original query, they may be annotative information
- ~40 RR's currently defined, including a couple of optional ones
- Responses are rarely just one message

Multiple Records Will Appear Simultaneously

	A	AAAA	CNAME	MX	NS	OPT	SOA	TXT
A	**99.33**	100.00	52.56	98.15	99.33	99.30	99.59	50.00
AAAA	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00
CNAME	0.69	0.00	**1.30**	0.00	1.30	1.36	0.00	0.00
MX	1.88	0.00	0.00	**1.90**	1.90	0.11	69.18	50.00
NS	100.00	100.00	100.00	100.00	**100.00**	100.00	100.00	100.00
OPT	49.56	0.00	52.03	2.94	49.57	**49.57**	0.41	50.00
SOA	2.65	0.00	0.00	96.29	2.65	0.02	**2.65**	50.00
TXT	0.00	0.00	0.00	0.05	0.00	0.00	0.04	**0.00**

- Table provides P(record of row type|record of column type); blue columns are P(record of row type)
- Some records (NS,A) are common
- Some (SOA) have a strong dependency P(SOA|MX)=96%
- Records will show up in group (5,10 NS records common)

REDJACK

What Are These Messages?

- A – IPv4 Address, 32 bit integer
- AAAA – IPv6 Address, 128 bit integer
- CNAME – Canonical Name, domain name string
- MX – Mail record, 16 bit preference value + domain name string
- NS – Nameserver name, domain name string
- OPT – Option record, variable option length
- SOA – 2 domain names and 128 bits of integers
- TXT – Variable length text

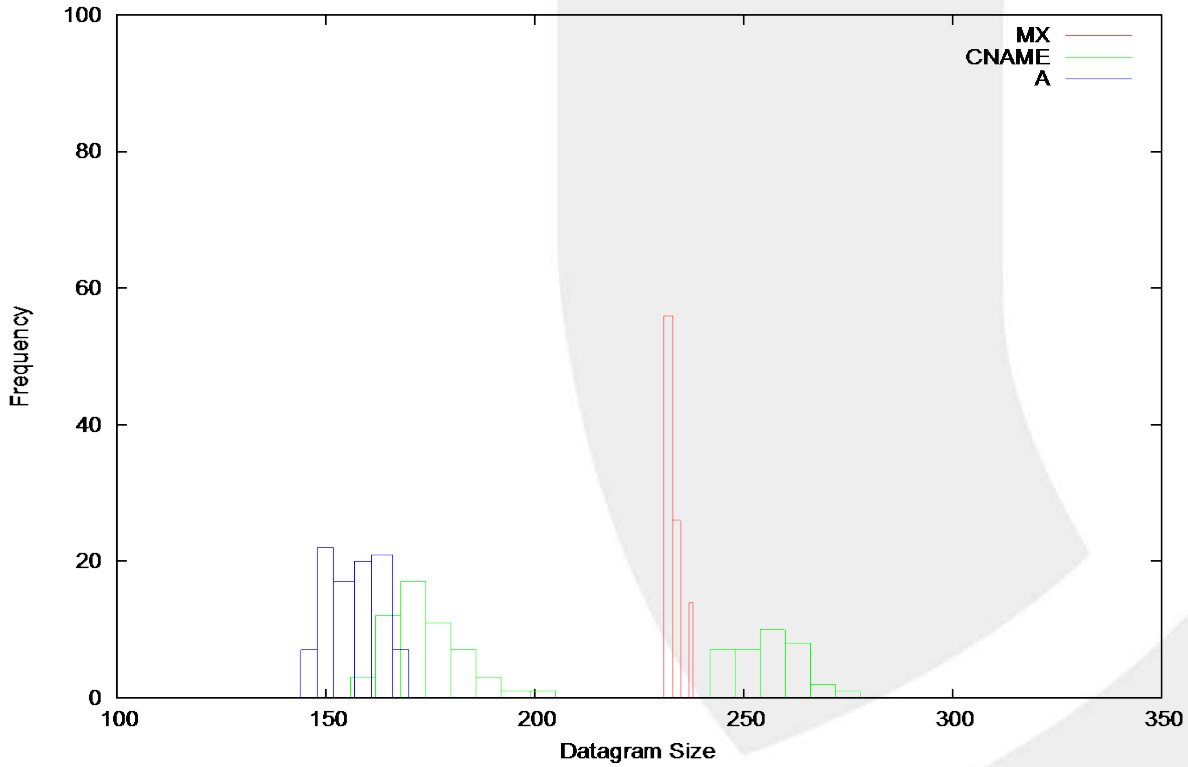
REDJACK

What Do We Do With DNS?

- Really, three major queries
  - Queries returning MX – Mail lookups
  - Queries returning CNAME – looking up aliases (CDN's love this)
  - Queries returning A on its own – simple lookups
- We can split out these queries and calculate frequencies for each one

REDJACK

Resulting In This



REDJACK

Observations

- Simple A records (least baggage) are smallest
- CNAME records broken into two groups
  - Differentiation is by NS records
  - 5 NS responses – smaller group
  - 10 NS responses – larger group
- MX is a very narrow spike (231-238 bytes)
  - Actual MX record is just a domain name, the rest of the offset is due to the SOA record

REDJACK

Conclusions

- Control messages in protocols can be used to differentiate the types of messages sent
  - We can use this information to differentiate protocols
  - Can use it to identify specific behaviors within protocols
- Variance in domain names is not significant enough to cause ‘overlap’ in messages
- Where can we go with this?
  - Facebook? Graph API? REST interfaces?
  - Markov Models?

Scalable Stacked Index to Speed Access to Multi Terabyte Netflow

Bruce Griffin
US-CERT

Summary

- In order to better optimize the analyst's workflow and to quickly dig into the > 26 GB/day of NetFlow day streaming into US-Cert, a scalable stacked index has been developed that identifies the when and where for any IP or collection of IPs. Basic statistics are also collected for each IP using Silk tools so that the analyst can quickly identify the government organizations involved, when the IP was seen, the type of flows (in, out , inweb, outweb, ext2ext, int2int) seen, the role that the IP played (source, destination), and how many times each type of flow from that IP was seen at each sensor each day.
- The time necessary to service an analyst's request is proportional to the number of days that the operator wants to review and to the number of days that any of the specified IPs are in flows within that review interval. If none of the IPs were seen within the review interval, the negative results are returned to the analyst in under 2 seconds. Positive results take about 4 seconds per day seen.
- The stacked index is defined to be N days deep, where N can be smaller or larger than the amount of historical flow data kept online. The indexing method cleans up after itself when it creates N+1 days of index, automatically removing the oldest indexes. By changing N, the index can easily grow or shrink as needed and the method to build up the index can be launched to build indexes of historical times not yet covered or to rebuild already covered indexes.

Agenda

- Size of collection
- Need for Speed
- Ops Floor Impact
- Cost
- Stacked Indexes?!
- Daily Index Content
- Many Sensors!
- Sensor Query
- Do the rwfilter pulls, if desired
- Examples
- Questions
- IP_Search Help
- IP set command summary

Size of Collection

- US-Cert, using Silk tools for Einstein 1, collects:
  - > 26 Gbytes of flow per day, currently 18 months deep
  - Total 18 months of flow > 14 Terabytes
  - 175+ active sensors
  - 2011- > 513 million unique routable IPs tabulated
  - 2012- > 998 million routable IPs in first 192 days
  - ~ 37 million IPs behind sensors
  - ~ 985 million IPs talking to Government
  - IPV6 additions complete, only 114K IPs so far(10/2012)

Need for Speed

- Typical question: Have we seen x in last 30 days?
- 30 days = 30 x 26Gb = 780 GB of flows to go through, can take hours
- Chris Hallenbeck Idea! - Build an index to identify when we saw x
- Silk tool IP sets to the rescue, version 2.4.1
- Desire quick negative response
- Elapsed time of query based on number of days v.s. volume of flows
- Information available before we dig into flows
- Try to limit user mistakes: date formats, position of keyword parameters, various spellings, IPs in various formats.

Ops Floor Impact

- 173 uses per week average
- 743 uses in September 2012 by 27 users
- 590 uses 1-24 Nov 2012
  - 491 used –s option for details
  - 12 used –r to get flow data
  - 538 used time relative option, -3 (days) most popular
  - 52 had specific time range
- Queries can be run in background
- IP_Search with flow data request: email you when all requested flow data combined in time order.

Cost

- 192 days of daily IP index takes 26 GB of disk space (Silk 2.4.1).
- Each day of sensor indexes takes ~ 1.2 GB
- Had to develop four “programs”
  - IP_Search to look in indexes
  - IP_Pull to build IP indexes, depth of daily index defined inside
  - Sensor_Pull to build sensor bags, depth of sensor index defined inside
  - Sensor_Merge to combine bags into text files

Stacked Indexes?!

- How to organize the index?
- Year/month/day
- Year/month/week/day
- 2,3,4,4,4 stacking for least number of IP sets to query
- 2 covers 192 days each index
- 3 covers 64 days each index
- 4 covers 16,4,1 days respectively

Stacked Index

- All IPV4 IPs for Y2012 stored in 2 sets
  - Y2012/S1.set covers first 192 days
  - Y2012/S2.set covers rest of year
- Y2012/S1/ has 3 sets, each covers 64 days:
  - S1.set, S2.set, S3.set and similar sub dir S1/, S2/, S3/
- Y2012/S1/S3 has 4 sets, each covers 16 days:
  - S1.set .. S4.set and S1/ .. S4/
- Etc
- Search computes best place to start to minimize index searches (1-2 indexes for negative test, more if found)

Daily Index Content

- Started with just an IP set for each day
- Later, modified to cover 6 categories of flow, Coded IS, ID, OS, OD, EX, IN
- Inbound, Outbound, External, Internal
- Source or destination
- Six IP sets per day at the single day level

Daily Index Query Example

```
>IP_Search -10 xx.35.11.2/31
Using IP indexing covering 2011/02/09-2012/08/01:12 and Sensor Indexes of 2012/04/23-2012/07/31
IP index covers 540 days.  Sensor specific coverage is 100 days.
2 IPs defined to search ...
    Searching from 2012/08/01 to 2012/07/22
set coverage found :

IP xx.35.11.2
    2012/07/22    IN,IS,OD
    2012/07/23    EX,IN,IS,OD
    2012/07/24    EX,IN,IS,OD
    2012/07/25    IN,IS,OD
    2012/07/26    EX,IN,IS,OD
    2012/07/27    EX,IN,IS,OD
    2012/07/28    EX,IS,OD
    2012/07/29    EX,IN,IS,OD
    2012/07/30    EX,IN,IS,OD
    2012/07/31    EX,IN,IS,OD
    2012/08/01    IN,IS,OD

IP xx.35.11.3
    2012/07/25    IN,IS,OD
    2012/07/26    IN
    2012/07/27    IS,OD
    2012/07/30    IN,IS,OD

Search took 29 seconds
```

Many Sensors!

- A single day pull is > 175 sensors x number of types (in, out, inweb, outweb, etc)
- One additional level of index: per sensor
- While we are at it, how about counting the number of times seen?
- 175 sensors x 6 categories = 1050 bag files to search!
- Can we dream up a faster sensor query?

Better Sensor Query

- Tabulate the sensor results for each IP and all sensors and categories
- N=16 text files, segmenting the IPV4 range such that each file is approximately the same size (bytes, not IP range). Similar segmentation for IPV6.
- A line in the file represents an IP and all sensor, category, count values seen in the bags
- Search time went from 40-50 seconds/day to 2-5 seconds/day.

Sensor Level Query Example1

```
>IP_Search -s -10 xx.35.11.2/31
…
2012/07/22
     BXX1M   xx.35.11.2     IS=24 OD=17
     DXXCV1  xx.35.11.2     IS=1 OD=1
     DXY1M   xx.35.11.2     IS=1
…
     DXYZ11       xx.35.11.2     IS=6 OD=3
     TXXXX6  xx.35.11.2     IS=2 OD=2
     TXXXX7  xx.35.11.2     IS=3 OD=3
     TXXXX8  xx.35.11.2     IN=62
     DXXX2   xx.35.11.2     IS=1 OD=1
     DX2     xx.35.11.2     IS=1 OD=1
     DX3     xx.35.11.2     IS=1 OD=1
     DX4     xx.35.11.2     IS=6 OD=6
2012/07/23
     BXX1M   xx.35.11.2     IS=63 OD=42
     DXZ1    xx.35.11.2     IS=5 OD=5
     DXXCV1  xx.35.11.2     IS=13 OD=13
…
     DXYZ1M  xx.35.11.2     IS=7 OD=7
     TXXXX6  xx.35.11.2     IS=64 OD=64
     TXXXX7  xx.35.11.2     IS=51 OD=51
     TXXXX8  xx.35.11.2     IN=1084
     TXXXX9  xx.35.11.2     IN=825
…
```

Sensor Search took 9 seconds

Sensor Level Query Example 2

- IP_Search -so -3 56.6.0.0/16 > using_56-6.txt &
- 65536 IPs to search for
- 4 days of IP index, 3 of Sensor index to search
- IP index search took 16 seconds
- Sensor index search took 6 seconds
- 164,000+ lines of output produced

Do the Rwfilter Pulls, if Desired

- Have the specific dates and sensors
- Perform a rwfilter pull for each day, which sensors seen each day, IPs searching for
- Run multiple rwfilter pulls in parallel if multiple days
- Merge everything together by time
- Results can be rwcut, IP sets, or raw flows recorded.
- User can add filtering criteria, change rwcut format, only see flows from specific organizations, ignore other orgs.
- New features added as needed: e.g. Talk2 to see flows between x and y, IPV6 searching.

Rwfilter Pull Example

- IP_Search proto=tcp org=txxxx,dxyz -r=packets=4- 2012/07/23-2012/07/22 xx.35.11.2/31 > pull_example.txt

- >more pull_example.txt
- Will include 4 sensors assoc with org txxxx
- Will include 2 sensors assoc with org dxyz
- …
- Sensor Search took 1 seconds
- Initial limit=2, code=[m]
- results in /analyst/home/bgriffin/dev/libsrc/Search8975.txt
- for [2012/07/22]
- cmd=[rwfilter --max-pass-records=1000000 --start-date=2012/07/22 --sensors=TXXXX6,TXXXX7,TXXXX8
  --anyset=/workspace/tmp/Search8975_1.set --type=in,out,inweb,outweb,inicmp,outicmp,innull,outnull,int2int,ext2ext
  --protocol=6 --packets=4- --pass=Fout-8975-2.dat]

- for [2012/07/23]
- cmd=[rwfilter --max-pass-records=1000000 --start-date=2012/07/23 --sensors=DXYZ1M,TXXXX6,TXXXX7,TXXXX8,TXXXX9
- --anyset=/workspace/tmp/Search8975_1.set --type=in,out,inweb,outweb,inicmp,outicmp,innull,outnull,int2int,ext2ext
- --protocol=6 --packets=4- --pass=Fout-8975-3.dat]

- Running multi rwfilter cmds in background...

Rwfilter Output

```
more Search8975.txt
                sIP|                dIP|sPort |dPort |pro|pkts  |                    sTime|    bytes|  flags|      dur|  sensor|     type|initialF|
       xx.35.11.2|xyx.123.213.170|32343|   443|  6|     11|2012/07/22T19:00:17.761|     911|FS PA  |   1.118|  TXXXX8| int2int| S      |
xyx.123.213.170|       xx.35.11.2|   443|32343|  6|     10|2012/07/22T19:00:17.763|    6021|FS PA  |   1.116|  TXXXX8| int2int| S  A   |
       xx.35.11.2|xyx.123.213.170|14325|   443|  6|      7|2012/07/22T19:00:25.483|     504|FS PA  |   0.327|  TXXXX8| int2int| S      |
xyx.123.213.170|       xx.35.11.2|   443|14325|  6|      5|2012/07/22T19:00:25.485|     346|FS PA  |   0.325|  TXXXX8| int2int| S  A   |
       xx.35.11.2|xyx.123.213.170|27612|   443|  6|     10|2012/07/22T19:00:25.859|    1397|FS PA  |  28.170|  TXXXX8| int2int| S      |
xyx.123.213.170|       xx.35.11.2|   443|27612|  6|      8|2012/07/22T19:00:25.861|    1596|FS PA  |  28.168|  TXXXX8| int2int| S  A   |
…
xyx.123.213.175|       xx.35.11.2|   443| 8092|  6|      9|2012/07/22T19:01:59.644|    1145|FS PA  |  29.417|  TXXXX8| int2int| S  A   |
       xx.35.11.2|xyx.123.213.175|17168|   443|  6|     12|2012/07/22T19:01:59.763|    2004|FS PA  |  29.298|  TXXXX8| int2int| S      |
       xx.35.11.2|xyx.123.213.175|33577|   443|  6|     11|2012/07/22T19:01:59.764|    3739|FS PA  |  29.298|  TXXXX8| int2int| S      |
       xx.35.11.2|xyx.123.213.175|25681|   443|  6|     20|2012/07/22T19:01:59.764|    5885|FS PA  |  29.299|  TXXXX8| int2int| S      |
       xx.35.11.2|xyx.123.213.175|16993|   443|  6|     12|2012/07/22T19:01:59.764|    2004|FS PA  |  29.298|  TXXXX8| int2int| S      |

      xx.35.11.2|      zyx.168.45.5|21268|    80|  6|      6|2012/07/23T22:03:29.117|    1085|FS PA  |   0.435|  DXYZ1M|  inweb| S      |
    zyx.168.45.5|        xx.35.11.2|    80| 21268|  6|      7|2012/07/23T22:03:29.121|    5288|FS PA  |   0.431|  DXYZ1M|outweb| S  A |
```

Questions?

- Email me at bruce.griffin@us-cert.gov
- See Also: 2012 Flocon presentation made by John McHugh entitled “Flow Indexing: Making Queries Go Faster”
- Info on Silk Tools: Google netsa or go to http://tools.netsa.cert.org/silk/index.html

IP_Search Help

- Help as of 07 October 2012 for IP_Search version 1.5 (found in /analyst/shared/scripts):

- keyword parameter: -h, --h, -s, org=xx,yy limit=nnx proto=xx -note -r 'xxx' or -r='xxx'
- The -h or --h option (or no arguments) gets you this help.

- -s or -S will give you additional information on the Sensors covering the IPs.
- The Sensor report will be presented below the normal IP and date report.
- If you are in a hurry, use the -so version, which will report JUST the sensor portion and
- not the normal IP report before it.

- -r option will auto run rwfilter cmd(s), supplying date range & IPs to search for.
- The use of -r will ASSUME a -s to produce better rwfilter performance, ignoring the sensors NOT coverd.
- If coverage is scattered, several rwfilter commands will be run to cover the times w/o excess searching.
- The results will be one file, in start time order.

- If you also specify -note, an email will be sent to your MOE account if it takes more than 2 minutes.
- You can also use -note=fred.smith,j.jones to send the notify to fred.smith and j.jones@us-cert.gov.

- As an additional feature of -r, if you want to see the flows between 2 parties, separate one party's IPs with
- the word talk2 or t2. All of the IPs to the left of the talk2 word will be index searched & tabulaated.
- An rwfilter command will pull all flows for those IPs. A second rwfilter command will then take those flows
- and pass them through another IP filter (anyset=) for all of the IPs to the right of the talk2 word. The end
- results being all flows between the IPs on the right and IPs on the left.

- org=xx,yy allows you to define the range of sensors for -r
- Normally, all sensors found in the indexes will be searched by rwfilter.
- As an example, using org=dos,treas,va will JUST search the
- sensors associated with Dept of State, Treasury, and DX.
- Additionally, if you do NOT want to see coverage for org xxx, code -not=xxx.

IP_Search Help (Cont)

- If you are interested in IP tabulations, use the -tab and -ip arguments.
- -tabsrc or -tabdst will produce an IP set of the source or destination Ips seen.
- -taball and -tabsall will produce a text file from rwuniq --fields=sip,dip.
- -tabdall will produce a text file from rwuniq --fields=dip,sip.
- -ipsrc or -ipdst will use the IPs that you entered as filters for the flows,
- selecting the src or dest IP as needing to match your Ips. (--ipall is default)
- Combining these options, you can get a listing of the good Ips talking to bad IPs on port 53 by
- IP_Search -r=dport=53 -ipdst --tabsrc <dates> <bad-IPs>

- limit=nnnx allows you to define the maximum number of pass records for -r.
- The nnnx operand allows you to specify a number (nnn) as well as a multiplier x=(k, m, g).
- The multiplier is optional and multiplies by 1000, 1 million, or 1000 million respectively.
- The default limit= is 2m. The minimum value is 200.
- The limit value will be split by n if n rwfilter cmds are performed.

- proto=x1,x2 allows you to quickly define the protocols for -r using simple names.
- The current names are: tcp, udp, icmp, esp, and eigrp.
- Normally, you get all of the protocols. If you enter proto=tcp Only TCP will be pulled.
- Alternatively, you can enter proto=-udp to get all BUT UDP.
- proto=tcp,udp would give you both TCP and UDP.

- Parms supplied will be added to command or just say -r= & We will supply the standard values.
- You can specify >output.txt to collect the results or
- we will collect tge output for you.

- Parms IP_Search is sensitive to:
- type=, protocol= change rwfilter characteristics
- pass=, fail=, all=, destination= truncates -r function to just rwfiter call, your named output file
- fields= will change rwcut output
- >xxx will direct rwcut output to your xxx file. NO spaces after >
- all other parms added blindly to rwfilter command.

IP_Search Help (Cont)

- PLEASE be patient. The initial query takes 3-5 seconds per day to search.
- Once the days have been found, it takes about 2 seconds per day for sensors information.
- The wider the time search, the more time it will take(if the IPs were found most days).
- The number of IPs does not affect the search time nearly as much as the time range does.

- positional parameters: {date_r} {IPs ... }

- date_r is SILK format and can be a single date, range, or -nn.
- i.e. 2012/04/23 or 2012/04/23-2012/05/11 or -20
- -20 will look twenty days into past.
- -2012/04/23 will look at each day back to April 23, 2012

- IPs are the IPs to search for, in one of three forms:
- 1) as text, with spaces between EACH IP.
- 2) as a file (.txt, .bag, or IP set) to get the IPs to look for.
- 3) redirecting a text file as STDIN. The contents can be of a mixture of form 1 and 2.
- e.g. IP_Search.pl 2012/04/23-2012/05/11 1.2.3.4 3.4.5.6 6.7.8.9 < theiplist.txt

- As IP_Search uses rwsetbuild, any format of IP that rwsetbuild likes is OK.
- so CIDR formats will work, integers will work, and Silk wildcard notation, like 10.x.1-2.4,5
- (10.x.1-2.4,5 will give you 10.x.1.4, 10.x.1.5, 10.x.2.4, and 10.x.2.5)a
- IP_Search will NOT handle IP range notation like 10.1.2.4-10.1.2.5
- Code it in wildcard notation 10.1.2.4-5

IP_Search Help (Cont)

- Examples of cmd:
- IP_Search 2012/04/23-2012/05/11 1.2.3.4 3.4.5.6 6.7.8.9
- IP_Search -20 ip_tabulation.set
- IP_Search -r= 2012/04/23-2012/05/11 1.2.3.4 3.4.5.6 6.7.8.9 talk2 41.215.45.0/24
- IP_Search -2012/06/01 ip_tabulation.txt

- Results will include codes to better identify the types of flows found.
- If running a basic search (no -s), each date found may have these codes:
- IS to denote that the IP was found on an Inbound flow as the Src IP.
- ID to denote that the IP was found on an Inbound flow as the Dest IP.
- OS to denote that the IP was found on an Outbound flow as the Src IP.
- OD to denote that the IP was found on an Outbound flow as the Dest IP.
- EX to denote that the IP was in an ext2ext type flow.
- IN to denote that the IP was in an int2int type flow.
- If the -s option is used, the same codes may be present by each sensor and
- a count of the number of flows of that type will also be present.
- i.e IS=2345 OD=256

- The following sensor groups are no longer active
- DXXHQ last seen 2009/08/13, Use DXXCV instead
- MICH last seen 2010/11/24
- NLRA last seen 2011/09/09
- MCI last seen 2011/05/19
- IDC last seen 2010/03/18
- COMM last seen 2012/05/19, Use HCHB instead
- TSA last seen 2009/08/13

IP set Commands

- Rwset take IPs from flows, build set(s)
- Rwsetcat look at or count # in set
- Rwsetbuild create an IP set from text
- Rwsettool set math
  - Union A + B
  - Intersect c = A also in B
  - Difference c = A - B
  - Sample neat way to rewrite an IP set

Bags have similar commands

DAMBALLA

Take Back Command-and-Control

Detecting Malware P2P Traffic Using Network Flow and DNS Analysis

John Jerrim

FloCon 2013

Overview

- More malware using P2P protocols for command and control
- BotTrawler, a research tool for detecting and classifying P2P traffic
- Use of Protocol Transaction Analysis for detection of P2P protocols
- Detection of ZeroAccess and TDLv4 using PTA
- Examination of Zeus using swarm analytics

The Problem: Malware Using P2P

- **Malware toolkits are including P2P as a means to avoid use of DNS for command and control. Examples include:**
  - Zeus v3
  - TDL v4 (Alureon)
  - ZeroAccess
  - Thor (??)
- **We have observed roughly a 10x increase in the number of malware samples using P2P in the past 12 months**

BotTrawler

- **A network flow and analysis research system that fuses multiple data sources including:**
  - YAF for flow creation and payload analysis
  - Associate DNS lookup with flows
  - Reverse DNS &Passive DNS for flows w/o DNS lookups
  - Geo-Location
  - Reputation
    - Public blacklists / spam lists
    - Private blacklists from DNS convictions
  - Binary file analysis
- **Active research project, but some aspects are being weaponized at this time.**

BotTrawler Processing

- **Identify possible P2P flows and group into "P2P sessions"**
- **Create features for classification based on flow, session, and multi-session analysis**
- **Classify vs. known (labeled) P2P applications for both benign and malware P2P**
  - If known, ignore or alert as appropriate
  - If unknown, cluster with other unknowns and test for suspect malware attributes

P2P Classification Feature Attributes

- **Scalable for high speed analysis**
- **No payload analysis (it's encrypted anyway)**
- **Robust Detection – High True Positive, Low False Positive**
- **Make detection avoidance expensive**
  - Require a protocol change rather than a simple port change, for example
- **Use features the enemy cannot easily control or manipulate**
  - Swarm member characteristics are good features
  - Flow rates and periodicity (automation detection) may be useful but are weaker features

Protocol Transaction Analysis (PTA)

- **Based on features created by examining the number of packets and payload exchanged between the local asset and the P2P swarm members via TCP and UDP**
  - Highly repetitive transaction sequences are readily observable with P2P as there are hundreds (or more) connections (think “connection handshakes”)
  - Easily processed and clustered
  - Typically use 3 to 5 unique transaction sequences to identify a P2P application to handle different command/ response sequences in the protocol
  - Some applications require multiple sets of transaction sequences for different behavioral aspects of the application

Connection Related Attributes

- **Connections to external IP addresses**
  - Focus on unique and rare connections
  - Repeated connections to external Ips
  - Avoid use of DNS
- **Swarm analysis**
  - Geographic dispersion
  - Session to session swarm overlap for same asset
  - Swarm overlap with other suspicious or malicious P2P from other assets

Possible Malware Attributes

- **Swarm members often have other malware installed**
  - % of swarm members on spam lists is generally significantly higher than the "noise level" of benign P2P swarms
- **The geographic distribution of swarm members is generally different than benign P2P swarms**
- **Hybrid P2P applications**
  - Hybrid uses a public network for resiliency and a private network as primary C&C
    - Menti (first observed January 2011) appears to be an example of a hybrid P2P: Uses both Tor and P2P

Additional Malware Attributes

- **Contextually associate P2P traffic with other malware behavior associated with the asset:**
  - P2P traffic begins shortly after (often within seconds) of a suspicious file download
  - Other suspicious activity may also be noted starting near or after the compromise (differential asset behavior):
    - Spamming
    - ClickFraud Activity
    - DoS participation

P2P Classifications

- **General Purpose P2P**
  - BitTorrent
  - eMule
  - Tribbler
  - And many others…

P2P Classifications

- **Specific Purpose P2P**
  - Benign or commercial
    - Skype
    - Spotify
    - And many others
  - Malware
    - ZeroAccess
    - Zeus v3
    - TDL v4
    - And a few others

Specific Purpose P2P

- Are often easily identified by DNS, reverse DNS or passive DNS means as they generally do not try to hide – unless they are malicious
- Swarms are often small (< 100) with some or significant overlap of swarm members between P2P sessions
- Swarms may be highly localized. For example, Spotify uses minimal distance algorithms to reduce propagation delays

General Observations

- **All members of a malware P2P swarm have been compromised with the same malware**
  - Detect one and you will quickly identify hundreds up to tens of thousands of compromised assets
- **P2P Protocols are reused by malware operators. TDLv4 uses the identical P2P protocol as ZeroAccess**
  - Identifying the technology and may identify the primary operator behind the malware, but may not identify the exact compromise

ZeroAccess

- **A rapidly growing click-fraud botnet that uses significant user bandwidth**
  - Over 2 million nodes estimated world-wide in November, 2012
  - Makes extensive use of P2P
  - Appears to be closely related to TDL v4 as it uses the same P2P protocol

Detecting TDLv4 and ZeroAccess

- **Using PTA as primary detection mechanism**
  - Created transaction sequence sets for three variants of the protocol as "labeled data" for the test
  - Simple decision tree for detection:
    - Sequences must be in the "top 5" for the P2P session
    - Three or more unique transaction sequences must be observed
    - Of the three, two must be bidirectional transaction sequences
    - Rank ordered detection is preferred for high confidence

PTA Detection Test Results

- **182,097,625 P2P flows clustered into 132,015 P2P Sessions over a six day period**
  - 168,188 flows in 86 P2P sessions on 49 assets were identified as malware using P2P. All 49 assets were confirmed as infected by the customer (100% True Positive)
  - Transaction Sequence Statistics:
    - An average of 1955 labeled transaction sequences were observed for the P2P sessions classified as malware
    - An average of 1188 labeled bidirectional transaction sequences observed per malware P2P session
    - Only 909 labeled transaction sequences were observed in the remaining 131,992 P2P sessions – all unidirectional
    - There were zero(!) labeled bidirectional transactions observed in the 131,992 non-malware P2P sessions

Zeus v3 BotNet

- Zeus is a botnet focused on banking and financial theft. Use of P2P started early in 2012 when v3 was released.
- Provides a good example of repeated swarm membership for a period of time. Identical swarms have not been observed on benign P2P applications.
- There is a strong indicator of a download containing a list of new swarm members followed by changes in subsequent swarms
- Swarm members exhibited significantly higher spam list rates than background noise.

DAMBALLA
Take Back Command-and-Control

Zeus Multi-Session Swarm Statistics

Session Start	LastTime	IntPkts	IntPayload	ExtPkts	ExtPayload	New IPs	Total
3/15/12 18:34	3/15/12 18:39	950	23912	905	12366	28	31
3/15/12 18:56	3/15/12 19:09	920	17310	901	8020	1	33
3/15/12 19:25	3/15/12 19:39	944	23532	871	8758	1	33
3/15/12 19:55	3/15/12 20:14	1623	26570	1570	8436	0	33
3/15/12 20:30	3/15/12 20:44	1022	36858	1213	136488	9	37
3/15/12 21:07	3/15/12 21:19	890	23240	829	7778	0	29
3/15/12 21:35	3/15/12 21:54	1780	26268	1744	8412	0	31
3/15/12 22:12	3/15/12 22:24	896	15542	888	9032	0	27
3/15/12 22:40	3/15/12 22:59	1724	29314	1648	7962	0	30
3/15/12 23:15	3/15/12 23:29	900	16298	867	6924	0	25
3/15/12 23:45	3/16/12 0:09	2762	72408	2884	162204	37	73
3/16/12 0:26	3/16/12 0:44	1812	35898	1726	9186	0	38
3/16/12 1:00	3/16/12 1:19	1820	29488	1966	102296	0	38
3/16/12 1:37	3/16/12 1:54	1744	27976	1665	8660	0	37
3/16/12 2:10	3/16/12 2:24	951	21502	898	7180	0	29
3/16/12 2:46	3/16/12 2:59	888	17254	1043	82294	0	26
3/16/12 3:16	3/16/12 3:29	966	31184	1128	117210	7	33
3/16/12 3:50	3/16/12 4:04	932	21334	1059	86596	0	28

Key Results

- Identifying new P2P malware works best when intelligently fusing data from a broad range of data sources including network flow and derived features, DNS, binary analysis, swarm analysis, differential behavioral analysis, and reputation systems.
- PTA shows great promise for extracting new information from network flow data to aid in malware and application detection.
- Multi-session swarm analysis provides additional insight into how the botnet is being utilized.

Questions?

?

john.jerrim@damballa.com or on LinkedIn

Enhancing Network Situational Awareness Using DPI Enhanced IPFIX

Hari Kosaraju

Prepared for Flocon 2013

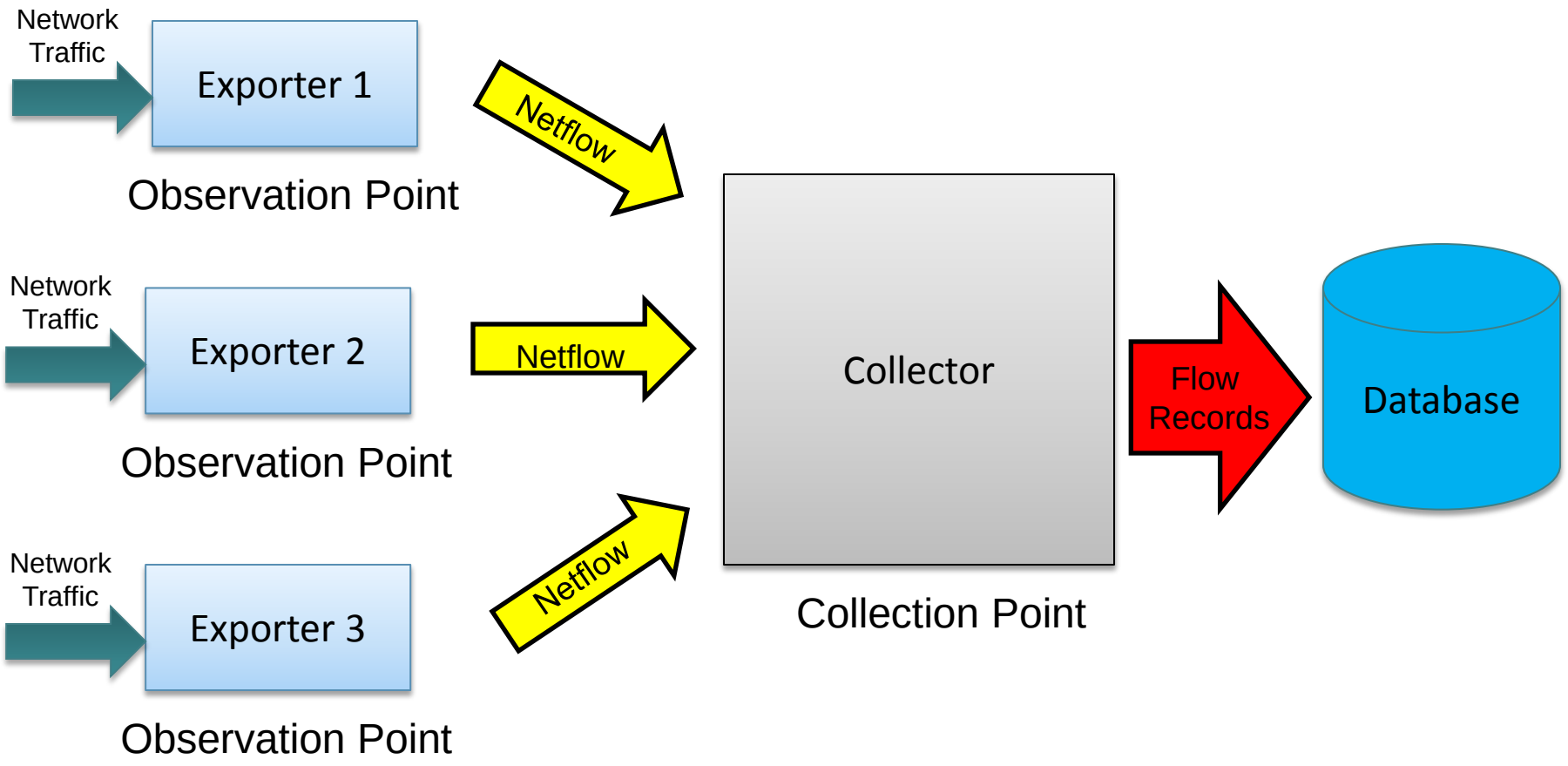
Agenda

- What is the difference between Netflow v5/v9 and IPFIX?
- How can we improve flow based traffic visibility?
- IPFIX Format for SessionVista
- How does this enhance Network Situational Awareness?
- Our implementation

Netflow Introduction

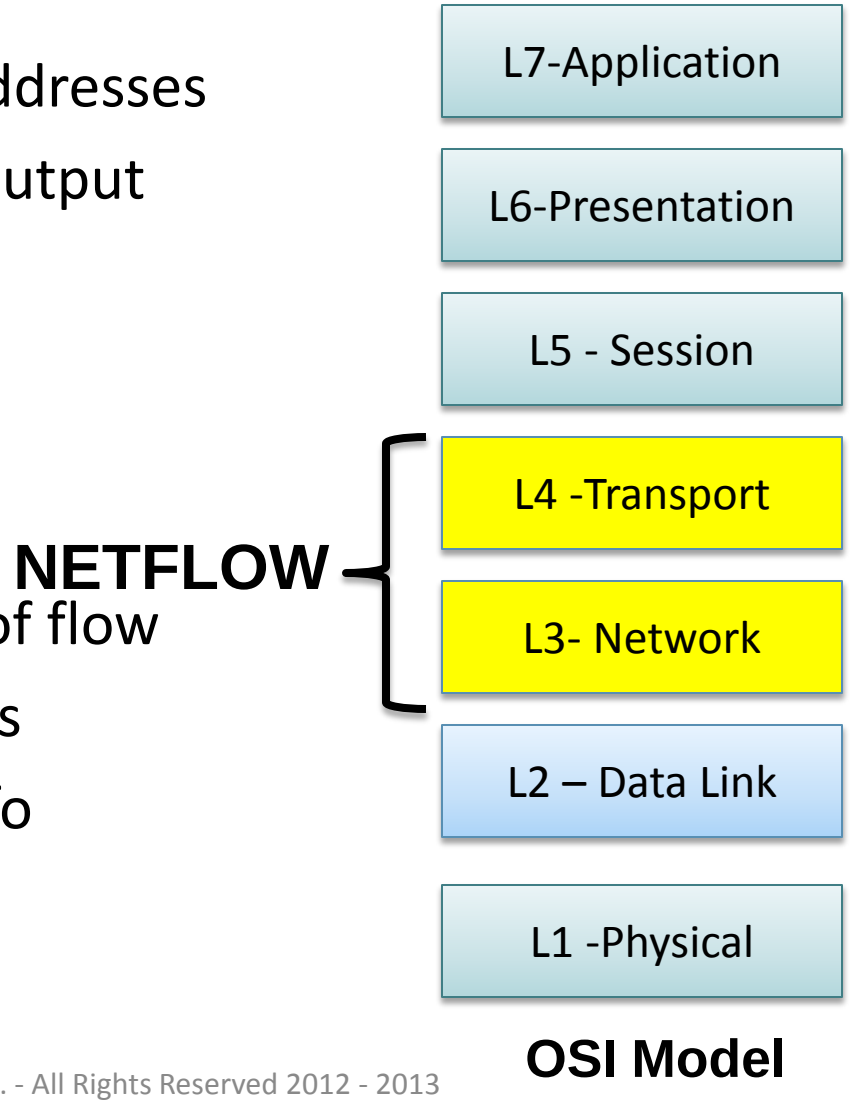
- Netflow is a protocol that was introduced by Cisco and is used for flow reporting on network traffic
- Information is typically reported on a flow basis, rather than on a packet basis
- However it is possible to report on packets via sampling
- The two popular versions are Netflow v5 and Netflow v9
- Other equipment vendors have their own variants but they are similar

Current Monitoring Paradigm



Information Reported in Netflow v5

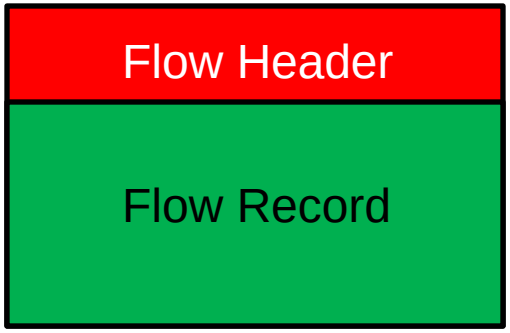
- Source and Destination IP addresses
- SNMP indices of input and output interface
- IP address of next hop
- Packets in the flow
- Total L3 bytes in flow
- Sysuptime of start and end of flow
- Source and Destination ports
- IP protocol, TOS, TCP flag info



The difference between Netflow v5 and v9

- Netflow v9 added support for IPv6 addresses
- **Concept of a template was introduced in Netflow v9**
- A template is a packet that is used to describe the structure of subsequent Netflow packets of the same identifier
- It is like a recipe that tells the Collector the format of the information to follow
- The advantage of this scheme is that the data sets are purely an identifier and associated data. They do not have any other parsing information which makes transport more efficient

Netflow v5

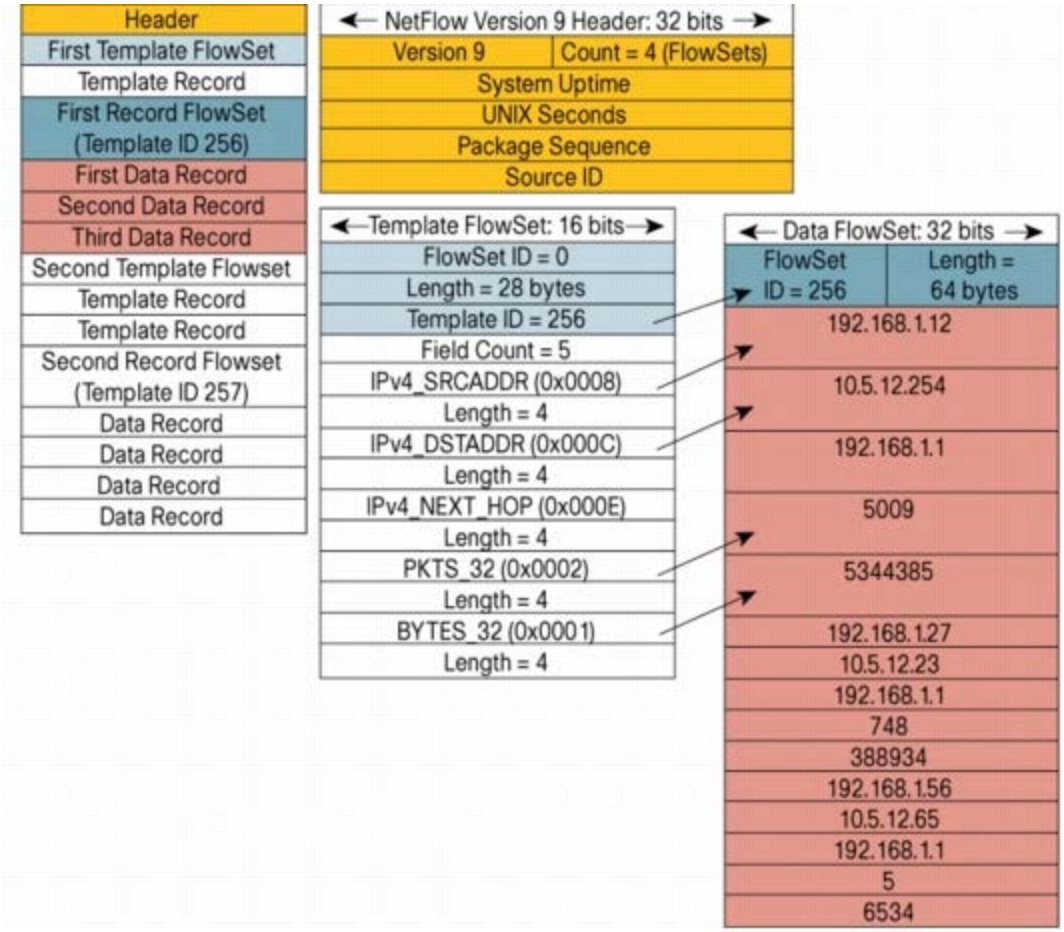


Fixed Format

Netflow v9

Extensible Format

Netflow v9 Structure



Source: Cisco.com

Pros and Cons of Netflow

Pro	Con
Gives flow level traffic visibility which enables numerous applications	Adds processing load to routers and switches
Reports on L3 and L4 information as well as flow timing	Is often run in sampled mode to reduce strain on the router and misses fidelity on small flows
Reports on flow length	Higher layer visibility limited to IP protocol field
Supported on many different networking devices natively	**Most analysis is based on ports**
	Does not handle tunneled traffic
	Only reports L3 and L4 metadata

How Do We Address These Issues?

IPFIX Introduced in 2008

- IPFIX was standardized by the IETF in Jan 2008
- It uses the template based approach started in

Netflow v9

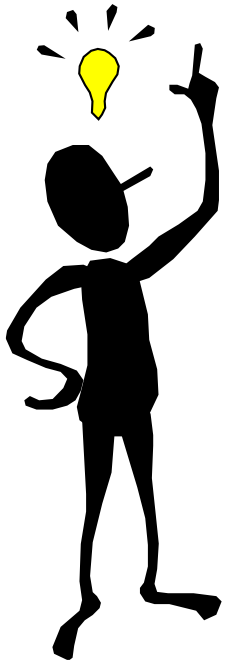
- Completely self-contained in that it adjusts the data format as new elements are added
- Added Two Very Important New Features:
    1. **An Enterprise specific field**
    2. **Variable length fields**

It is space efficient and gives us flexibility to include Enterprise specific data!

Problem:
We need deeper traffic visibility

- We'd Like to See More Than L3 and L4 Metadata for better Network Situational Awareness

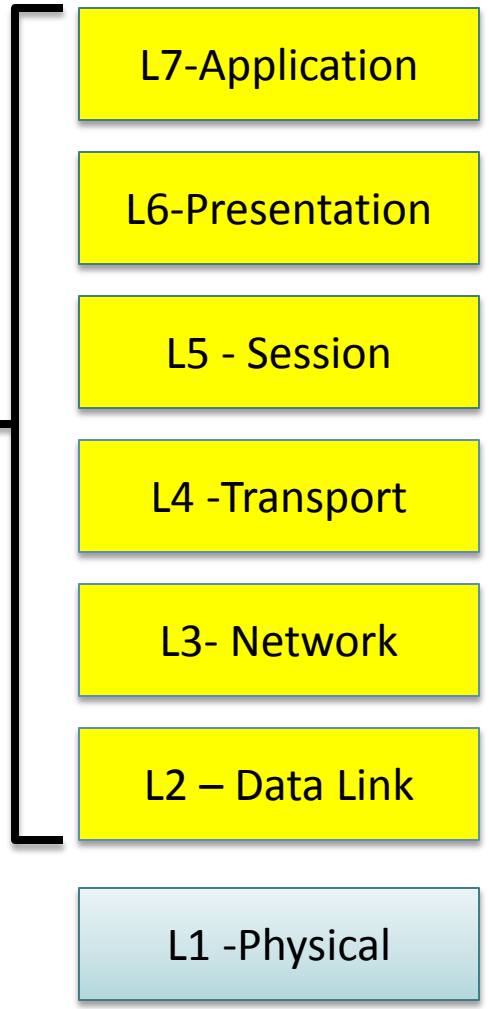
- Combine Deep Packet Inspection with IPFIX!



Deep Packet Inspection for L2 through L7 Visibility

- IPFIX has enterprise specific fields
- SessionVista has created one to encapsulate metadata extracted through Deep Packet Inspection
- What we do is report session level metadata using an IPFIX enterprise specific field
- The DPI engine can extract application layer metadata from different protocols (700 protocols and over 4000 metadata attributes)

SessionVista IPFIX



SessionVista IPFIX Format Design Challenges

1. Must work with multiple protocols and multiple attribute types
2. Must be able to handle transactional situations within a flow:
    1. Multiple emails within an SMTP download
    2. Multiple attachments on each email
3. Needs to encode data efficiently and perform an information reduction exercise (100: 1)
4. Report on flows with little to lots of metadata
5. Handle attributes that appear multiple times in one flow
6. Handle tunneled protocols and deep protocol stacks
7. Must be easy to add new protocols and attributes without changing the protocol

SessionVista IPFIX Format

- 64 Bit Flow Identifier to identify a bi-directional flow
- Generate an IPFIX report on the end of a session (bi-directional flow)
- Multiple IPFIX packets can be used to report on one Session
- Transactions within a flow are handled using a sub identifiers and transaction identifiers

- Every protocol we report on has a template
- Each template has a number of fixed elements and a variable data field
- All integer encoded data is placed in the fixed portion of the IPFIX packet, unless, it is data that can happen multiple times within a session.

SessionVista
Enterprise Specific Field

- Encodes Protocol Attributes using identifier, length and value semantics.
- This way, any number of variable length data items for a protocol can be stored.

[email-protocol-data]
// fixed fields (packet count and duration)
[2(attachments)**] [43** (seconds)**]**
// variable data field
[<length=65>
<emailSender><len=14>"alice@home.com"
<emailReceiver><len=12>"bob@work.com"
<emailReceiver><len=14>"carol@work.com"
<emailSubject><len=17>" Fwd:Status Report"]

Encoding Example

IPFIX Header
[flowID-data] [12abcd90-12efabcd (flowID)] [0000000013414111 (totalSysPackets)] [0000000000000013(totalFlowPackets)] [0000000049c1901e000000000003c4cb(flowStartTime)] [0000000049c1901f000000000006bd2b (flowEndTime)] [0001 (flowStatus)] [11626173652e69702e7463702e6874747000 (flowPath)]
[ip-protocol-data] [192.168.1.2 (source)] [172.16.17.23 (dest)] [200 (ttl)] [6 (tcp protocol)] [0x0123 (flags)]
[tcp-protocol-data] [3123 (source port)] [80 (dest port)]
[http-protocol-data] *// variable data field* [<length=80> <httpUrl><len=26>"books/list/bestsellers.htm" <httpServer><len=14>"www.amazon.com"]

Note: flowPath translates to base.ip.tcp.http

So what are the advantages?

- 15 layer deep protocol decode of 700 protocols along with per flow statistics
  - i.e. `"base-eth-ip-udp-gtp-tcp-http"`

- 4000 Metadata attributes from 700 protocols that are completely configurable

- Much richer dataset than existing tools without the storage costs associated with packet capture

- Easy to add new protocols and attributes without changing the IPFIX implementation

How does this enhance Network Situational Awareness?

- Survey the network and see things like:
  - Tunneling for obfuscation
  - DDoS (Application layer) detection
  - Network Asset Inventory : What is on your network and what services are they running (port agnostic)
  - Statistics provide the ability to perform application performance monitoring
  - Anomaly Detection (traffic trends, policy violation, data exfiltration)
  - Beaconing detection
- Alert on traffic events as they happen rather than doing a retrospective analysis of packet captures

SessionVista

How have we implemented this?

- IPFIX Exporter implemented in a multi-threaded Linux implementation
  - Scalable to over 10 Gbps in a 1U platform
  - Support for both Napatech and PCAP interfaces
  - Configurable to many multi-core architectures and memory requirements
- IPFIX Collector implemented in C++ on Linux
  - Accepts multiple connections
  - Scalable multi-threaded implementation
  - Backend to log file, MySQL, Hypertable and CEP engines

SessionVista

Thank You!

Hari Kosaraju

hkosaraju@mantaro.com

www.sessionvista.com

Scalable NetFlow Analysis with Hadoop

Yeonhee Lee and Youngseok Lee

{yhlee06, lee}@cnu.ac.kr

http://networks.cnu.ac.kr/~yhlee

Chungnam National University, Korea

January 8, 2013

FloCon 2013

Contents

- Introduction
- Overview
- Hadoop-based traffic processing tool
- Evaluation
- Summary

INTRODUCTION

Internet Measurement

- Challenges
  - Scalability
  - Fault-tolerant system
  - Extensibility
- CAIDA data
  - Capture, Curation, Storage, Search, Sharing, Analysis, and Visualization
    - Ark topology: 1.8 TB
    - Telescope: 102 TB
    - Packet headers: 18.8 TB

Josh Polterock, "CAIDA: A Data Sharing Case Study,"
Security at the Cyber Border: Exploring Cybersecurity for International Research Network Connections workshop, 2012

Harness Distributed Computing and Storage ?

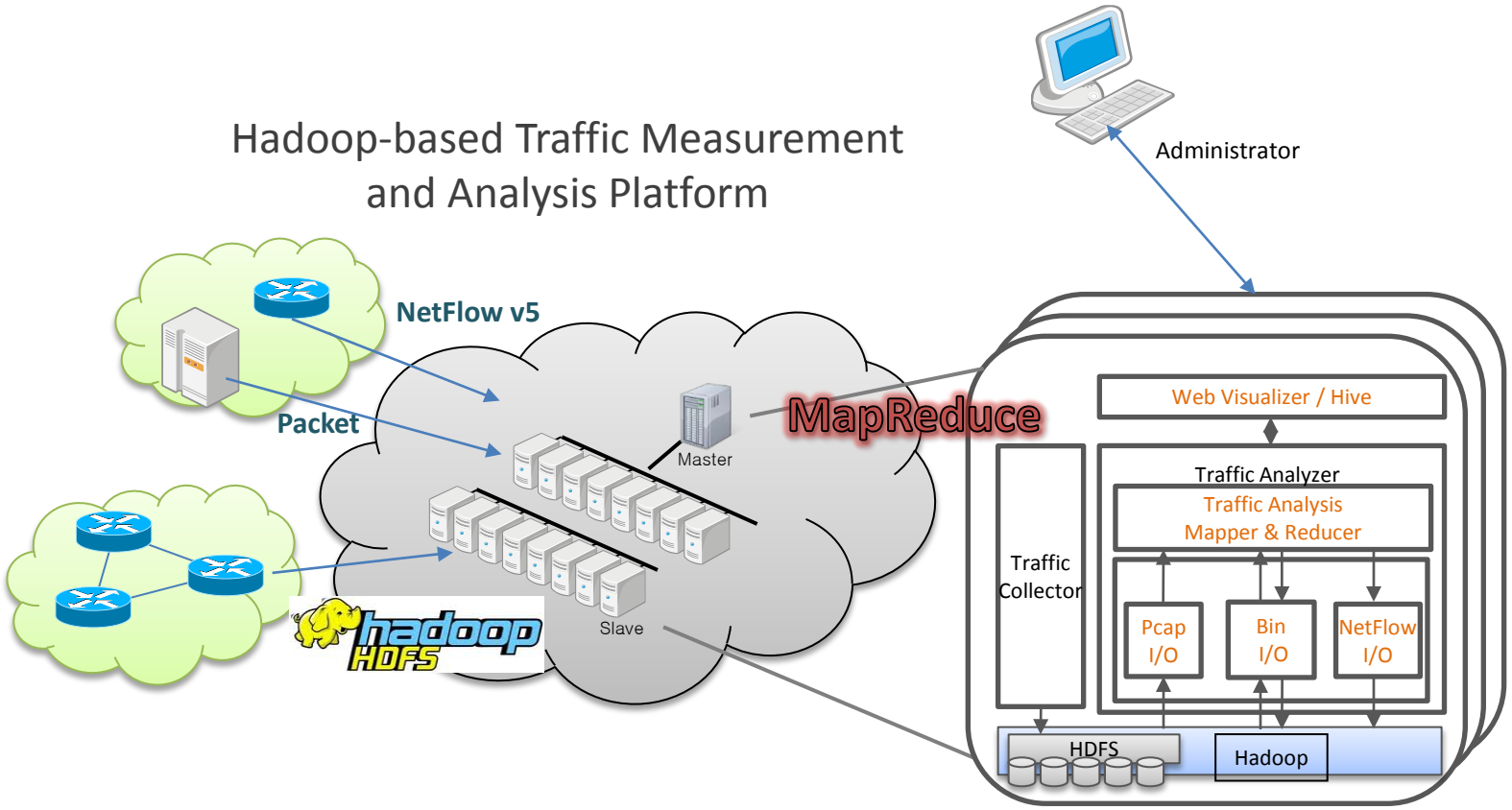
Google MapReduce, 2004

- 1 PB sorting by Google
  - 2008: 6 hours and 2 minutes on 4,000 computers
  - 2011: 33 minutes on 8000 computers
  - 2011: 10PB, 8000 computers, 6 hours and 27 minutes

Google

Apache Hadoop project

Our Proposal



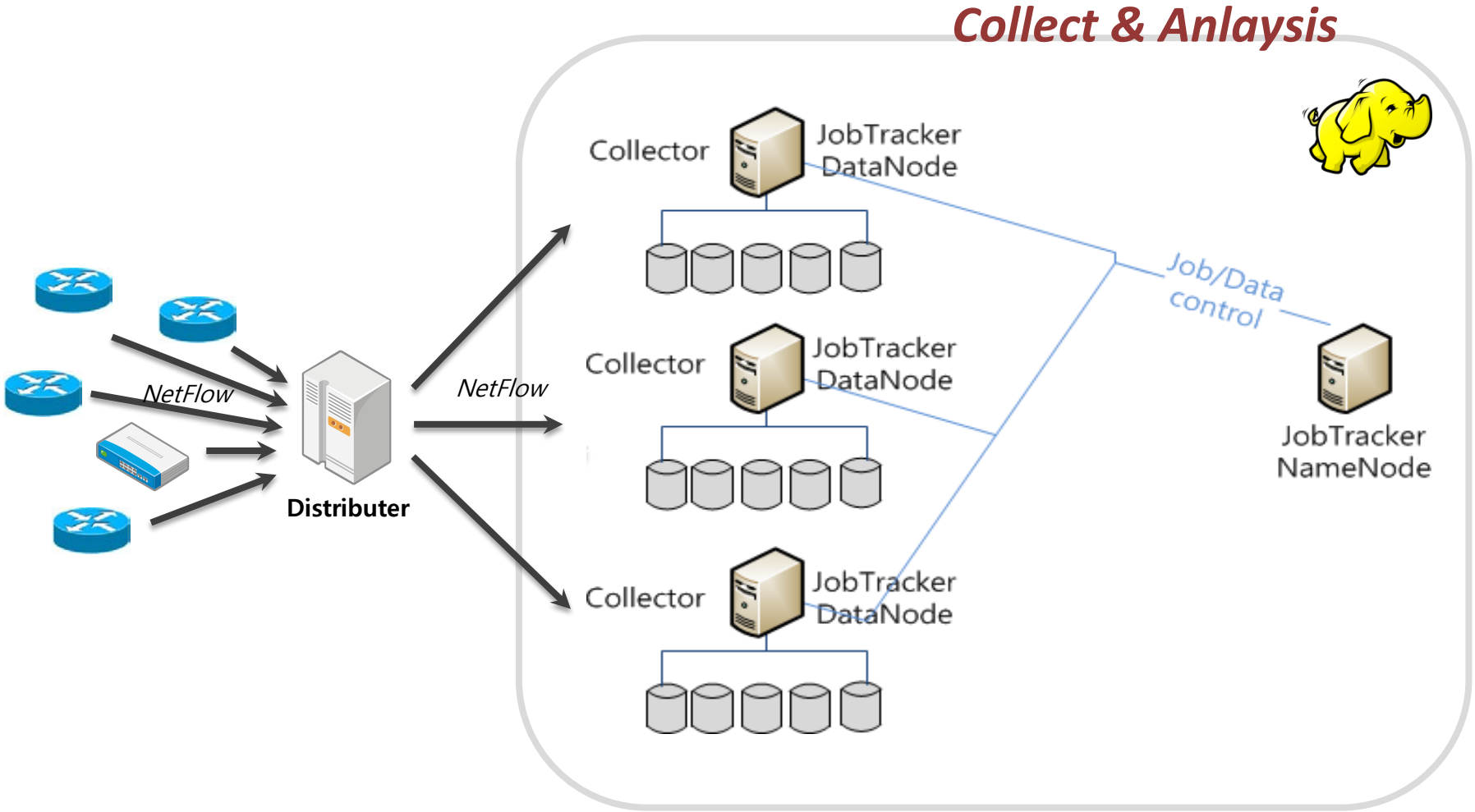
1. *Yeonhee Lee and Youngseok Lee, "Toward Scalable Internet Traffic Measurement and Analysis with Hadoop," ACM SIGCOMM Computer Communication Review (CCR), Jan. 2013*
2. *Yeonhee Lee and Youngseok Lee "A Hadoop-based Packet Trace Processing Tool" , TMA, April 2011*
3. *Yeonhee Lee and Youngseok Lee, "Detecting DDoS Attacks with Hadoop", ACM CoNEXT Student Workshop, Dec, 2011*

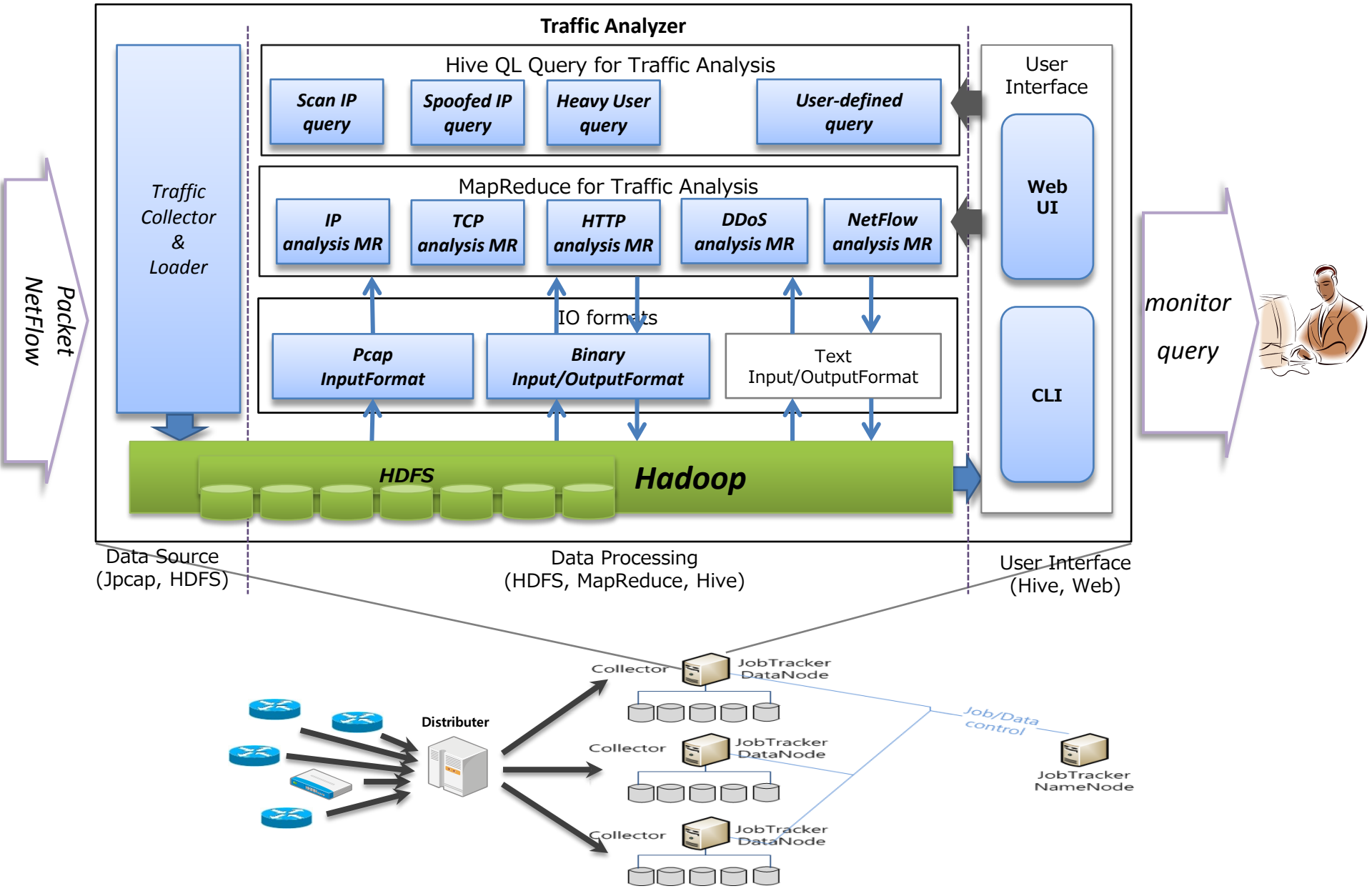
Related Work

- Traffic analysis of DNS root server (RIPE, 2011.11)
- PacketPig (2012.03) - Big Data Security Analytics platform
- Sherpasurfing – Open Source Cyber Security Solution, Hadoop World 2011
  - Firewall/IDS logs, netflow/packet
- Performing Network and Security Analytics with Hadoop, (Travis Dawson, Narus), Hadoop Summit 2012
- Distributed Bro (IDS)

OVERVIEW

Hadoop-based NetFlow Analysis




Traffic Analyzer
Hive QL Query for Traffic Analysis
Scan IP query
Spoofed IP query
Heavy User query
User-defined query
MapReduce for Traffic Analysis
IP analysis MR
TCP analysis MR
HTTP analysis MR
DDoS analysis MR
NetFlow analysis MR
IO formats
Pcap InputFormat
Binary Input/OutputFormat
Text Input/OutputFormat
HDFS
Hadoop
Traffic Collector & Loader
User Interface
Web UI
CLI
Packet NetFlow
monitor query
Data Source (Jpcap, HDFS)
Data Processing (HDFS, MapReduce, Hive)
User Interface (Hive, Web)
Distributer
Collector
JobTracker DataNode
Collector
JobTracker DataNode
Collector
JobTracker DataNode
Job/Data control
JobTracker NameNode

HADOOP-BASED TRAFFIC ANALYSIS

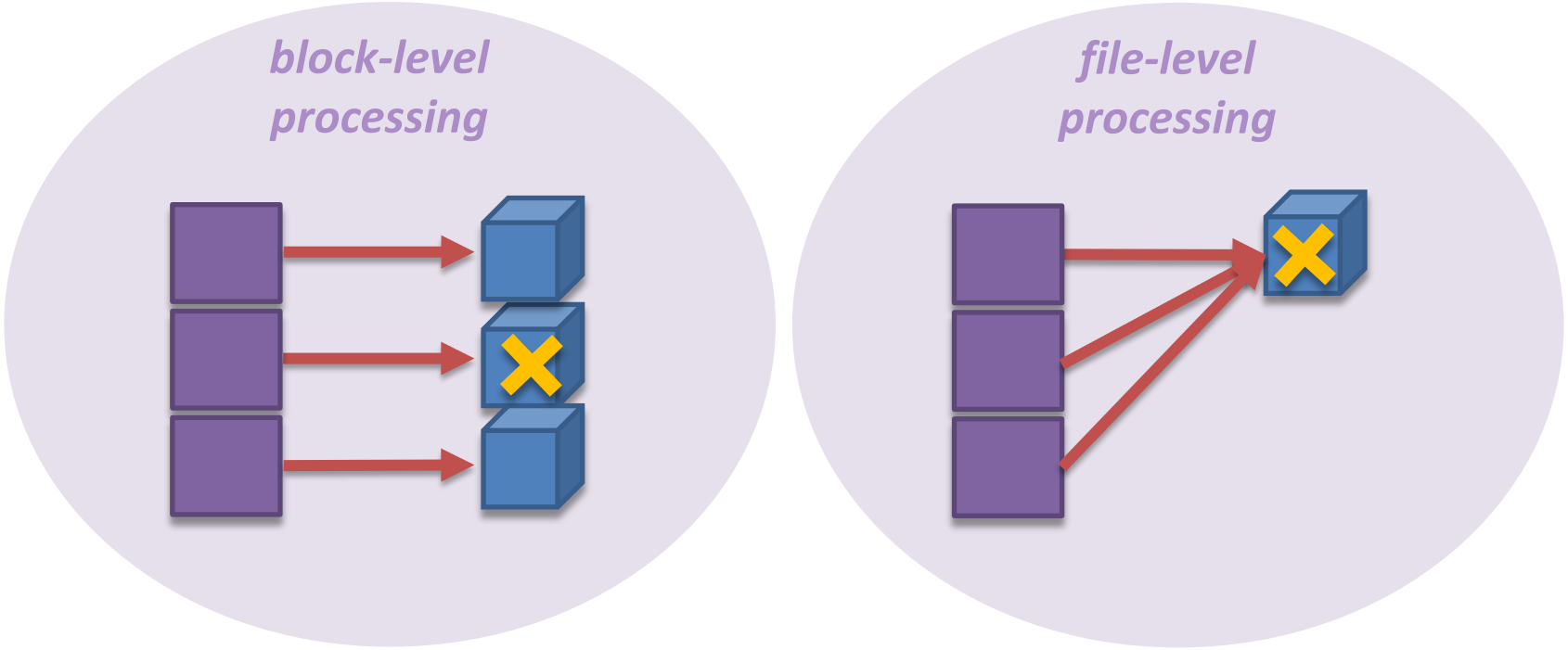
Challenges

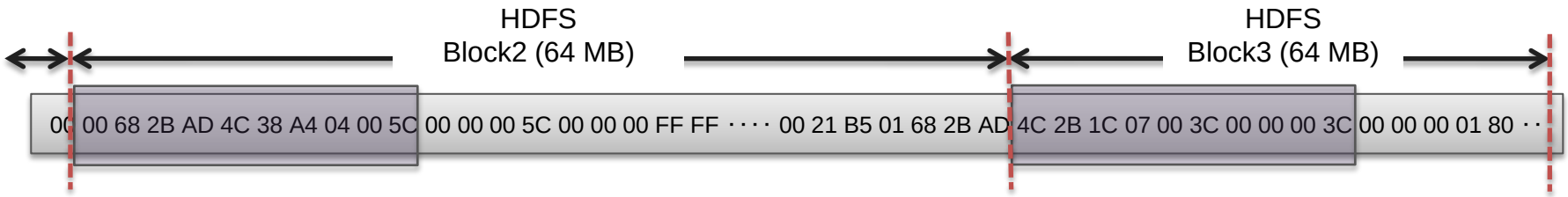
1. Data handing issue in HDFS
2. Distributed traffic analysis MapReduce algorithms
3. Performance tuning in a large-scale Hadoop

Challenges

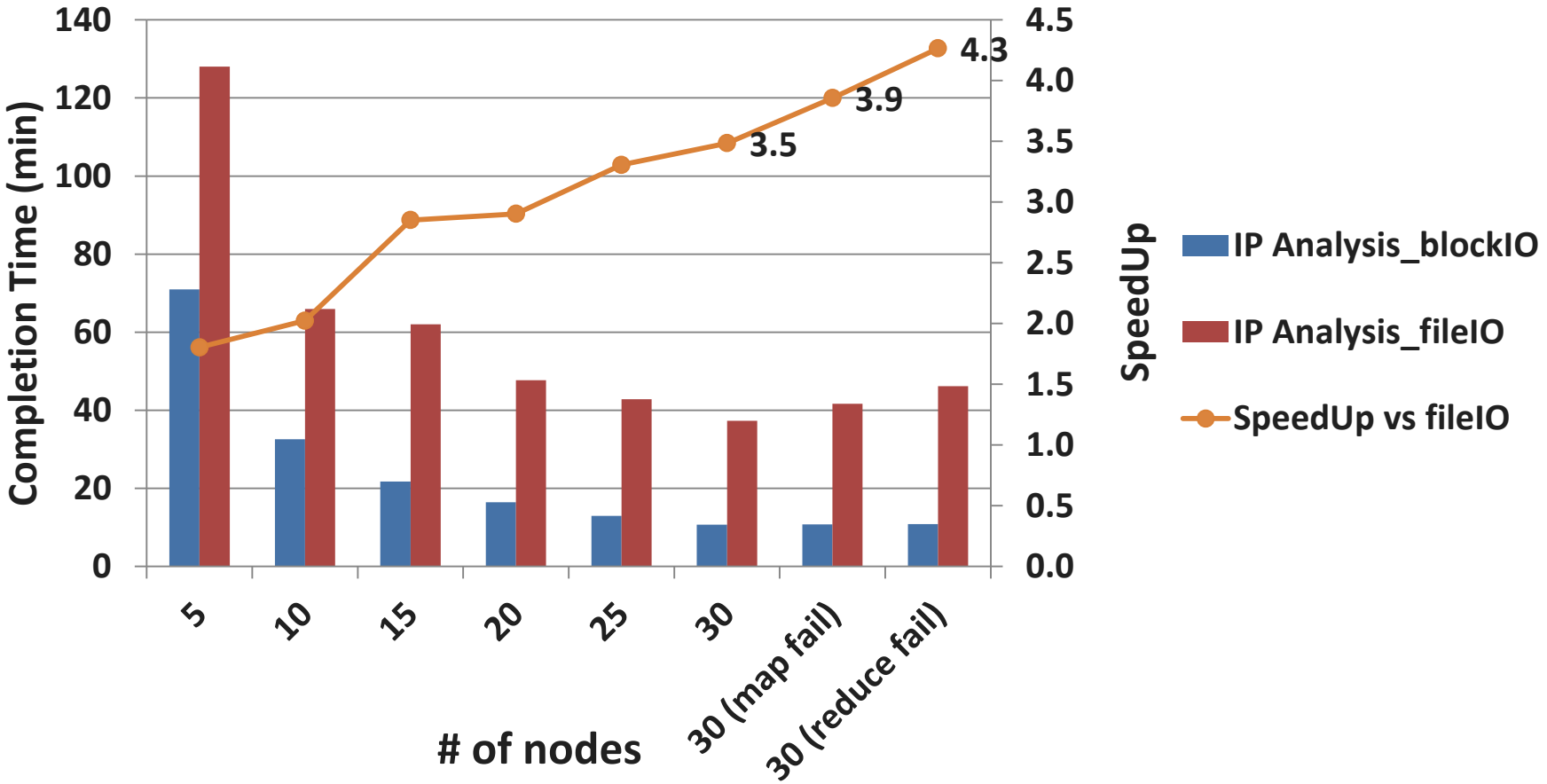
1. Data handing issue in Hadoop
2. Distributed traffic analysis MapReduce algorithms
3. Performance tuning in a large-scale Hadoop testbed

Block-level Parallelism





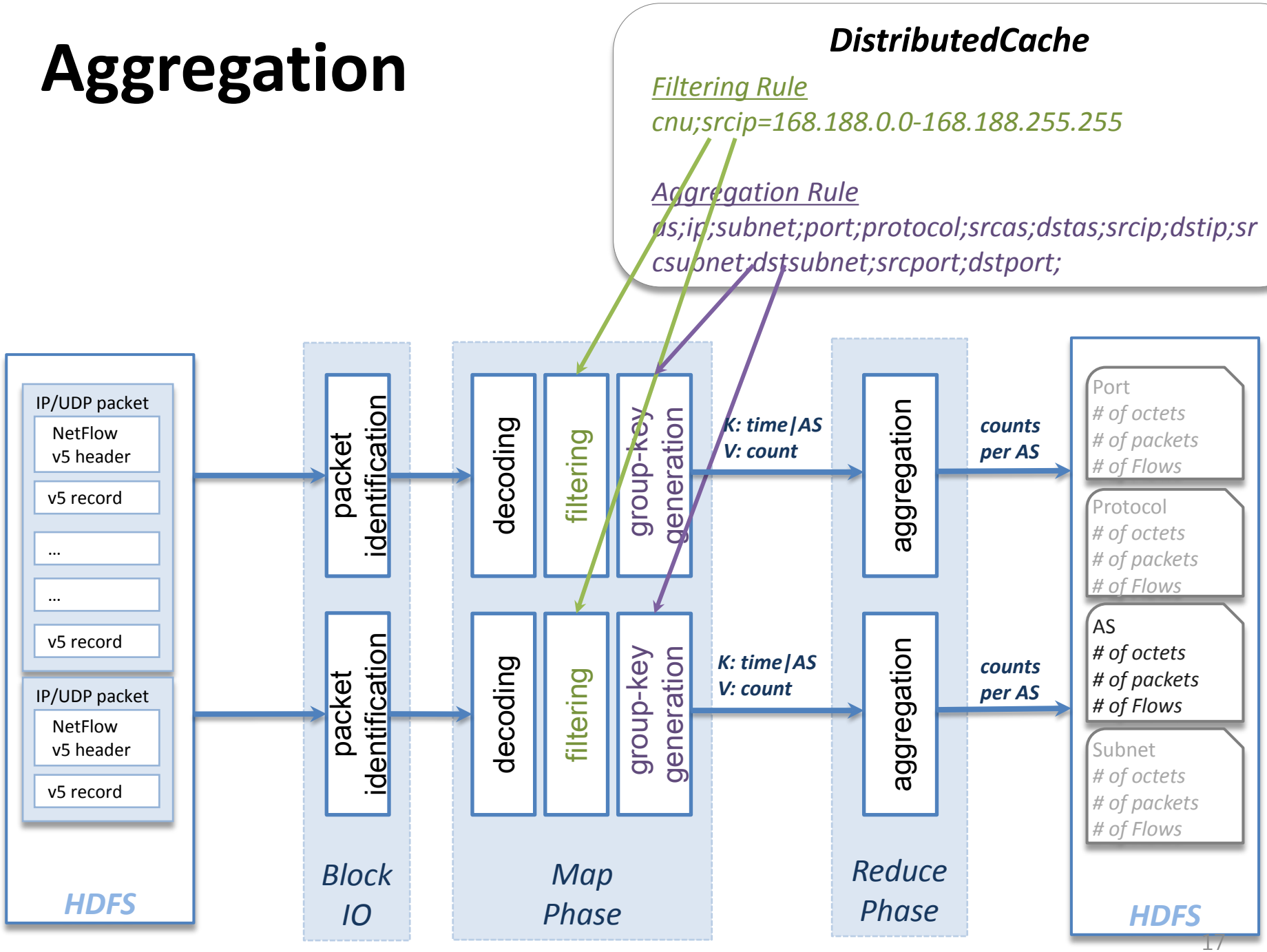
Block-level IO vs. File-level IO



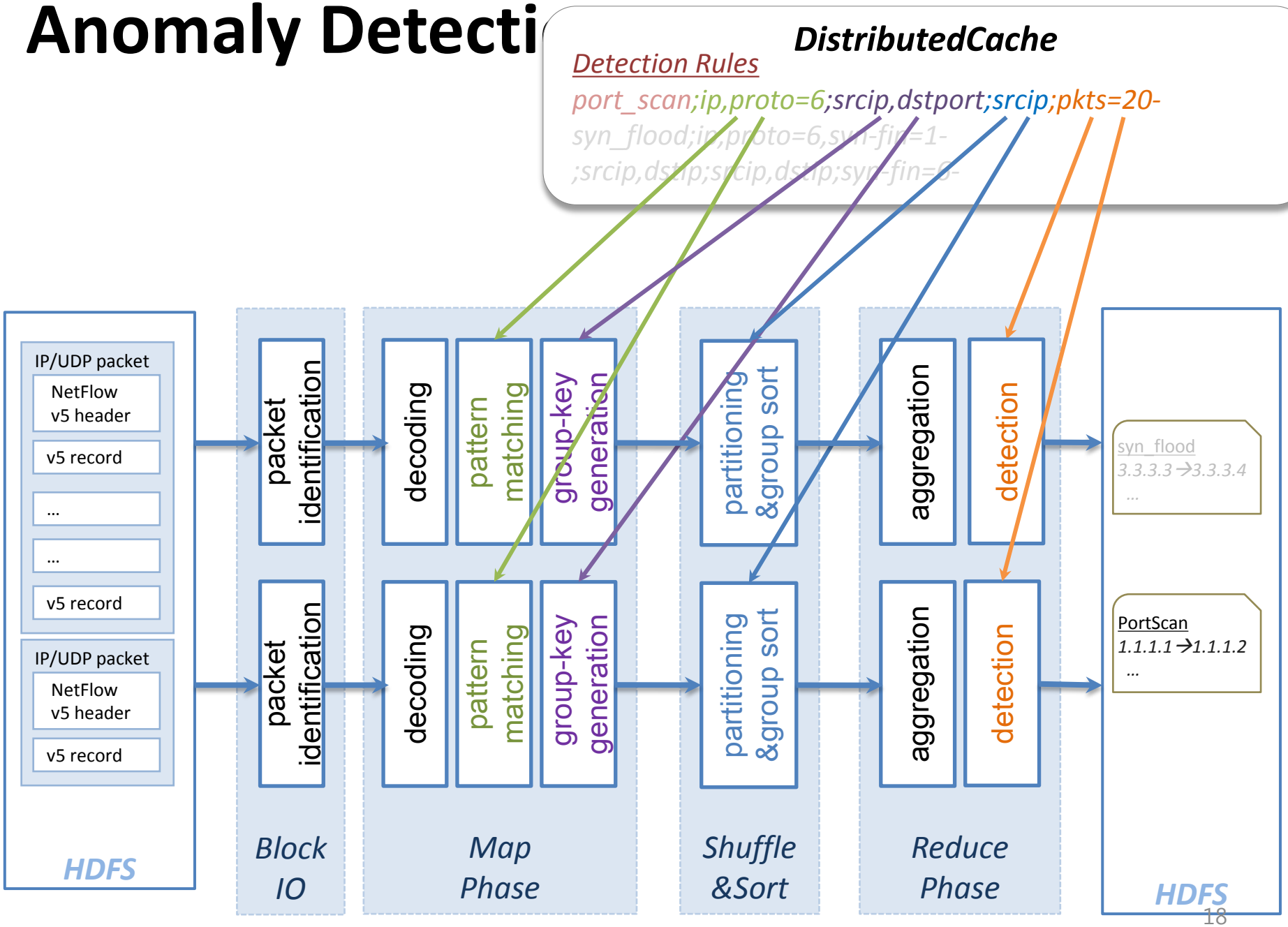
Challenges

1. Data handing issue in Hadoop
2. Distributed traffic analysis MapReduce algorithms
3. Performance tuning in a large-scale Hadoop testbed

Aggregation



Anomaly Detecti



Challenges

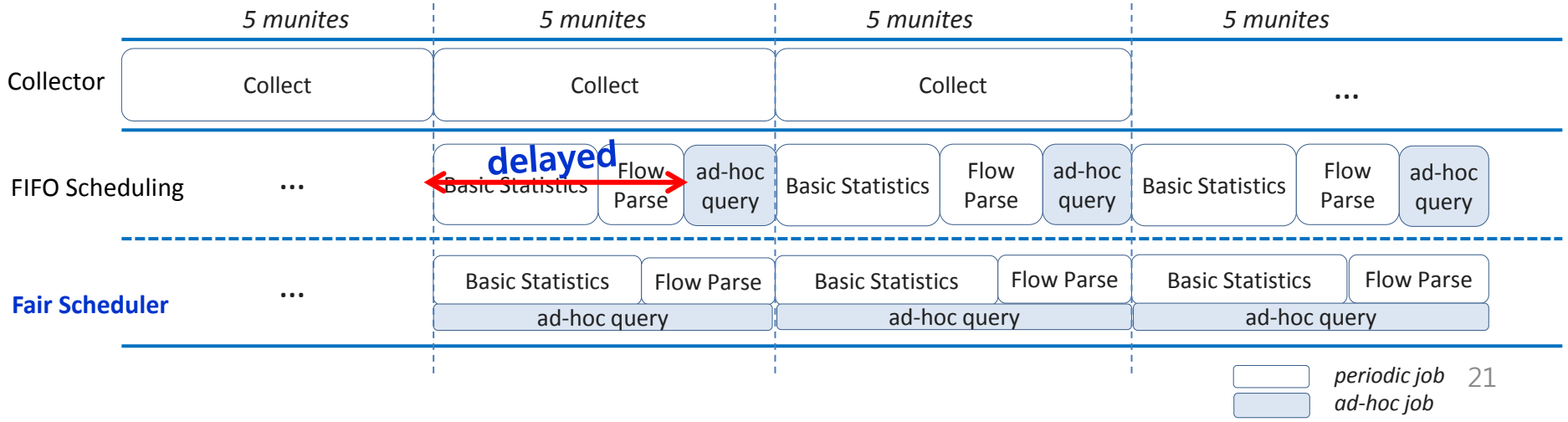
1. Data handing issue in Hadoop
2. Distributed traffic analysis MapReduce algorithms
3. Performance tuning in a large-scale Hadoop

Performance Tuning

- Configuration
  - Hadoop IO Buffer (128K → 1 MB)
  - Java heap space (300 MB →1 024 MB)
  - # of MapReduce Slots (→# of cores)
- MapReduce Algorithm
  - normal combiner vs inMapper combiner
- Job scheduling

Job Scheduling

- Different job types
  - Periodic jobs (for monitoring)
    - guaranteed service within time
    - e.g Aggregated Statistics for monitoring, Flow Parse job for analytics
  - Small ad-hoc query job (for analytics)
    - fast response time



PERFORMANCE EVALUATION

Experiments

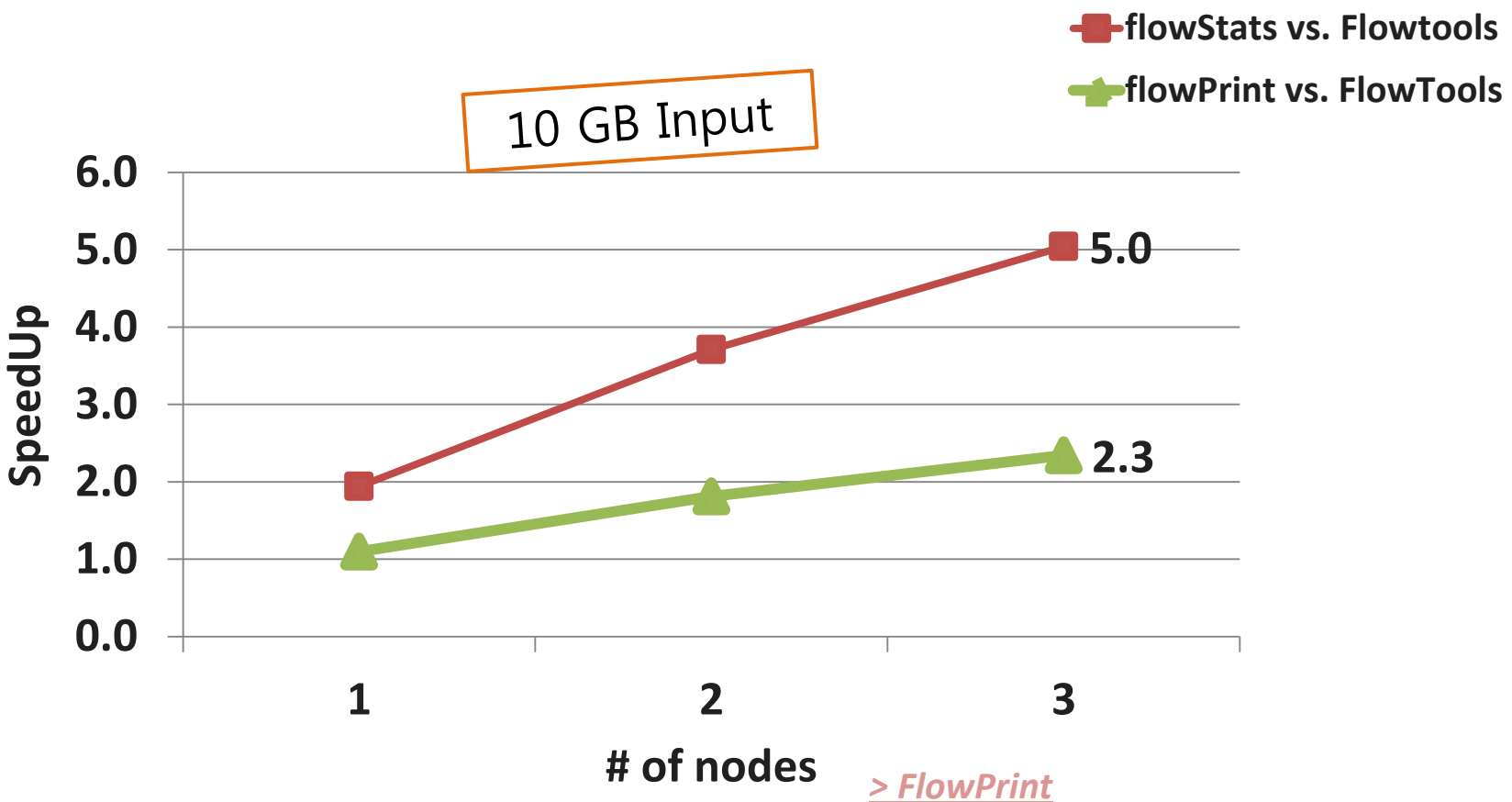
- Testbed

Type	Nodes	Cores	CPU	Memory	HardDisk	Rack
Small	3	24	3.4 GHz 8 core	16 GB	2 TB	1 Rack
Medium	30	240	2.93 GHz 8 core	16 GB	4 TB	1 Rack
Large	200	400	2.66 GHz 2 core	2 GB	500 GB	4 Racks

- Data and MapReduce jobs

Type	Dataset	MapReduce Job	Testbed
NetFlow	1 TB from KOREN	flowStats, flowDetect, flowPrint	Small
Packet	1 ~ 5 TB from CNU campus N/W	IP, TCP, Web (webpop, User Behavior, DDoS)	Medium, Large

NetFlow: SpeedUp (vs. Flowtools)



> FlowPrint

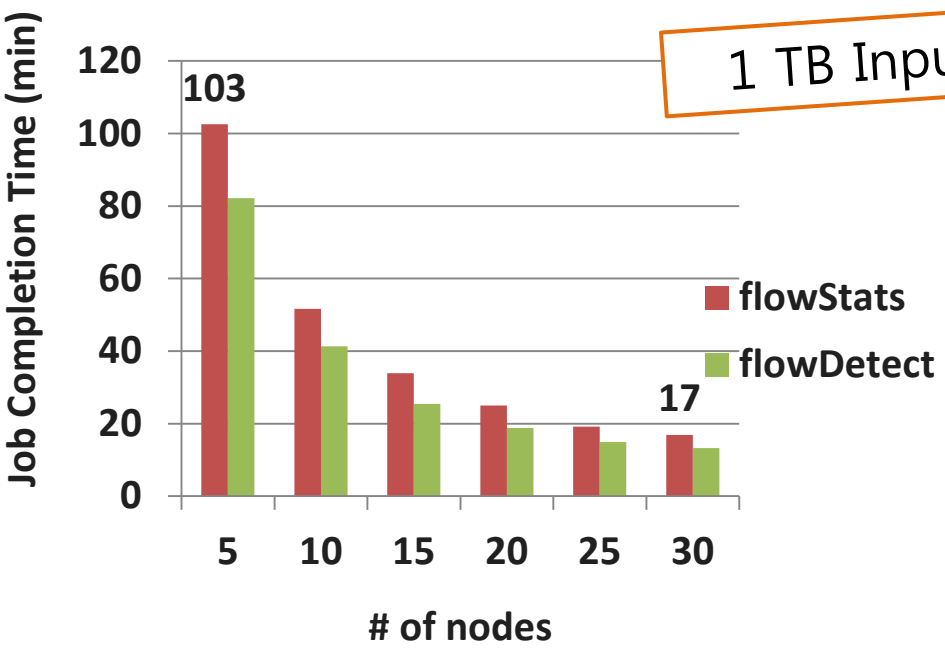
flow-cat -p flowfile |flow-print –f14

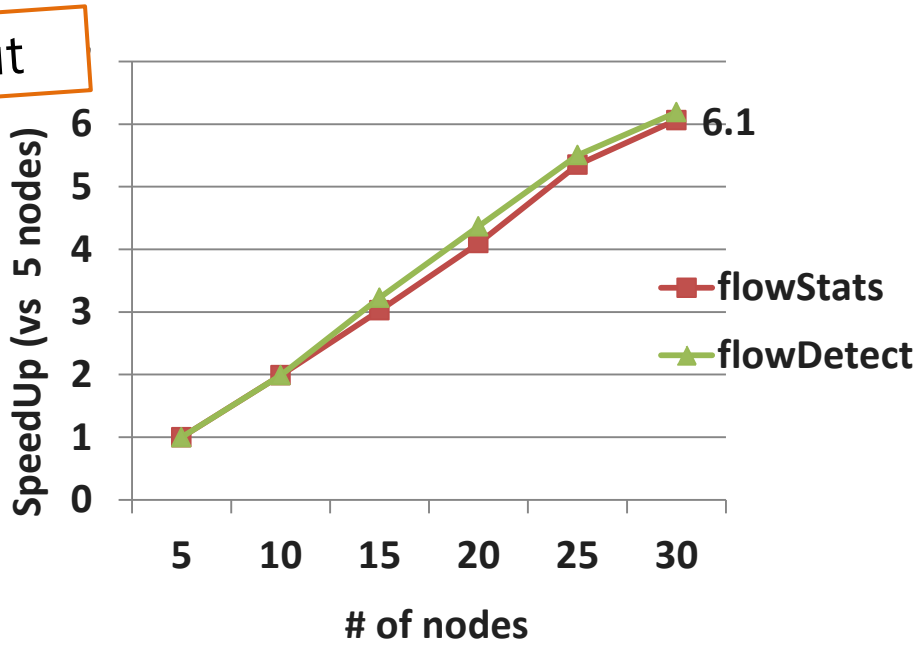
> FlowStats

flow-cat -p flowfile|flow-stat -f12

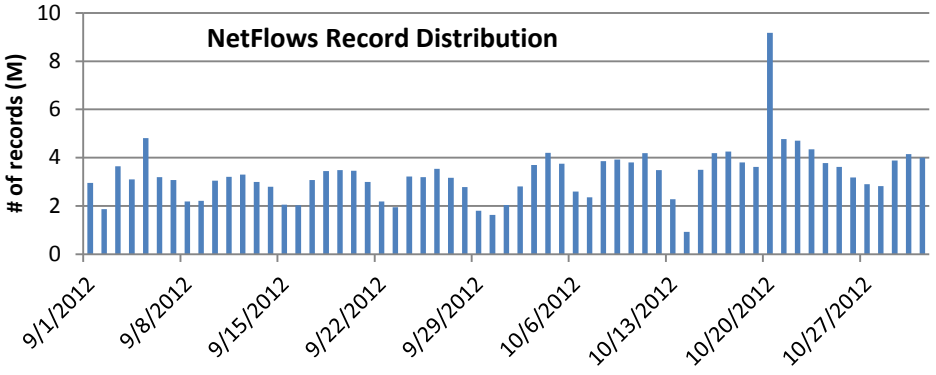
flow-cat -p flowfile|flow-stat –f5

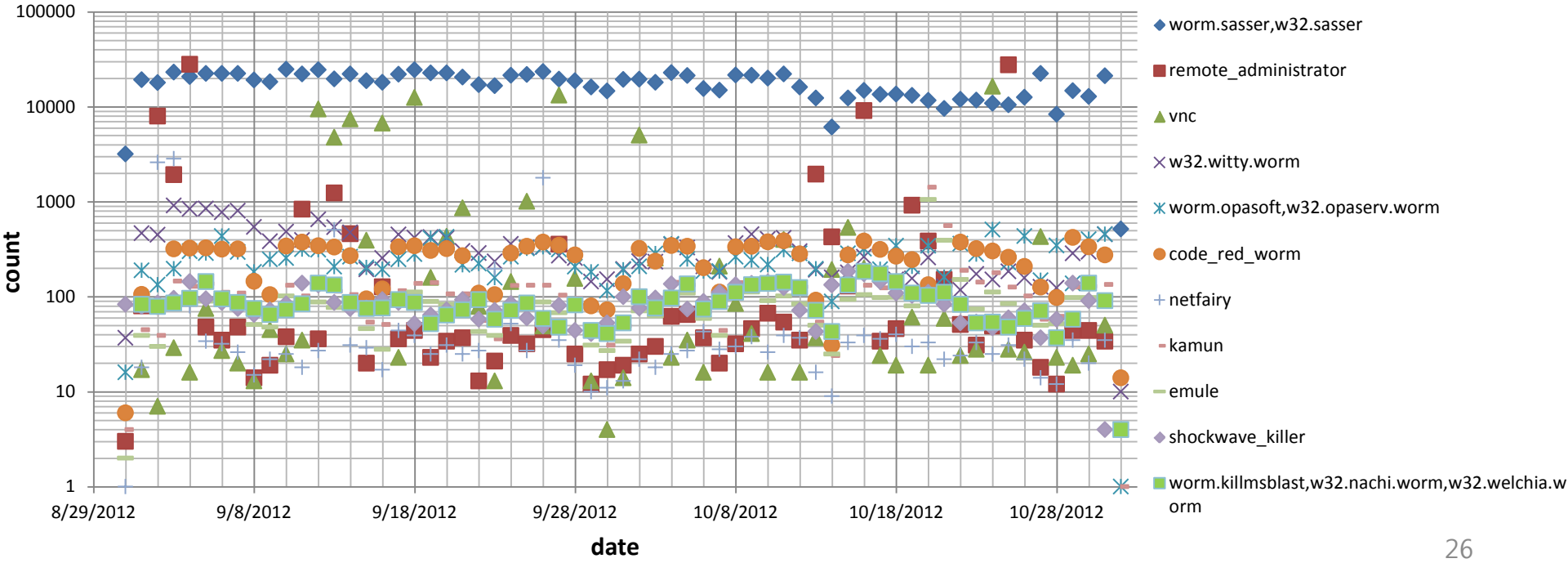
NetFlow: Scalability



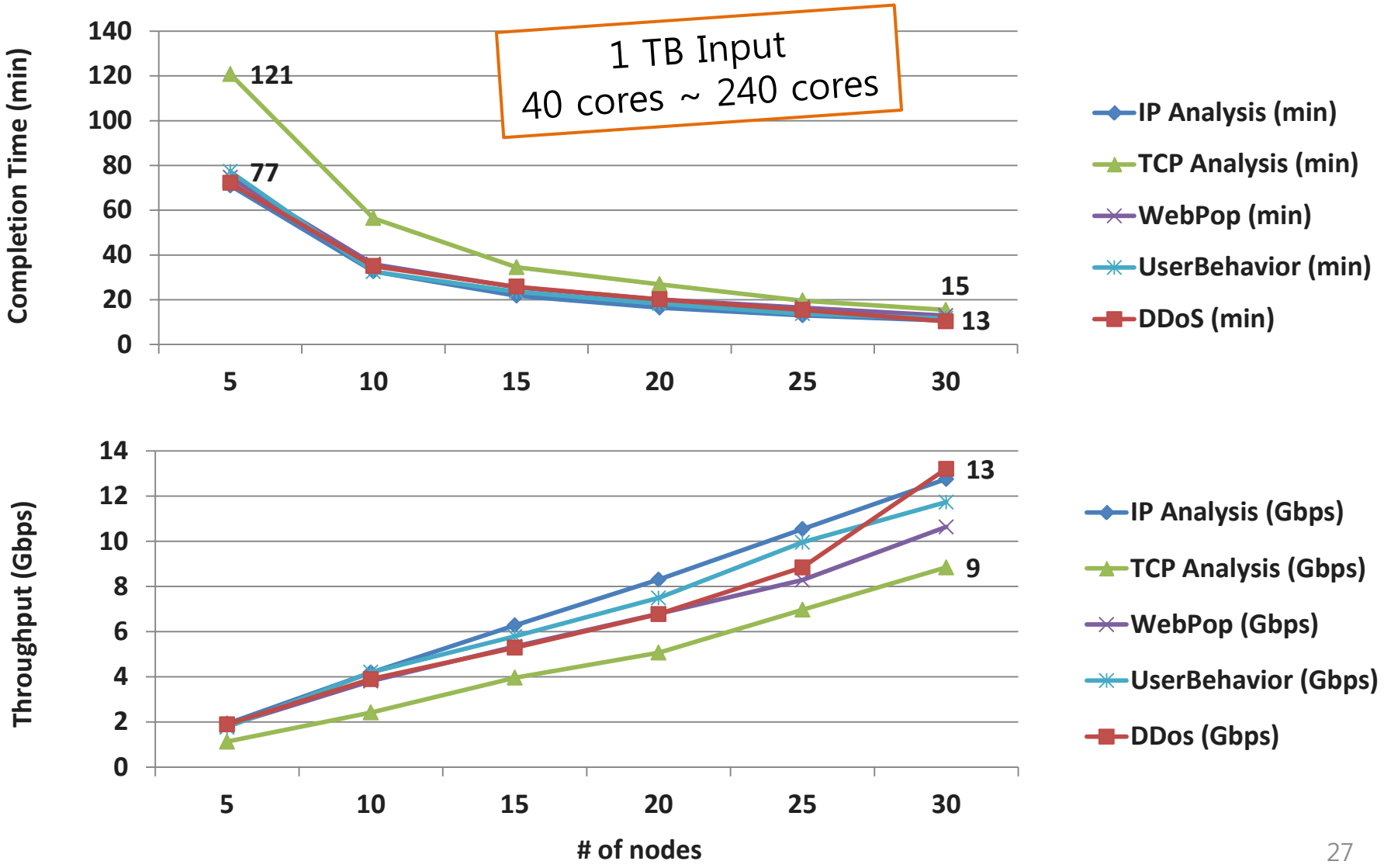


NetFlow: Pattern Matching Result

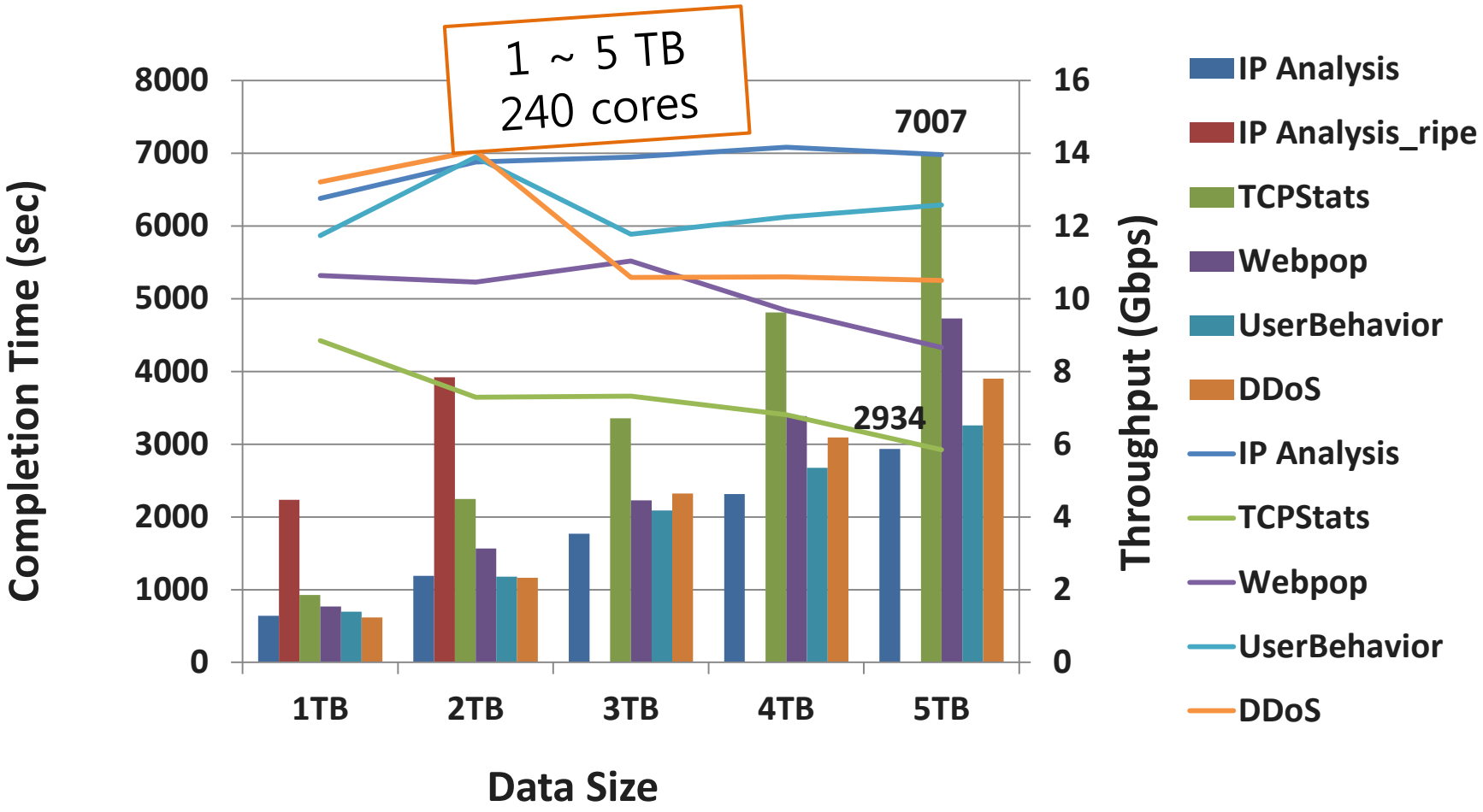




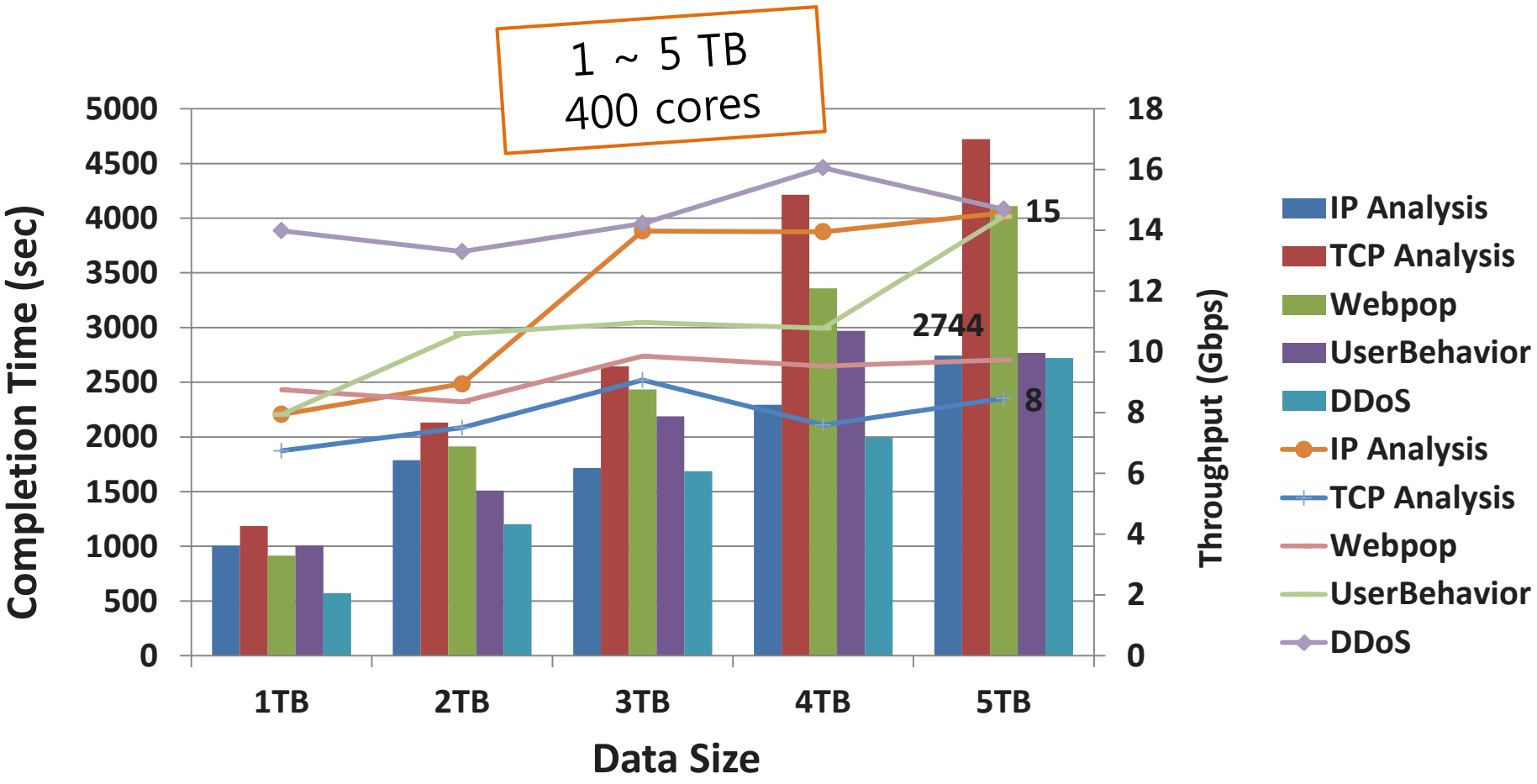
Packet: ScaleOut



Packet: SizeUp (30 nodes)



Packet: SizeUp (200 nodes)



SUMMARY

Summary

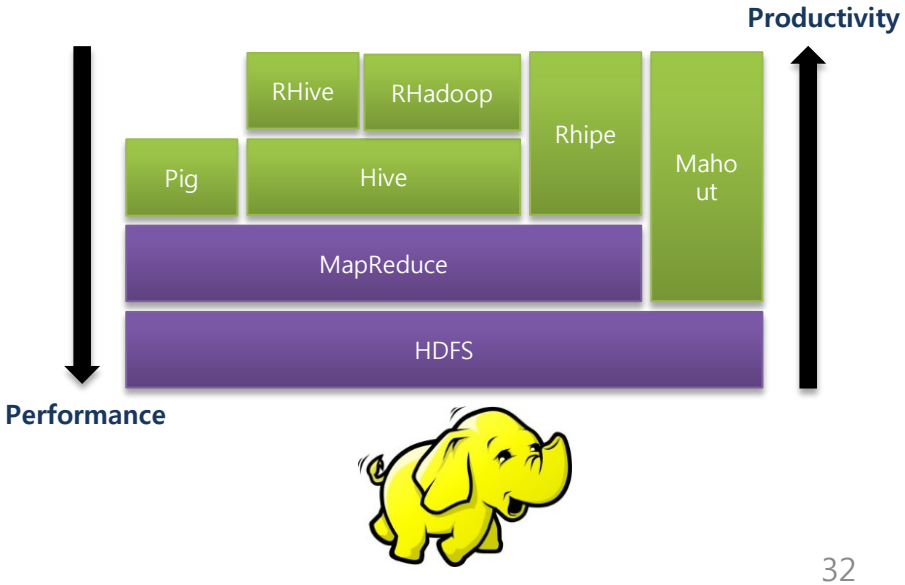
- NetFlow analysis with Hadoop
  - NetFlow v5 processing module
  - MapReduce algorithms: statistics

- Distributed computing and storage with Hadoop
  - Fits Internet measurement application
  - Scalability

- Source codes are available at
  - Packet, NetFlow
  - https://sites.google.com/a/networks.cnu.ac.kr/dnlab/research/hadoop
  - https://github.com/ssallys/pcap-on-Hadoop

Ongoing Work

- Distributed real-time monitoring
  - Rule matching for Streamed NetFlow
  - Developing rule for MapReduce
  - Rule classification for dedicated rule matching
- Integration
  - Streaming packages
  - Enhanced analytics
    - Data mining: Mahout
    - Machine learning
- Scalable collection
  - E.g.) 10GE → 10 X 1 GE HDFS



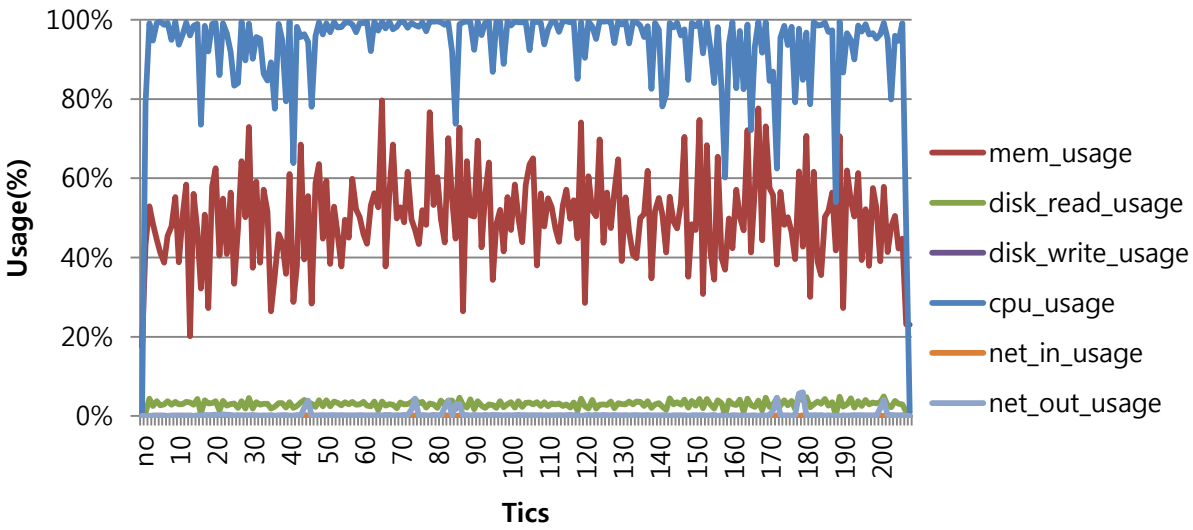
Reference

- Papers
  1. *Y. Lee and Y. Lee, "Toward Scalable Internet Traffic Measurement and Analysis with Hadoop," ACM SIGCOMM Computer Communication Review (CCR), Jan. 2013*
  2. *Y. Lee, W. Kang, and Y. Lee, "A Hadoop-based Packet Trace Processing Tool," The Third TMA, April 2011*
  3. *Y. Lee and Y. Lee, "Detecting DDoS Attacks with Hadoop", ACM CoNEXT Student Workshop, Dec, 2011*

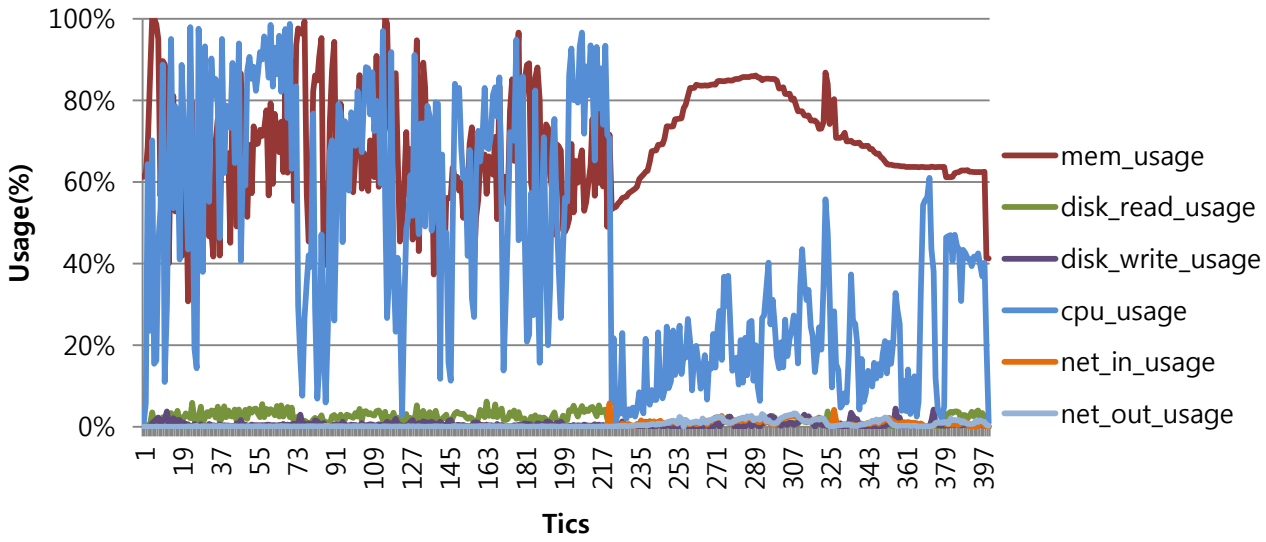
- Software
  1. *http://networks.cnu.ac.kr/~yhlee*
  2. *https://sites.google.com/a/networks.cnu.ac.kr/dnlab/research/hadoop*
  3. *https://github.com/ssallys/pcap-on-Hadoop*

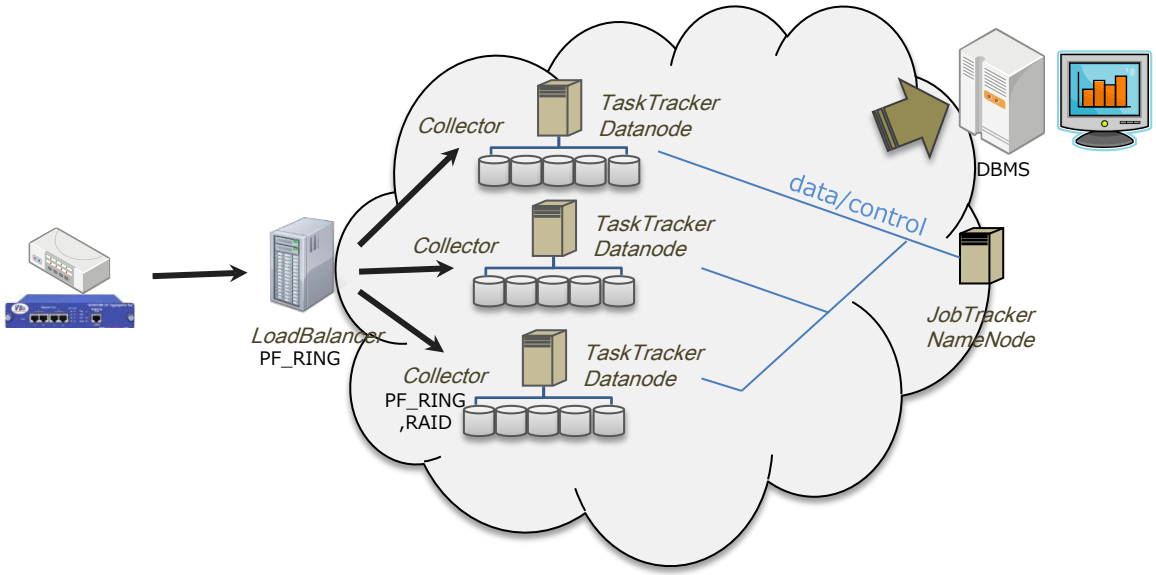
THANK YOU !

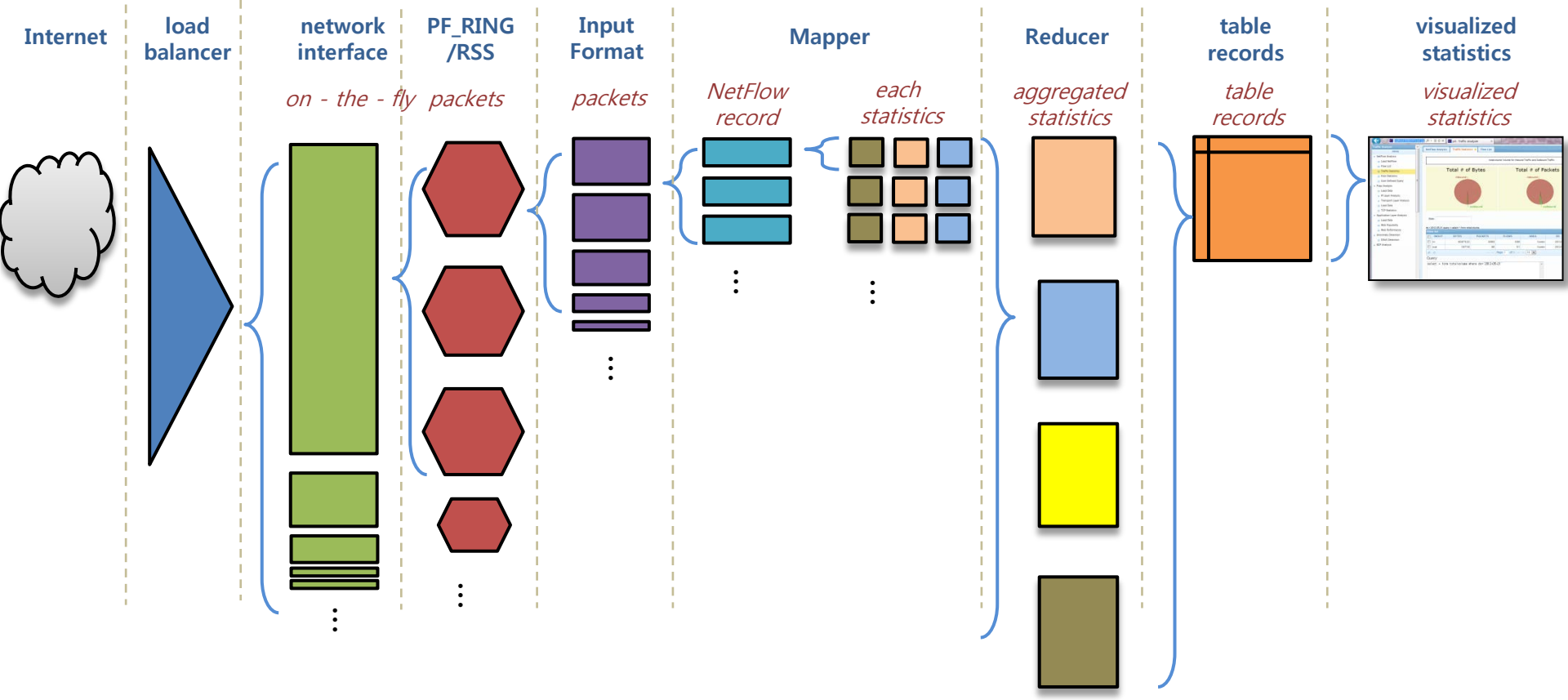
IP Analysis



TCP Analysis




Collector
TaskTracker
Datanode
Collector
TaskTracker
Datanode
Collector
TaskTracker
Datanode
PF_RING
,RAID
LoadBalancer
PF_RING
data/control
DBMS
JobTracker
NameNode



rule name; filter pattern; mapout key; patition&groupsort key;detection condition; action

ex)
port_scan;ip,proto=6;srcip,dstport;srcip;pkts=20-
syn_flood;ip,proto=6,syn-fin=1-;srcip,dstip;srcip,dstip;syn-fin=6-

A Distributed Network Security Analysis System

Based on Apache Hadoop-Related Technologies

Bingdong Li,

Jeff Springer , Mehmet Gunes , George Bebis

University of Nevada Reno

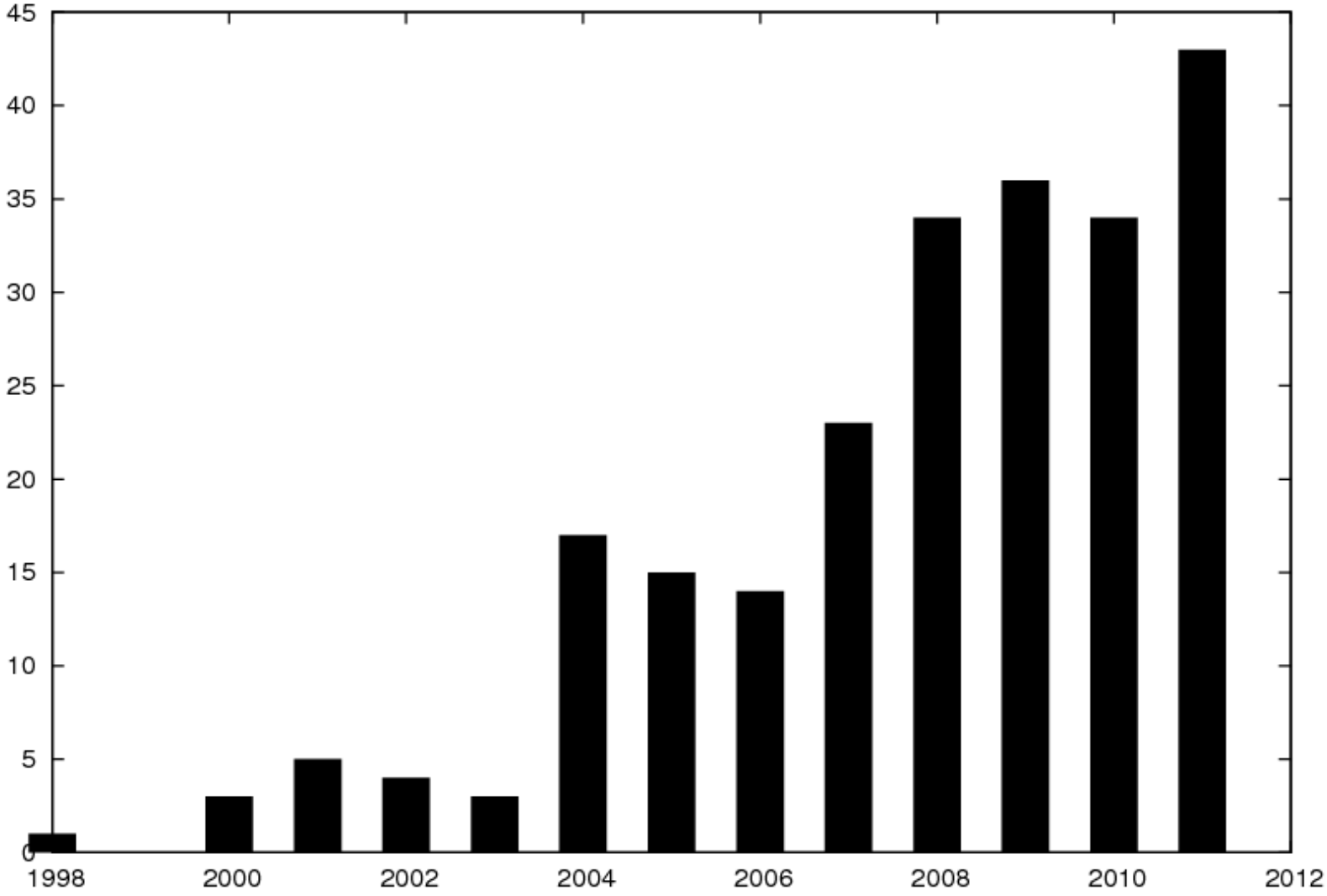
FloCon 2013

January 7-10, Albuquerque, New Mexico

Agenda

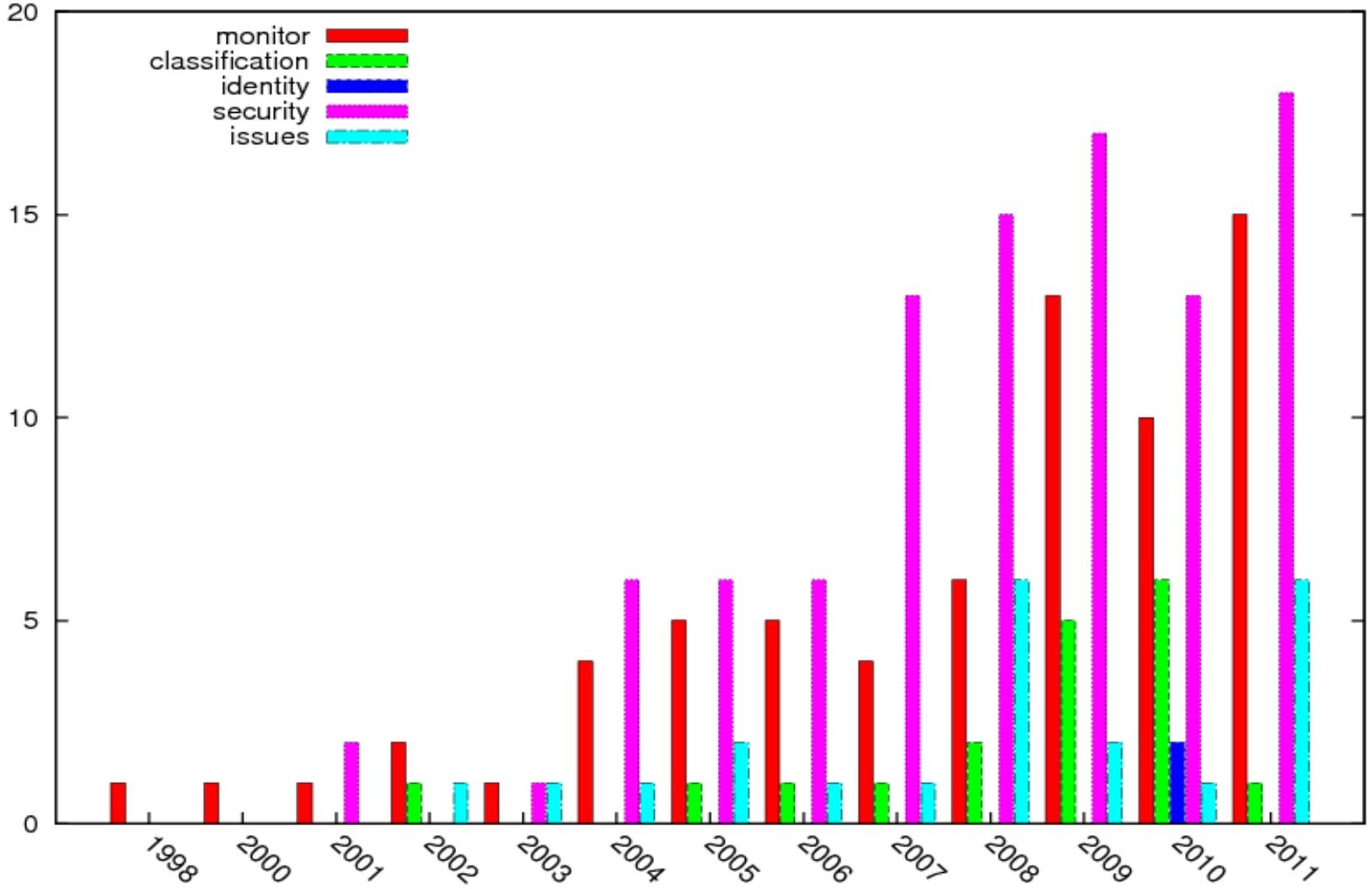
- Review
- Challenges
- Apache Hadoop Related Technologies
- System Design
- Demonstration
- Thoughts and Pitfalls
- Summary

Publications By Years



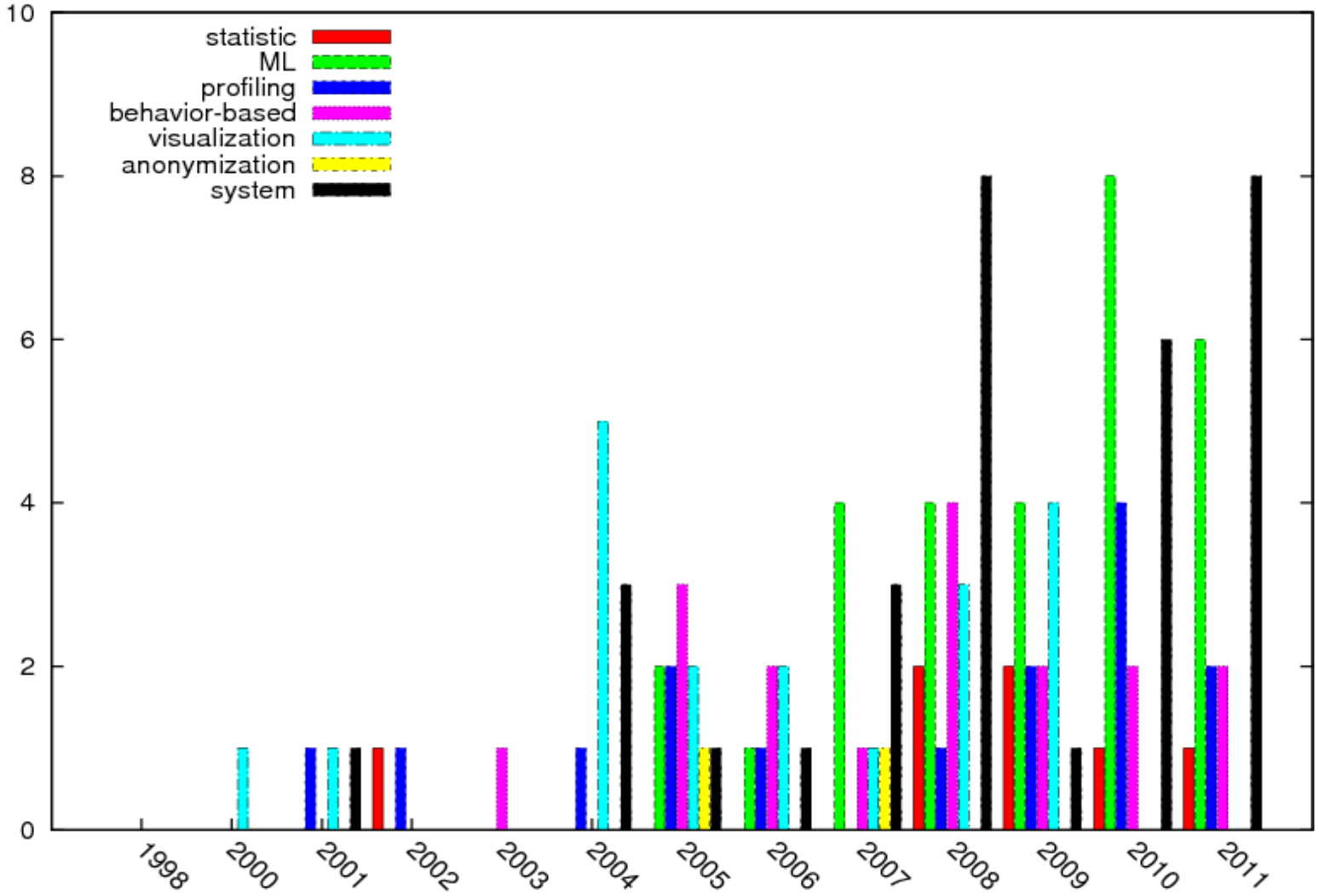
Bingdong Li, Jeff Spinger, George Bebis, Mehmet Hadi Gunes, A Survey of Network Flow Applications, Journal of Networks and Computer Applications (accepted).

Research Perspectives By Years



Bingdong Li, Jeff Spinger, George Bebis, Mehmet Hadi Gunes, A Survey of Network Flow Applications, Journal of Networks and Computer Applications (accepted).

Methods By Years



Bingdong Li, Jeff Spinger, George Bebis, Mehmet Hadi Gunes, A Survey of Network Flow Applications, Journal of Networks and Computer Applications (accepted).

Challenges

- Too much data (volume)
- Real Time and On Demand (velocity)
- Various types/sources of data (variety)
- Changing requirements(variability)

Big Data – Volume, Velocity, Variety (Gartner's Doug Laney) ,

Variability (Forrester's James Kobielus G. etc.)

http://blogs.sas.com/content/datamanagement/2011/11/05/big-data-defined-its-more-than-hadoop/

Apache Hadoop Related Technologies

- **What is Apache Hadoop?**

  Open source, storing and processing Big Data

- **Main Systems:**
  - Hadoop Distributed File System (HDFS)
  - MapReduce

Apache Hadoop Related Technologies

- **Data collection:**

  Flume, Chukwa, …

- **Storage:**

  HDFS, Cassandra, CouchDB, …

- **Processing:**

  MapReduce, Pig, Hive, Mahout …

- …

Design

- **Goals**
- **Philosophy**
- **Components**
  - Data Collecting
  - Data Storage
  - Data Schema
  - Data Process
  - User Interfaces

Design Goals

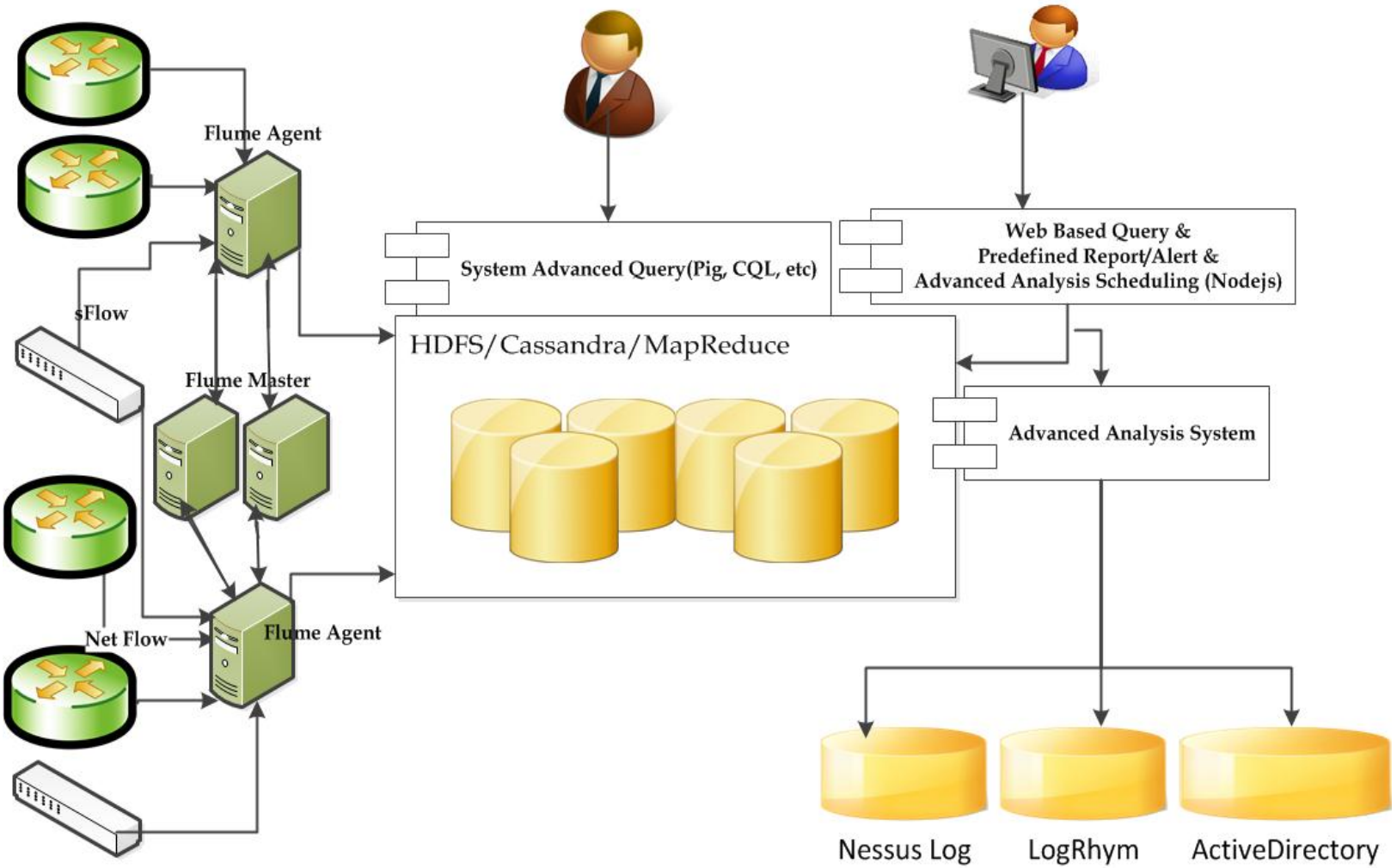
- Real time network query, near real time measurement and analysis
- Distributed system for data collecting, storing, accessing, measuring and analyzing NetFlow and other log data
- Models of detection and classification based on profiling and behavior

Design Philosophy

- Leverage existing technologies

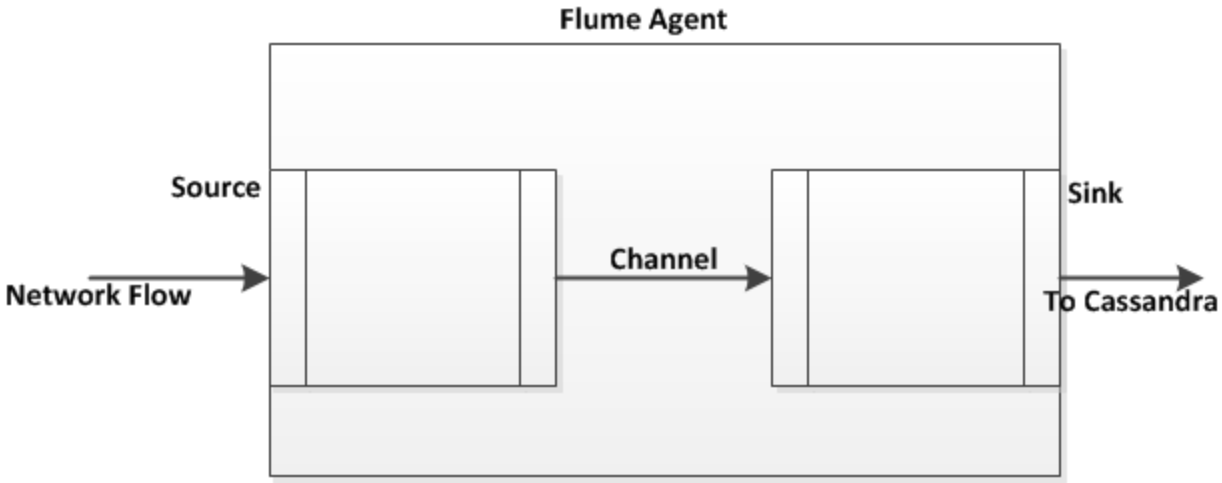
- Modeling known objects rather than unknown objects
  - or use white list rather than black list

Design: Components



Design: Components

- **Flume**: open source collecting, aggregating, and moving data from many different sources to data store
  - **Masters**: keep track all the nodes and inform them
  - **Agents**: Sources accept data, Sinks aggregate and send data, Decorator filter, sample and modify data flow.



Design: Components

C A P Conjecture

A web service can only satisfy any two of

- **C**onsistency
- **A**vailability
- **P**artition Tolerance

Cassandra is AP, arguably CAP with specifying consistency level

Any, one, quorum, local_quorum, each_quorum, ALL

Gilbert, Seth and Lynch, Nancy, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, SIGAACT News, 2002

Design: Components

- Cassandra Data Scehma
  - Keyspace
  - Column family
  - Rows and Columns

Design: Components

- Cassandra Index
  - Primary Index (row key)
  - Secondary Index (column values)
  - DIY with wide row or inverted index
  - Composite Column
  - Third party indexing
    - such as ElasticSearch, Solandra, DataStax Enterprise
- Counter

Design: Components

- Data Processing
  - Query network by CQL, or Web UI (Nodejs)
  - Network measurement by Pig scripting, R
  - Advanced data mining and network modeling by programming written by C++ and Java
  - Scheduling tasks

Design: Components

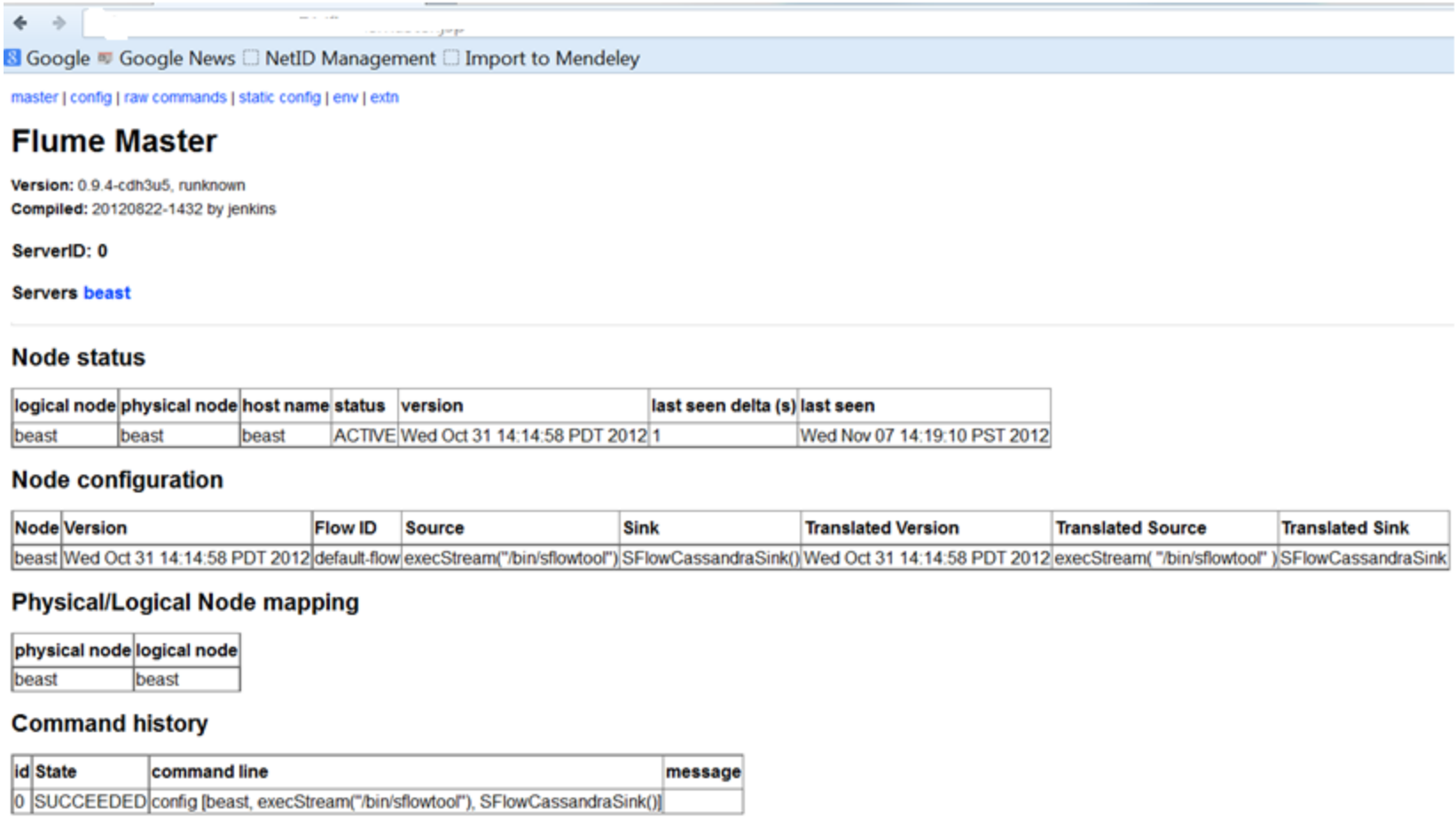
- User Interface
  - **Web User**:
    - through a secure internal web page to
      - see reports,
      - schedule advanced analysis tasks
  - **Advanced System User**:
    - use cassandra-cli, CQL, Pig, and R to do advanced measurement and analysis

Design: Features

- Query Network Status
- Network Measurement
- Advanced Network Modeling
  - Host Role's Behavior
  - Roles of Subnet Behavior
  - User Behaviors of Hosts

Demonstration

Flume



Google Google News NetID Management Import to Mendeley

master | config | raw commands | static config | env | extn

Flume Master

Version: 0.9.4-cdh3u5, runknown
Compiled: 20120822-1432 by jenkins

ServerID: 0

Servers beast

Node status

logical node	physical node	host name	status	version	last seen delta (s)	last seen
beast	beast	beast	ACTIVE	Wed Oct 31 14:14:58 PDT 2012	1	Wed Nov 07 14:19:10 PST 2012

Node configuration

Node	Version	Flow ID	Source	Sink	Translated Version	Translated Source	Translated Sink
beast	Wed Oct 31 14:14:58 PDT 2012	default-flow	execStream("/bin/sflowtool")	SFlowCassandraSink()	Wed Oct 31 14:14:58 PDT 2012	execStream("/bin/sflowtool")	SFlowCassandraSink

Physical/Logical Node mapping

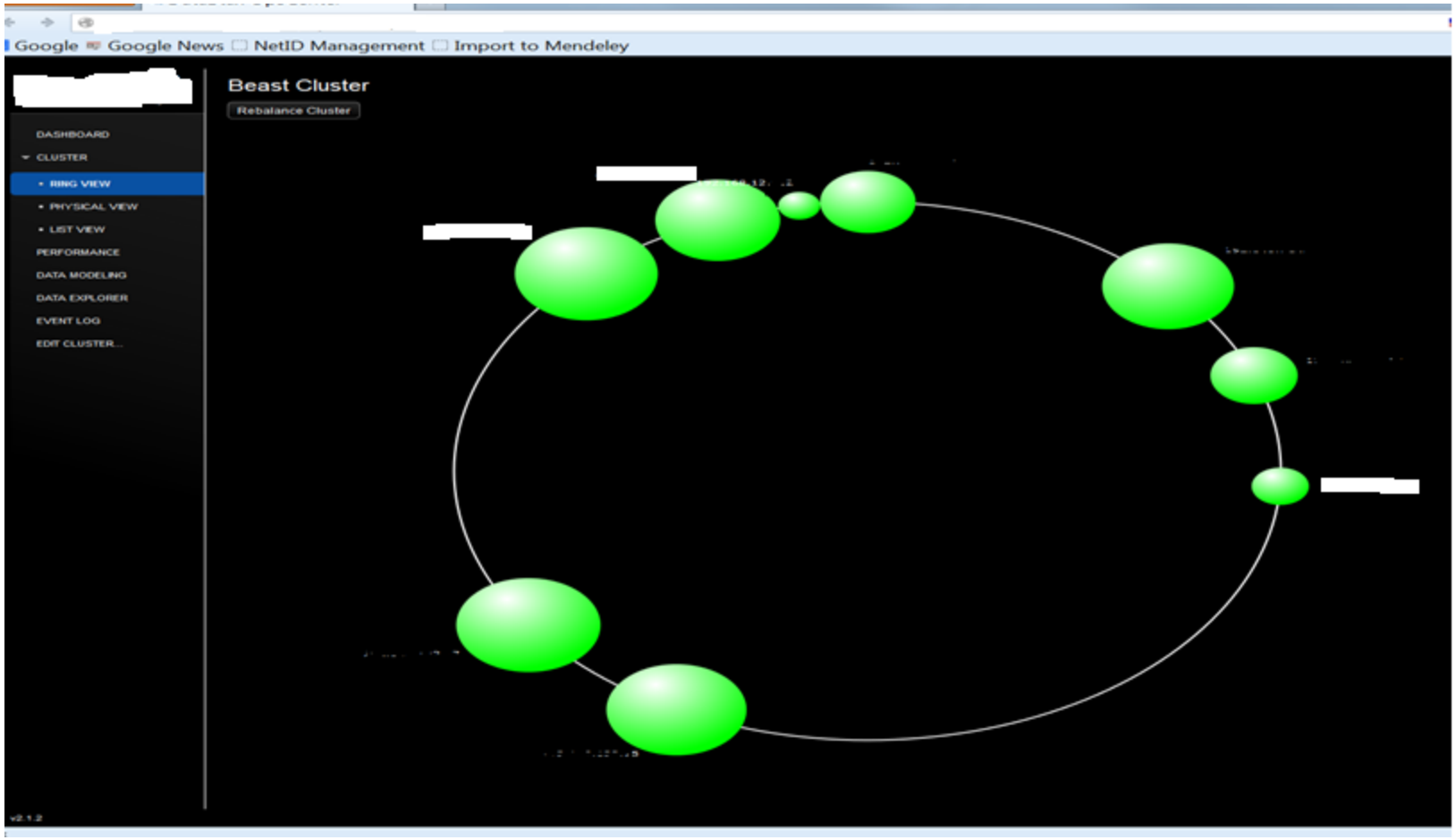
physical node	logical node
beast	beast

Command history

id	State	command line	message
0	SUCCEEDED	config [beast, execStream("/bin/sflowtool"), SFlowCassandraSink()]	

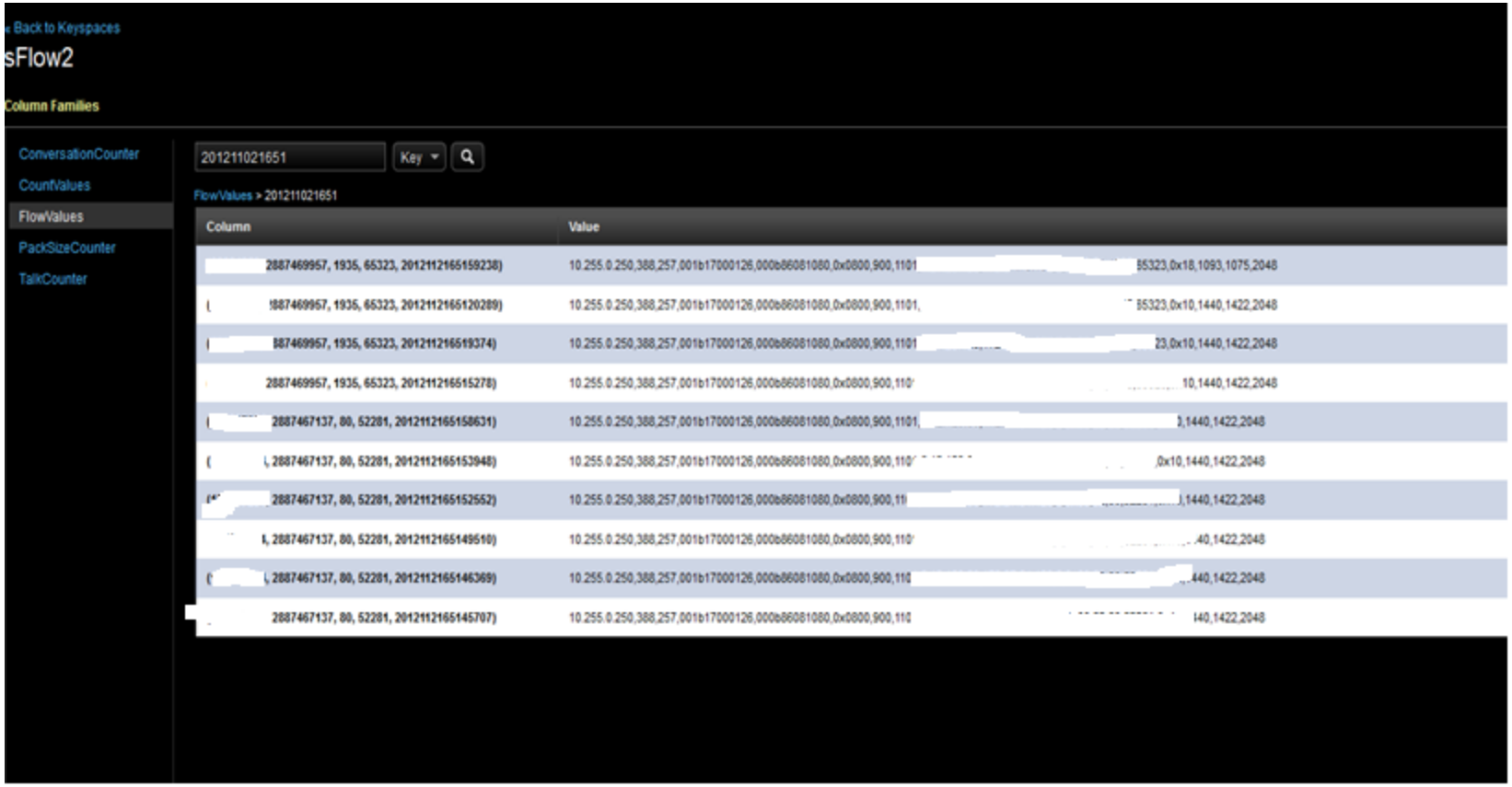
Demonstration

Cassandra Cluster



Demonstration

- Query by Key



Demonstration

- Measuring anonymity network usage on campus by using Pig scripting

It takes less than 10 minutes to process 205 million packets, about 1.44TB data, writing less than 200 lines of Pig scripting code.

Bingdong Li, Esra Edrin, Mehmet Hadi Gunes, George Bebis, Todd Shipley, A Study of Anonymity Technology Usage on the Internet, submitted to Computer Communication

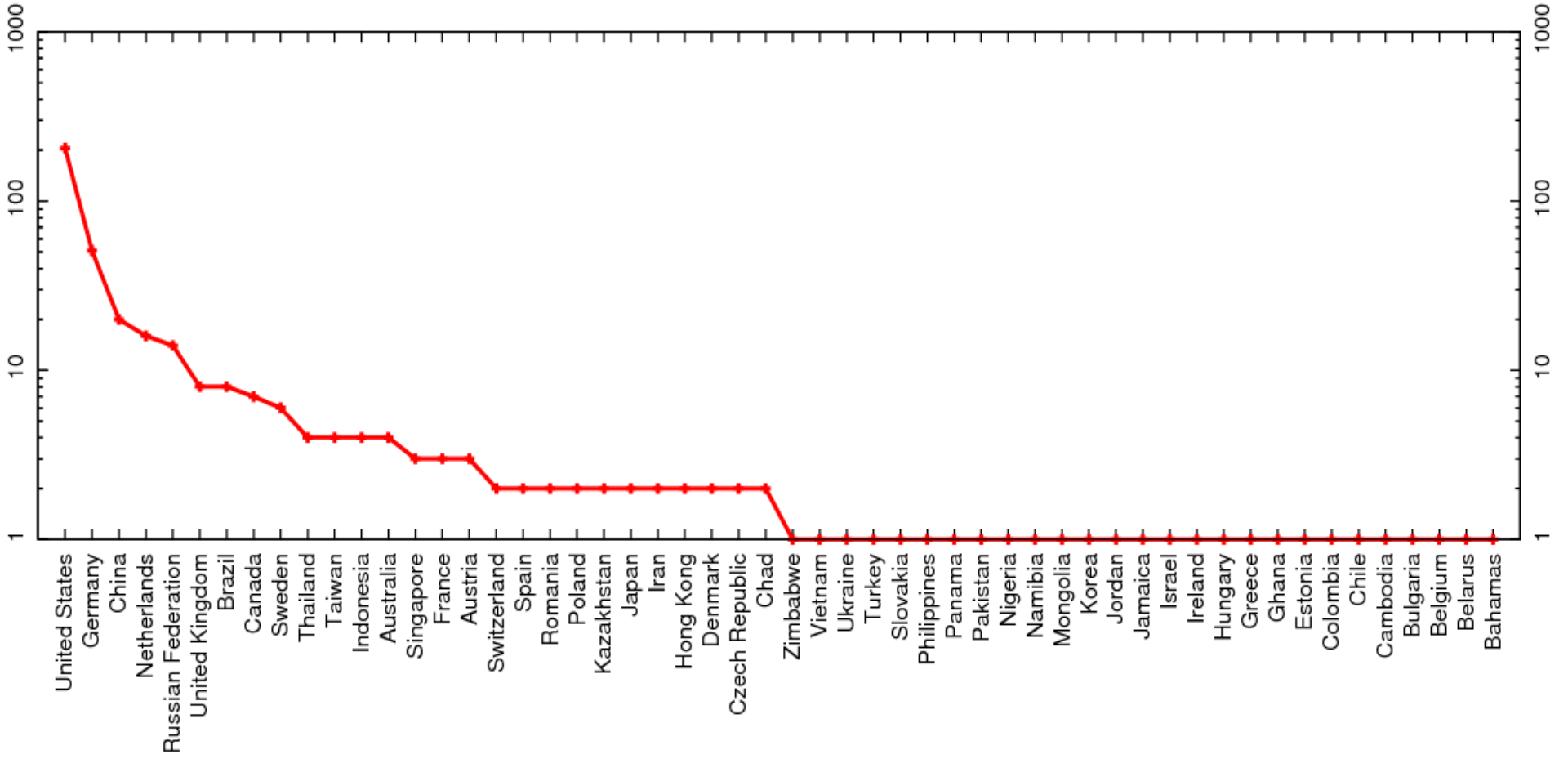
Demonstration

Analyzed Anonymity Networks

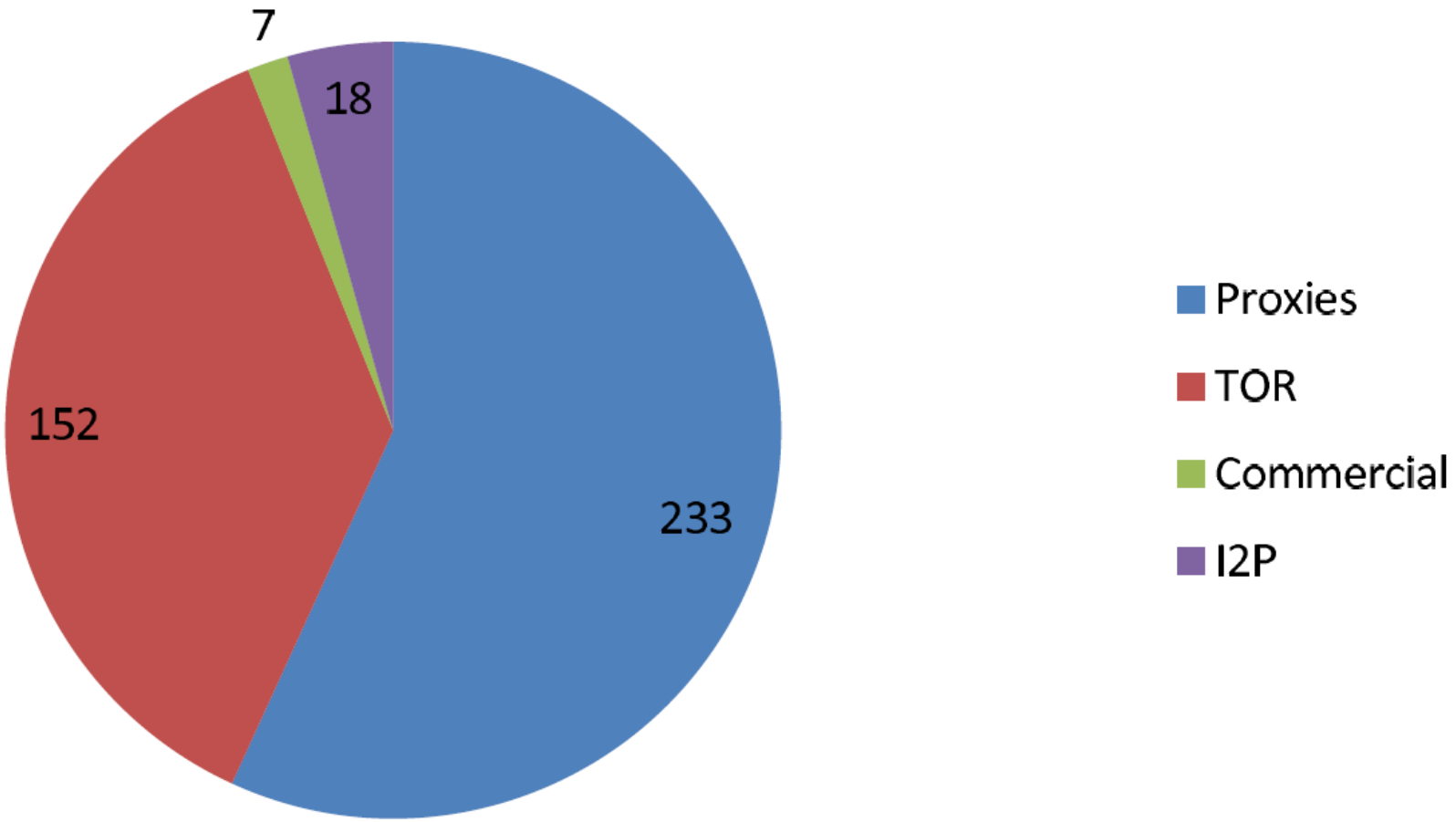
Network	Servers	Service
Tor	61,798	General
I2P	2,267	P2P
JAP	11	General
Remailers	15	Email
Proxies	7,246	General
Commercial	Anomymizer,Gotrusted	General

Bingdong Li, Esra Edrin, Mehmet Hadi Gunes, George Bebis, Todd Shipley, A Study of Anonymity Technology Usage on the Internet, submitted to Computer Communicatio

Anonymity Network Usage Geolocation



Anonymity Network Usage Distribution



Demonstration

- Example of Advanced Network Modeling
  - Model Host Role's Behaviors

Algorithms:

On-line SVM based on Bordes Methods

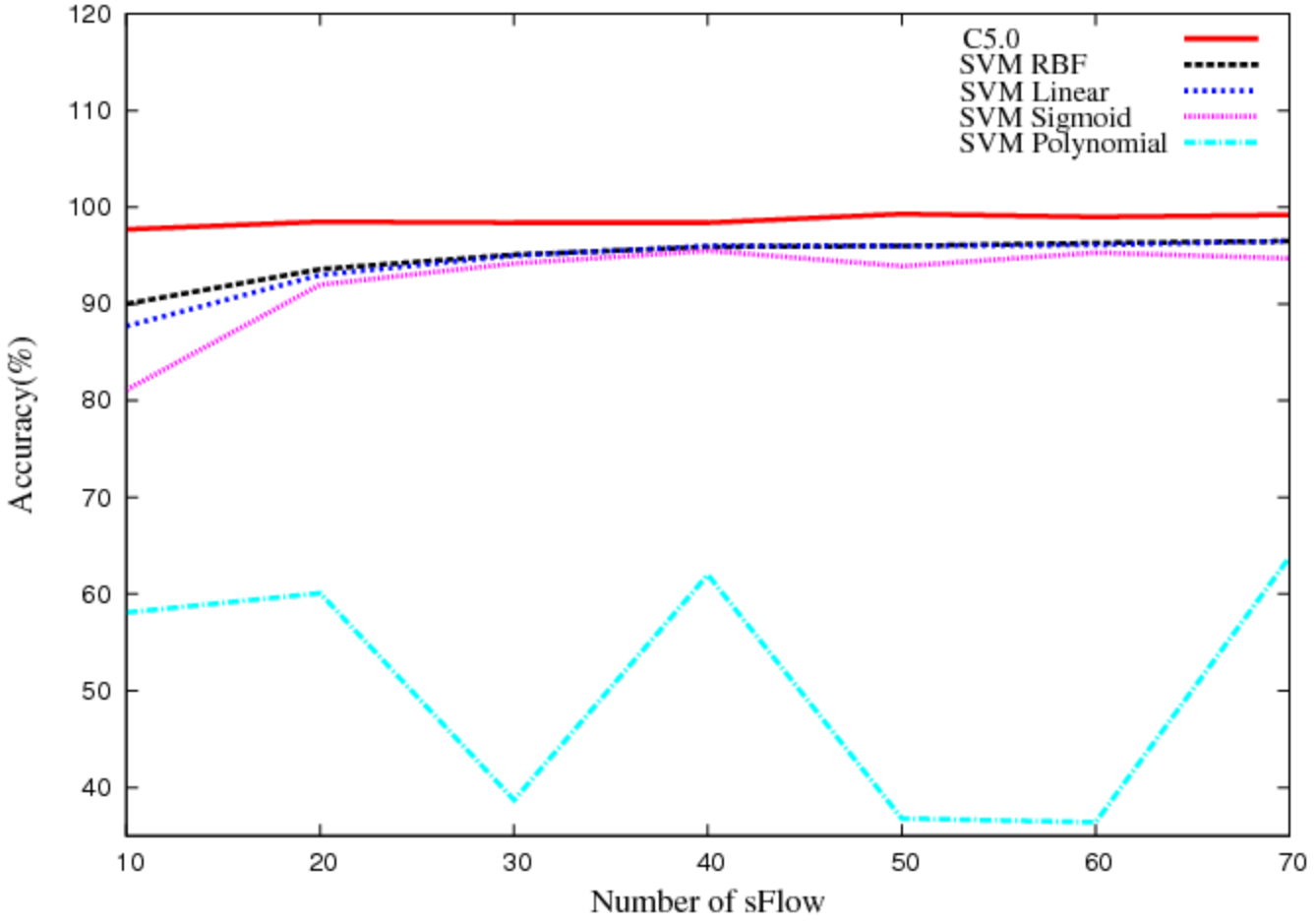
Ground Truth:

Host Information in Active Directory and vulnerability scanner Nessus database.

Antoine Bordes, etc. Fast kernel classifiers with online and active learning. Journal of Machine Learning Research, 6:1579–1619, September 2005.

Demonstration

Client vs Server Classification Accuracy



Thoughts and Pitfalls

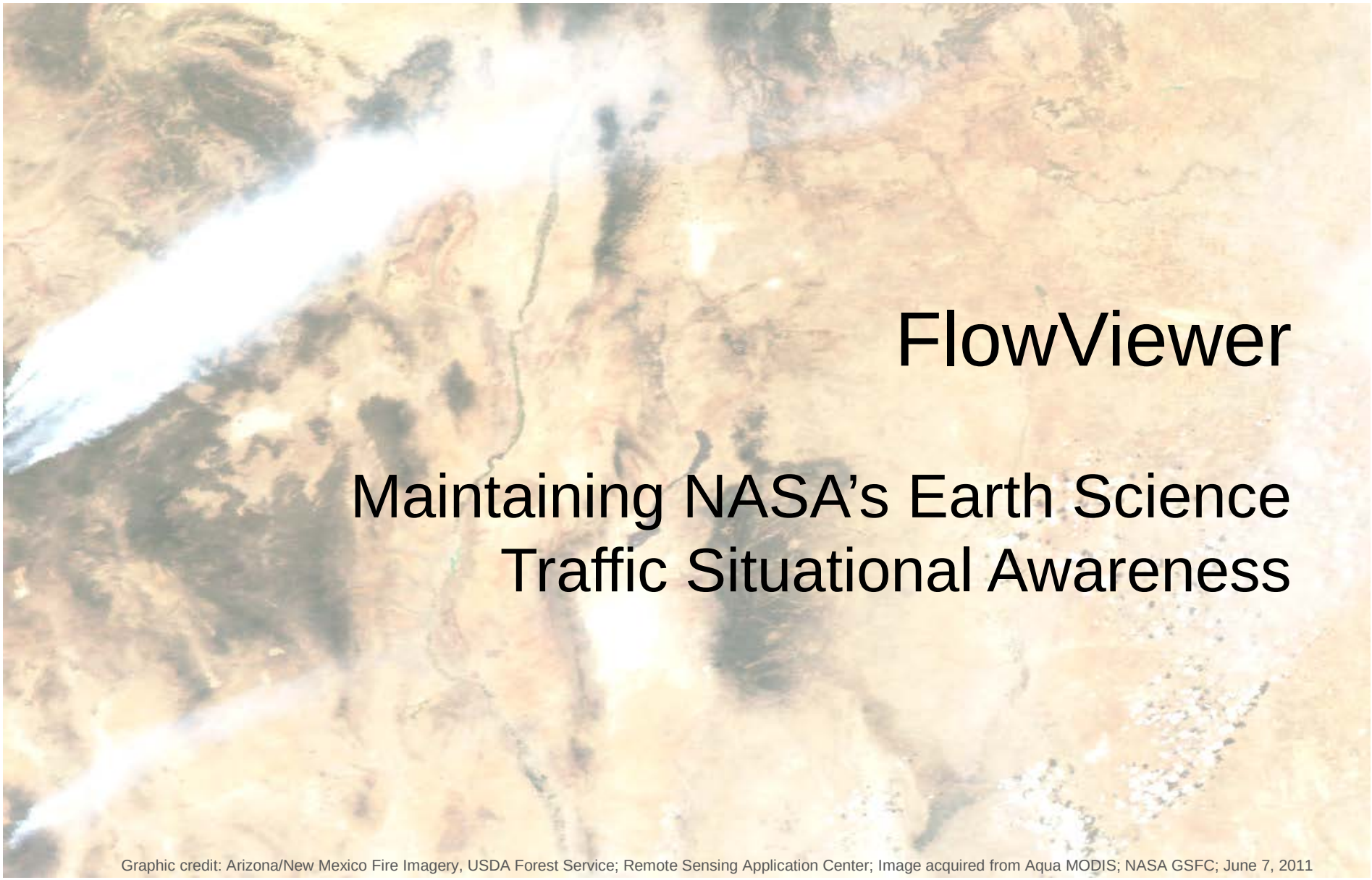
- Low Cost – Open Source, Distributed
- Be patient and careful for Incompatibility between different versions of components
- Be willing to learn, it is a new era of big data
- Cassandra Replica Factor = 1? Do not even try
- What do you do for Exception error? Handle, Ignore or throw it

Summary

- A design of distrusted real time network security system based on Apache Hadoop related technologies
- Demonstration
- Thoughts and pitfalls

Questions and Discussions

Contact:
Bingdong Li
bingdongli@unr.edu



FlowViewer

Maintaining NASA's Earth Science Traffic Situational Awareness

Graphic credit: Arizona/New Mexico Fire Imagery, USDA Forest Service; Remote Sensing Application Center; Image acquired from Aqua MODIS; NASA GSFC; June 7, 2011

FlowViewer provides a convenient web-based user interface to Mark Fullmer's flow-tools suite, and now with v4.0, CMU NetSA group's SiLK. The inclusion of the underlying SiLK tool set enables FlowViewer users to continue to use the tool with the newer IPFIX netflow data protocol.

FlowViewer has been developed for NASA's Earth Sciences Data and Information System (ESDIS) networks, and credit goes to NASA for their usual outstanding support of innovation.

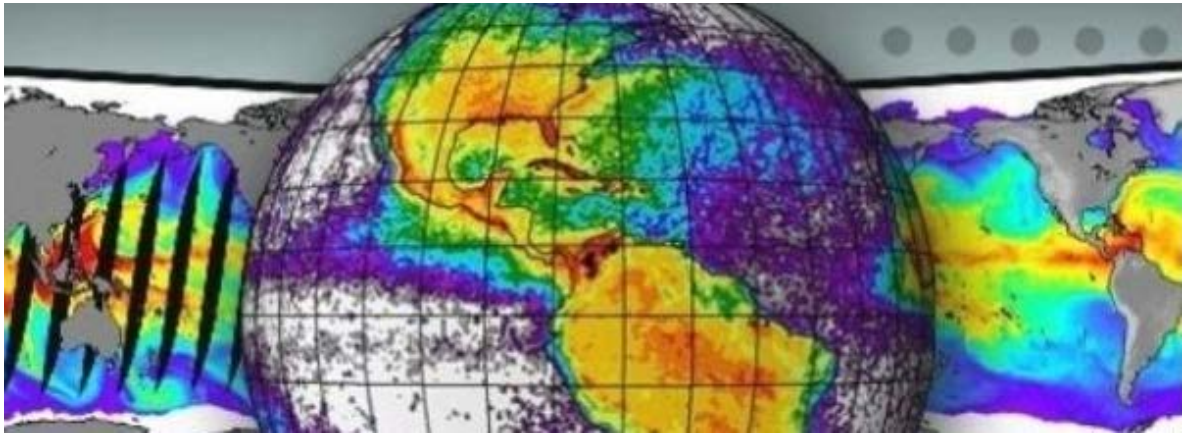
Graphic credit; Hurricane Sandy, October 29, 2012 Captured by Aqua MODIS; EOSDIS Website; NASA official: Kevin Murphy

- Complete open-source netflow collector analyzer
- Web-based UI provides dynamic front-end to open source collectors
- Dashboard provides user keep network traffic 'situational awareness'
- Ability to analyze IPFIX netflow (e.g., v9) data captured by SiLK
- Ability to continue to support netflow v5 installations via flow-tools
- Users can graph filtered traffic sets across a specified time period
- Background software tracks filtered traffic over long-term (ala MRTG)
- Ability to save filters and reports for later use and review
- Users can be alerted by email to abnormal data traffic situations

NASA Earth Science Network use of Netflow Data

EOSDIS NASA's Earth Observing System Data and Information System

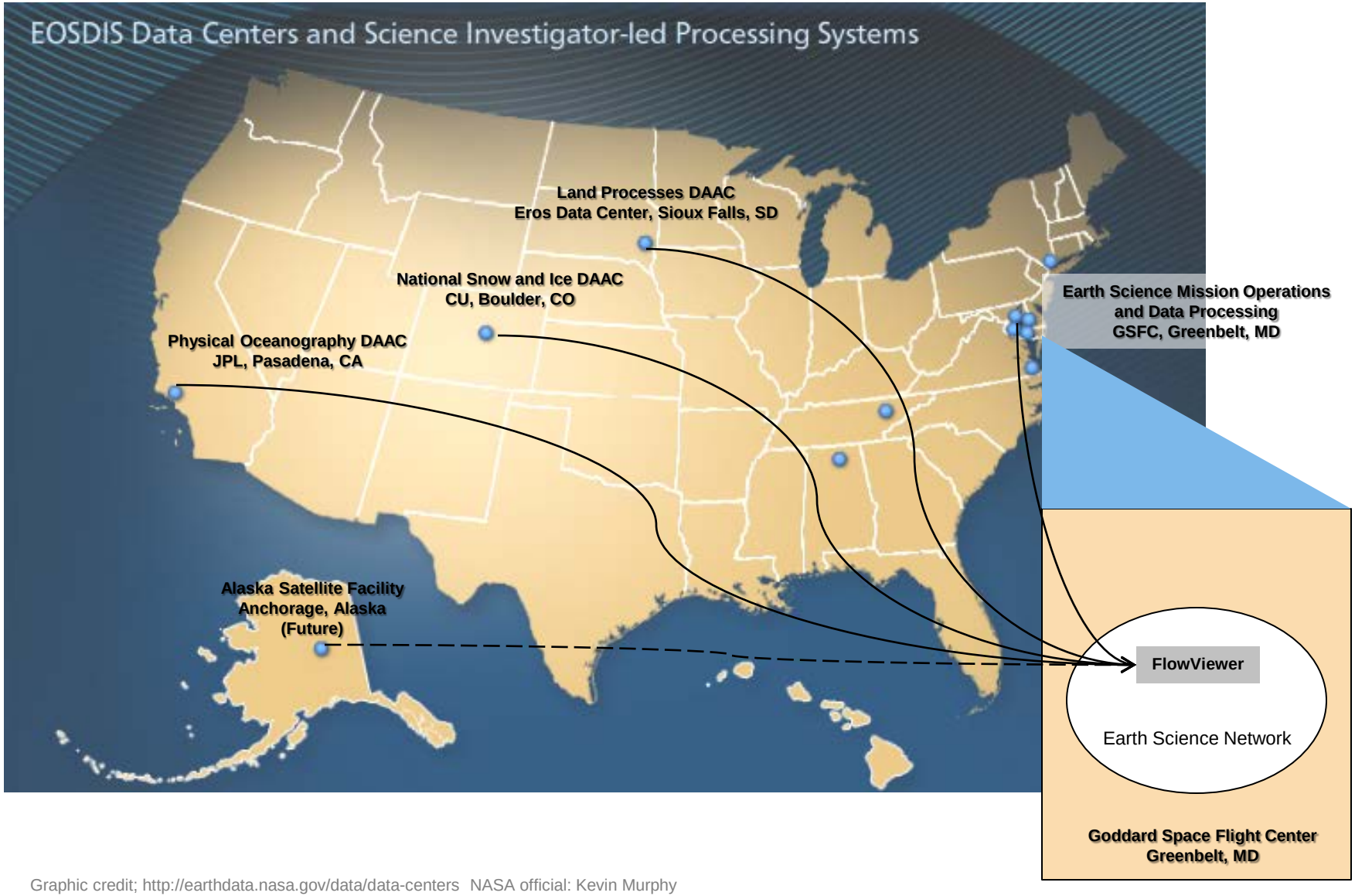
The Earth Observing System Data and Information System (EOSDIS) is a core capability in NASA's Earth Science Data Systems Program. It provides end-to-end capabilities for managing NASA's Earth science data from various sources – satellites, aircraft, field measurements, and various other programs. The EOSDIS serves a broad international community of Earth Science and meteorological scientists and users. Several TBytes of satellite and science data traverse its network every day.



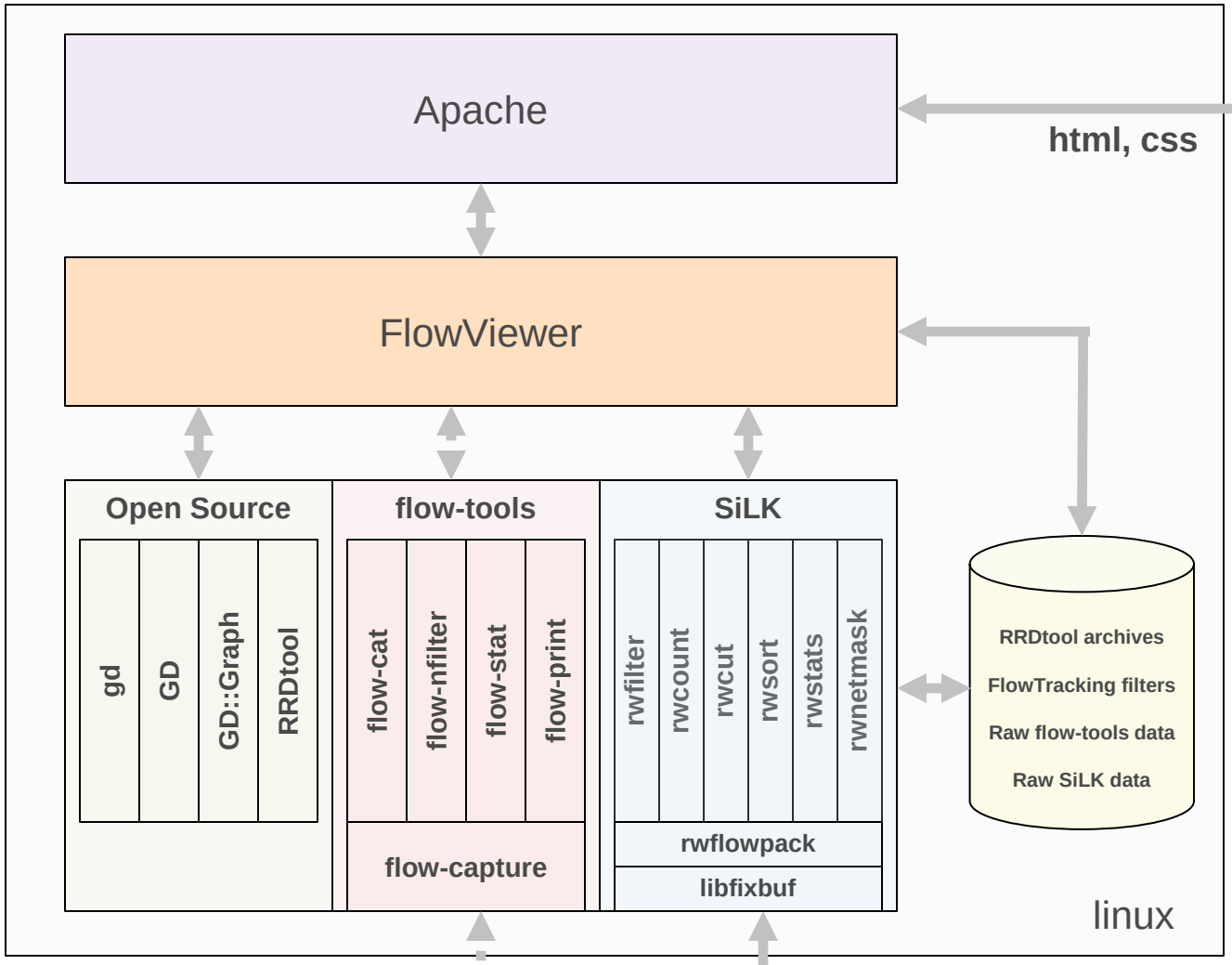
- In 2003 NASA and CSC worked to capture netflow data to help monitor traffic
- Initial capture/analysis system was based on *'cflowd'*
- FlowViewer was developed to aid traffic analysis (away from the command line)
- Today, NASA monitors over 200 Earth Science flows of interest (FlowTrackings)

Graphic credit; OPeNDAP is a data transport architecture and protocol widely used by Earth scientists to access remotely distributed data; EOSDIS Website; NASA official: Kevin Murphy

Netflow data sources

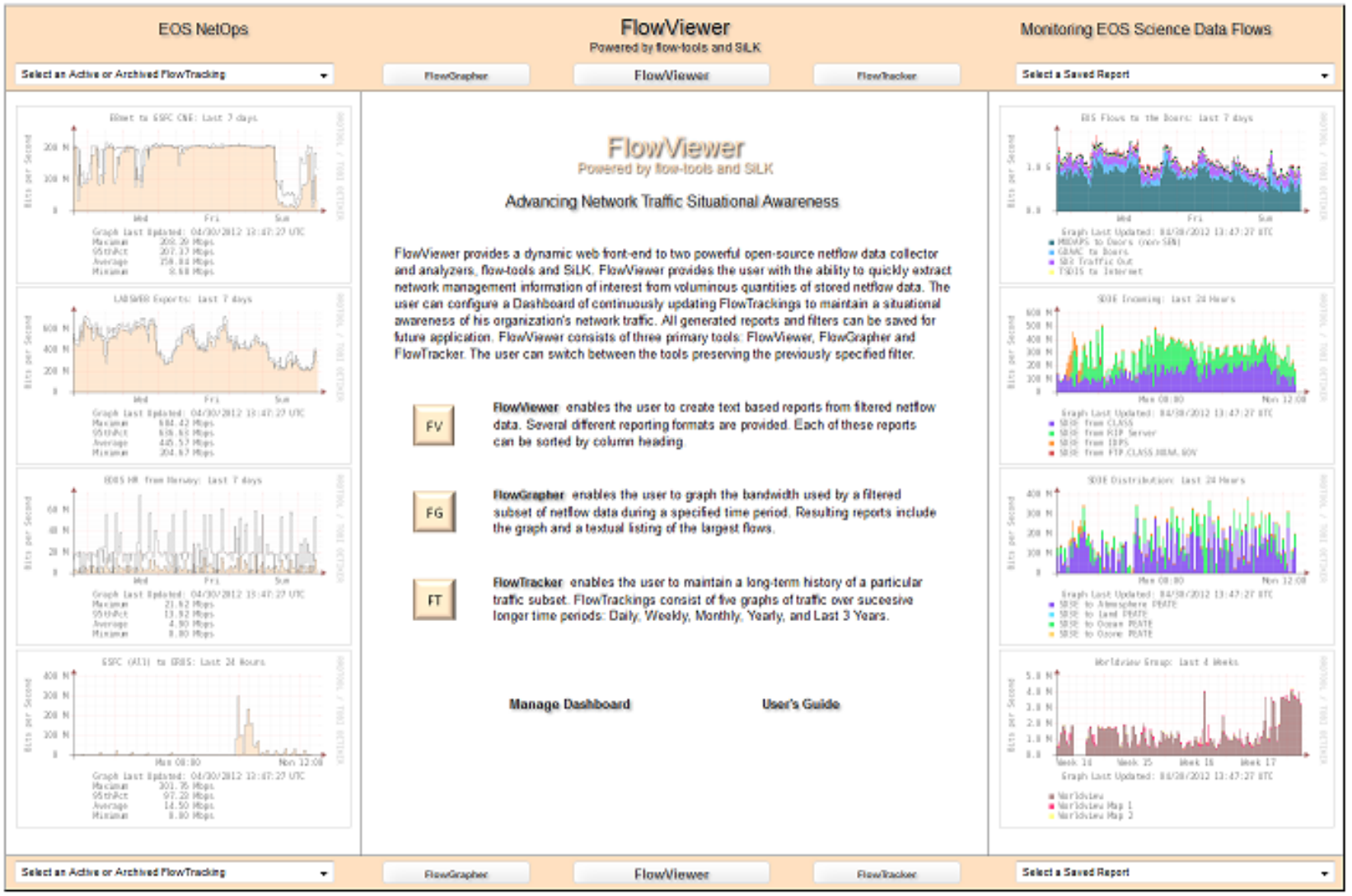


Graphic credit; http://earthdata.nasa.gov/data/data-centers NASA official: Kevin Murphy

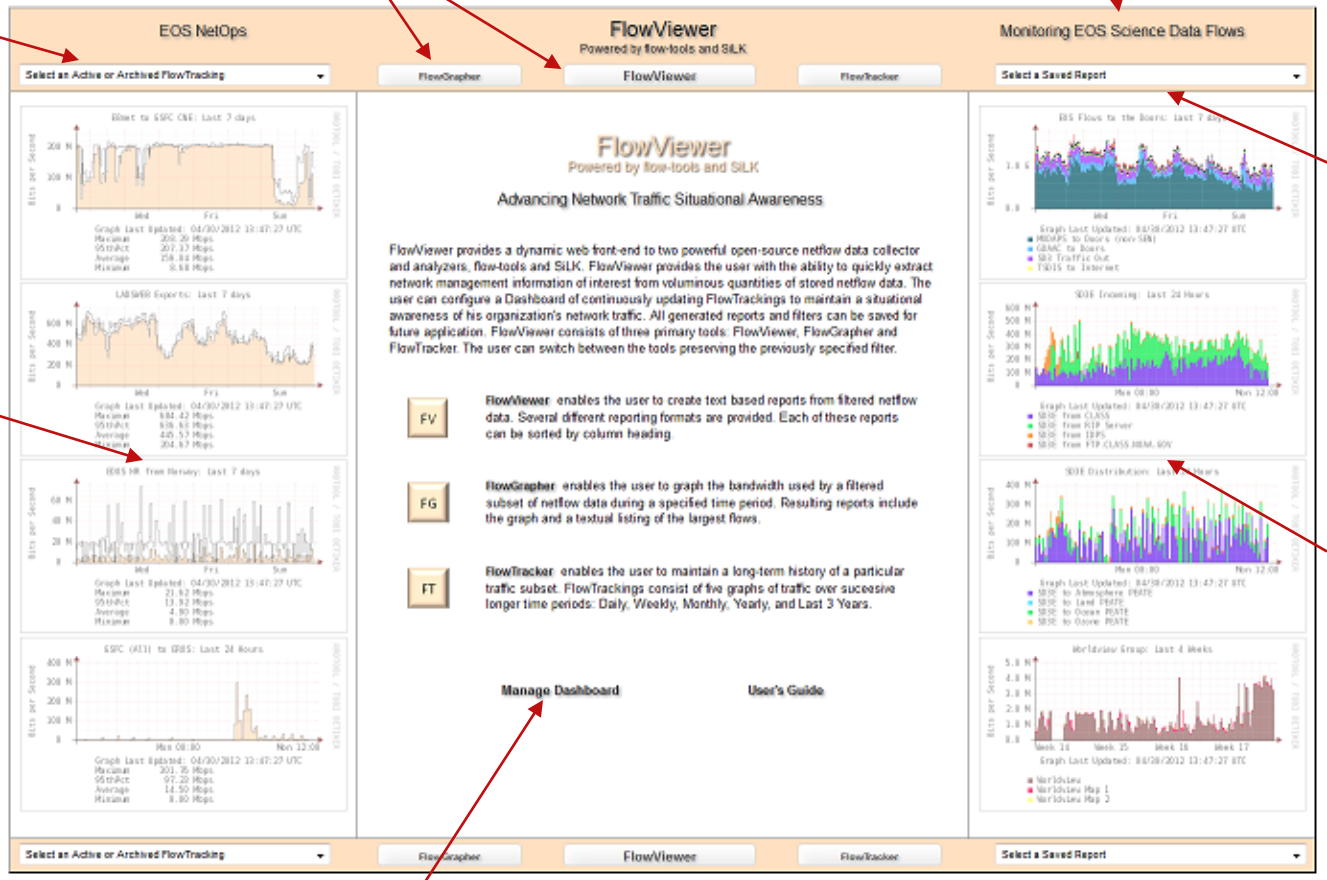


FlowViewer is an entirely open source netflow collector, analyzer and reporter. HTML/CSS user interface provides easy and wide deployability.

FlowViewer Main Screen

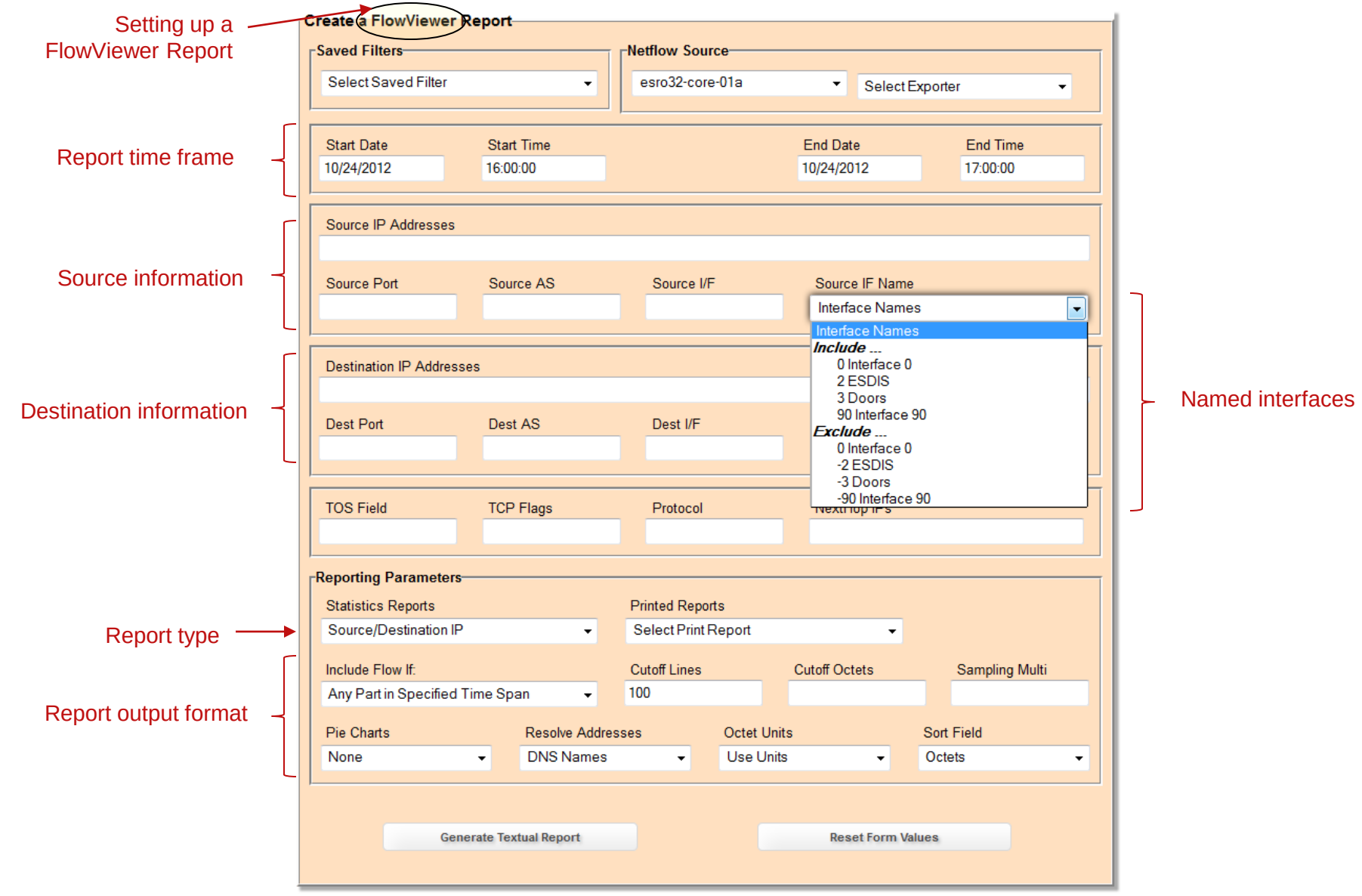


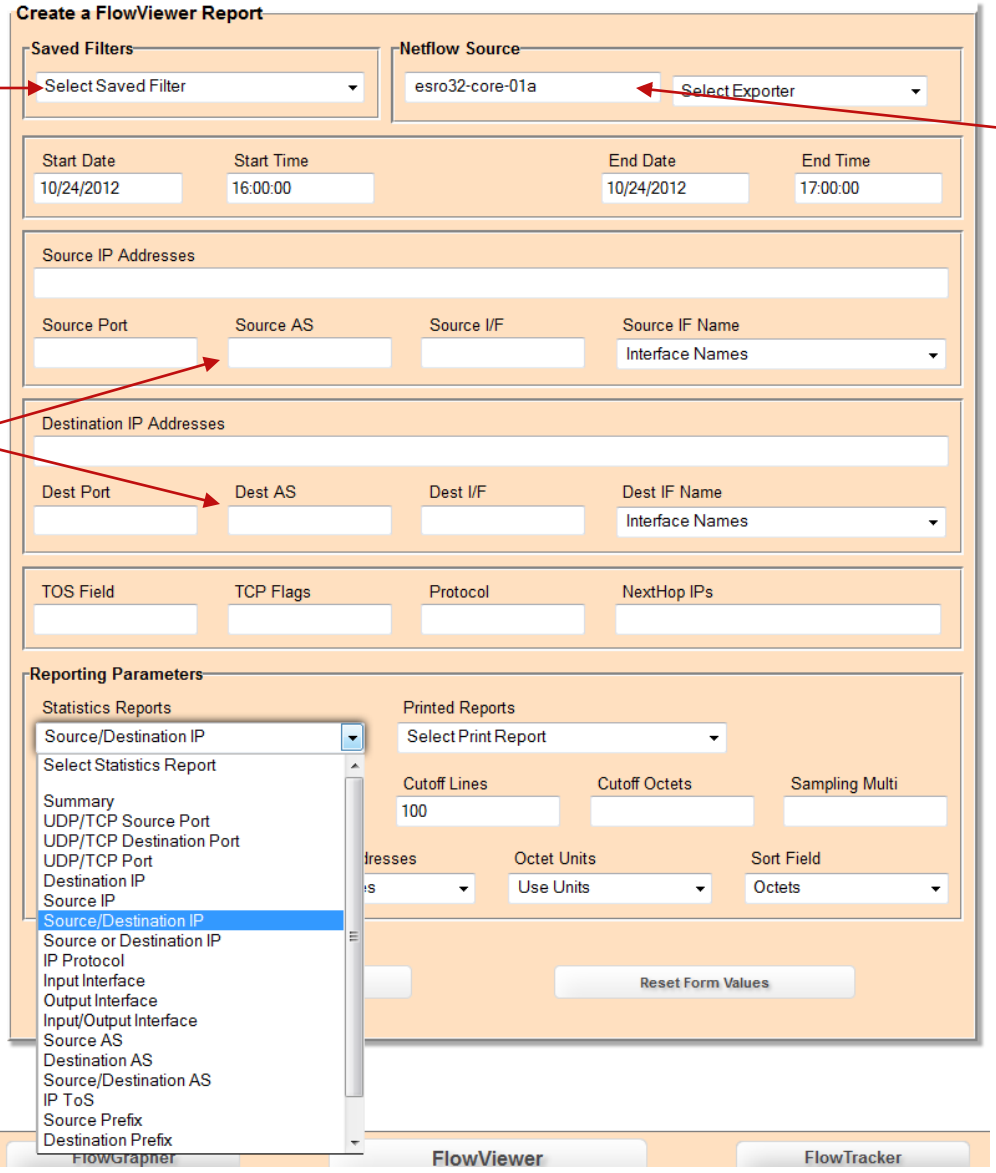
FlowViewer Main Screen



FlowViewer Input Screen

FlowViewer Input Screen - 1





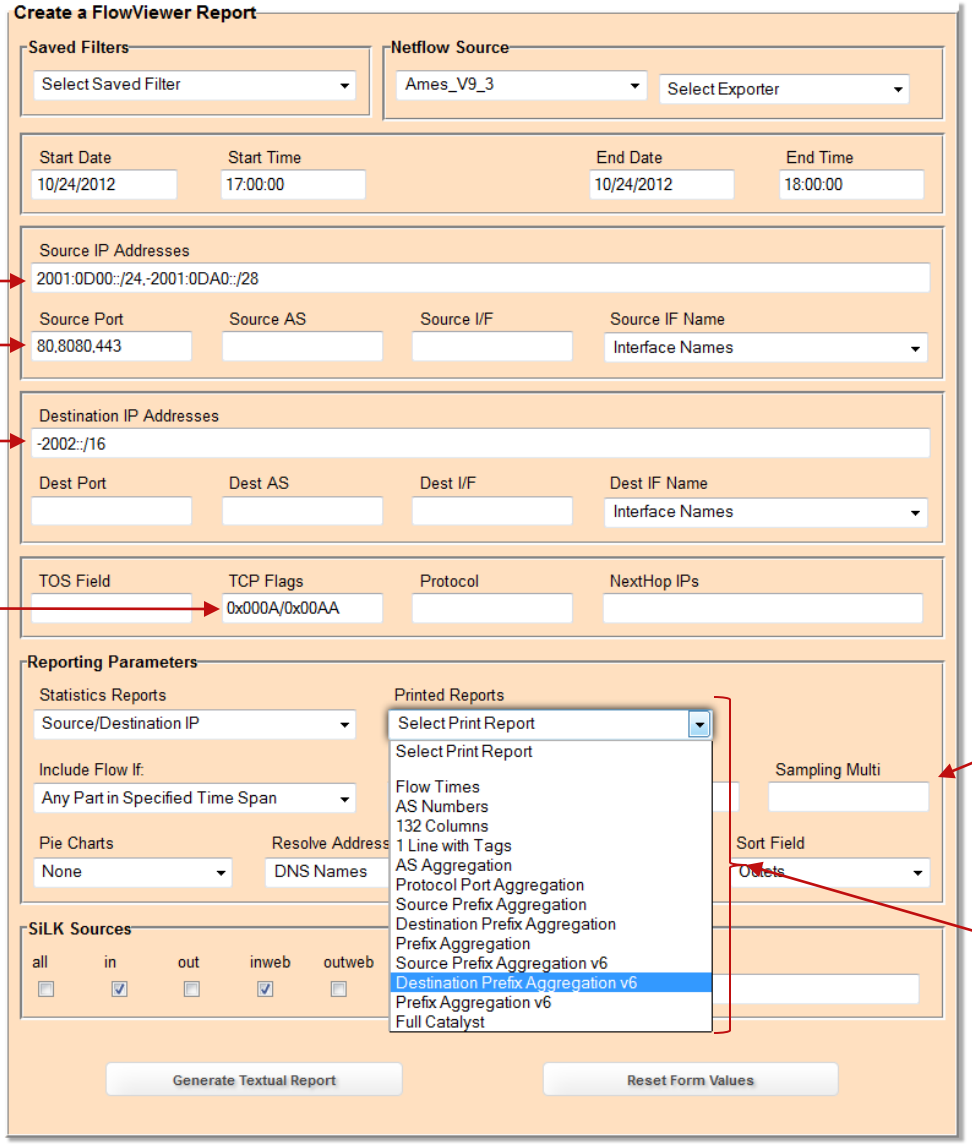
Reuse saved filter

Select from different devices

Autonomous systems (flow-tools only)

Report types

FlowViewer Input Screen - 3



Excluding within a network

Multiple entries

Excluding
(works on all fields)

TCP Flags

When using SiLK devices

Sampling multiplier

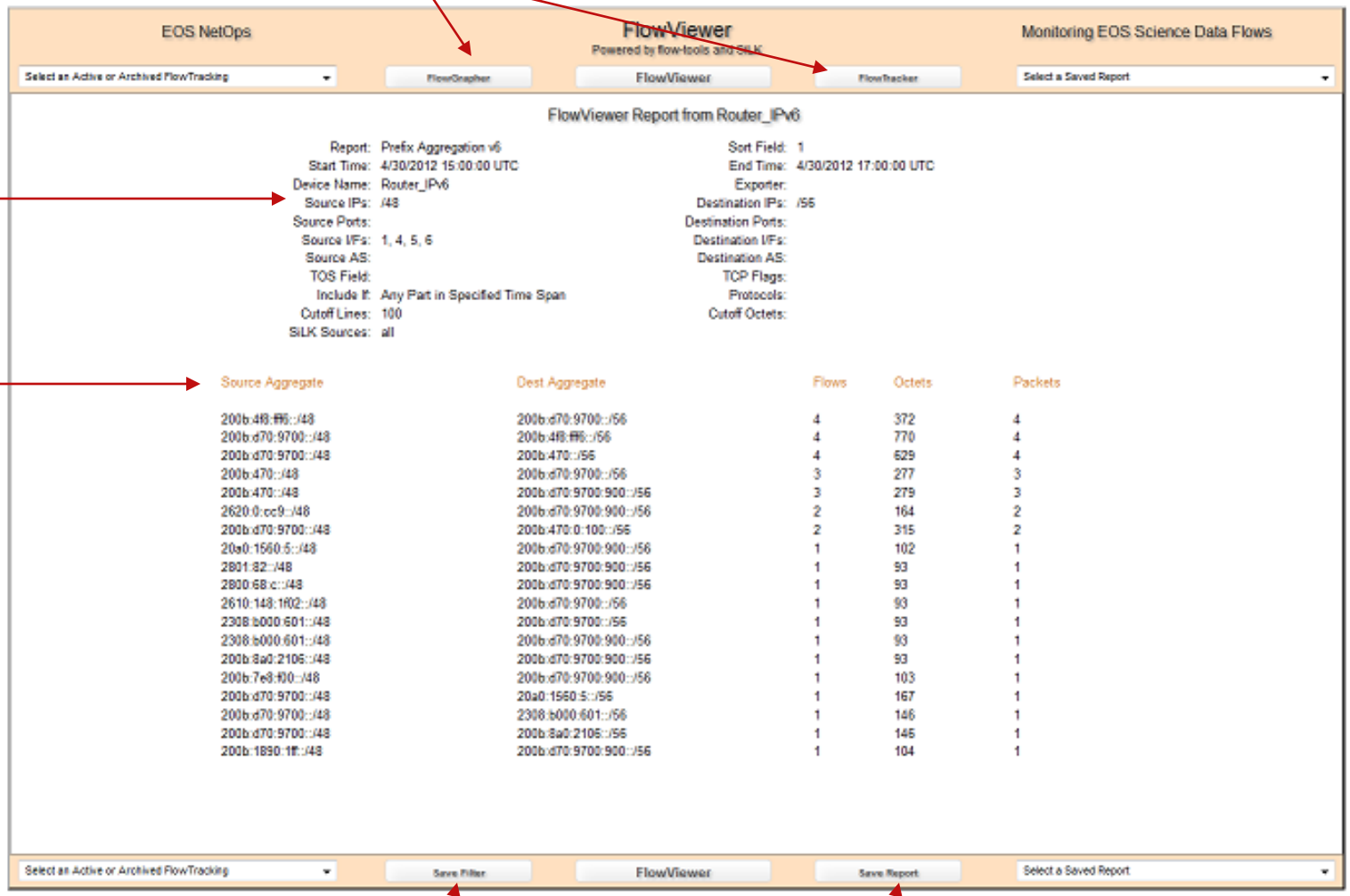
Additional reports

FlowViewer Report

Can switch to other tools with filtering criteria preserved

Aggregation filtering

Sortable by column



Save the filter

Save the report

Setting up a FlowGrapher Report

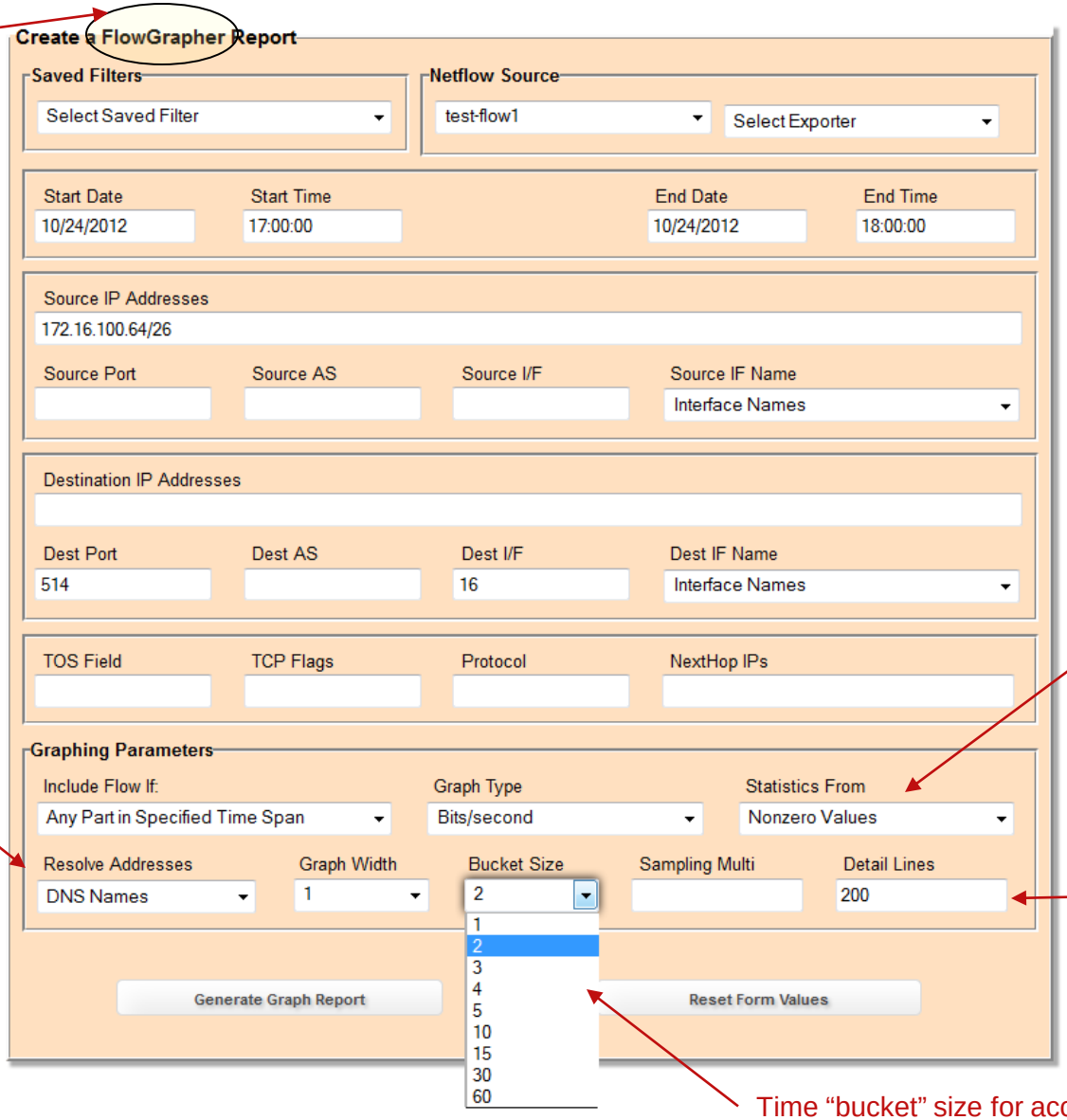
Same filtering criteria

Resolved host names or IP addresses

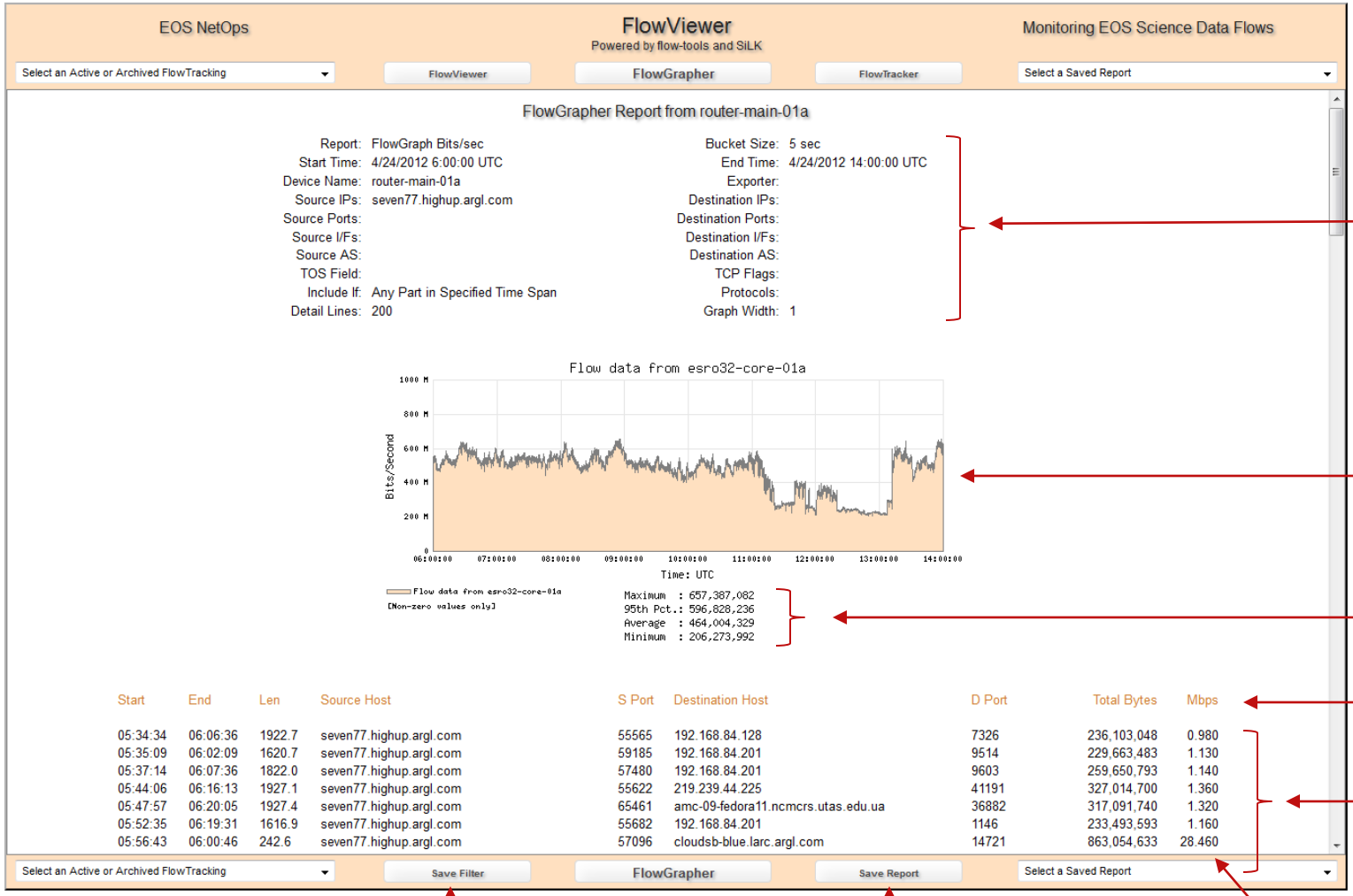
How to determine statistics (Max, 95th, Avg, Min)

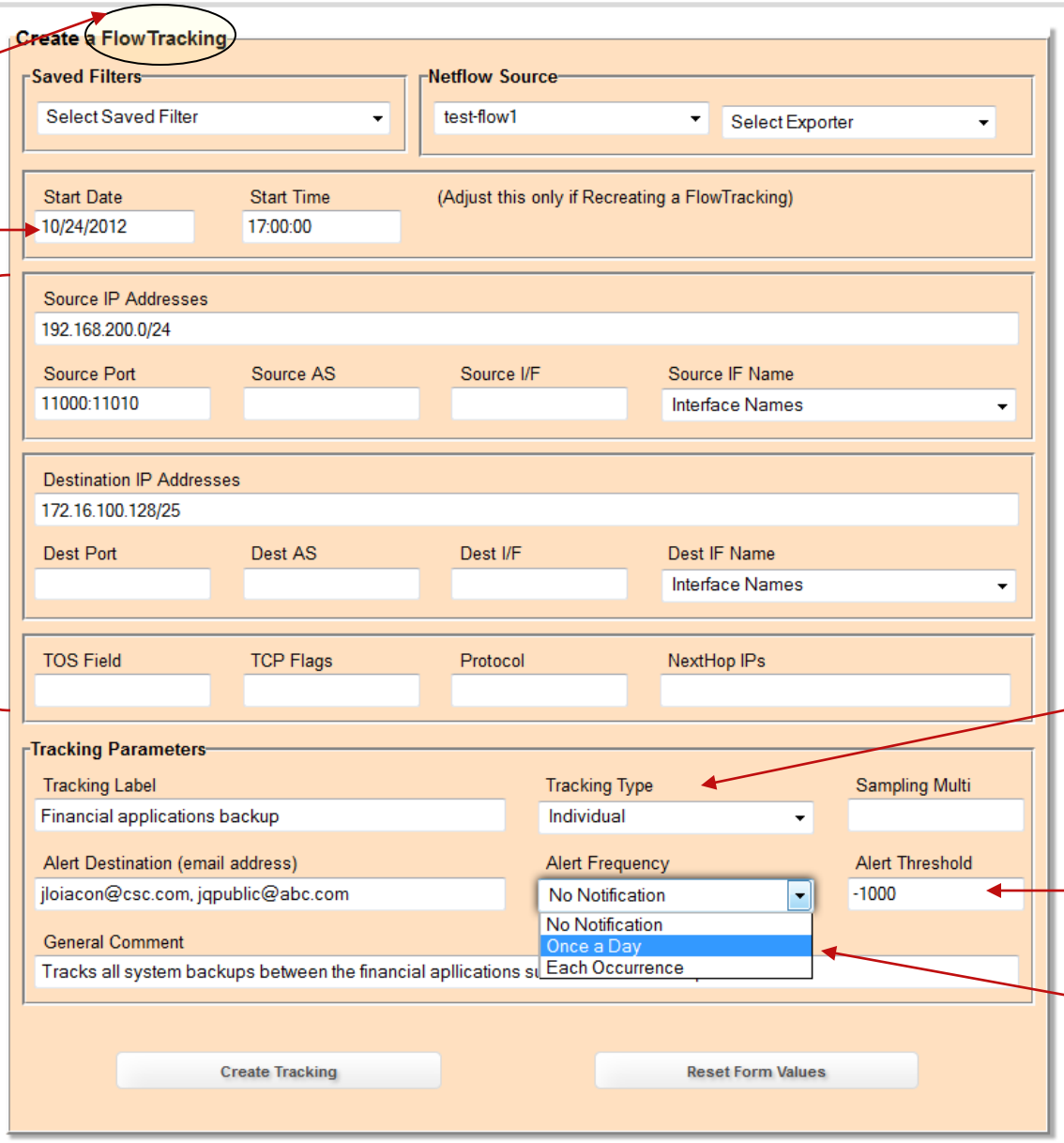
Number of longest flows to list in detail

Time "bucket" size for accumulating bits / period



FlowGrapher Report





Creating a FlowTracking

Option to start a FlowTracking in the past

Same filtering criteria

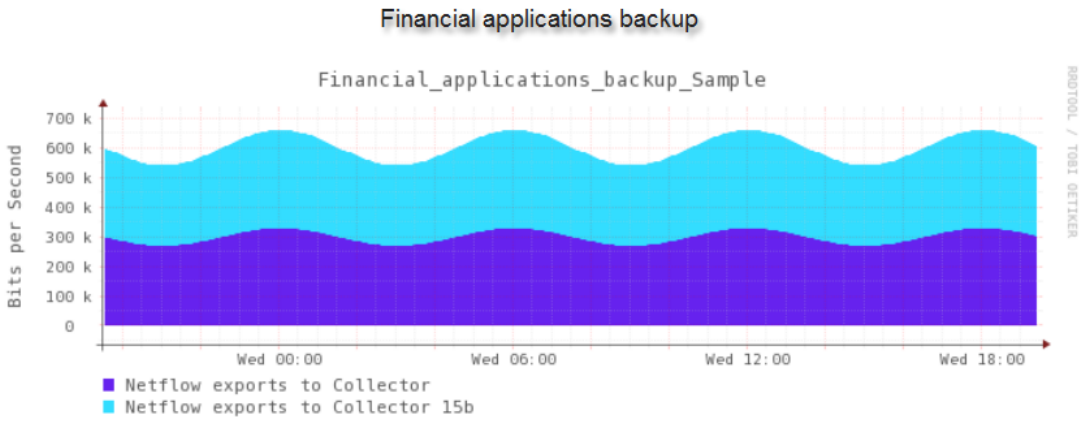
Individual or Group FlowTrackings

Email alerting

Alert thresholds

Alert frequency choices

'Groups' stack Individual FlowTrackings →



Select Individual FlowTracking Group components →

Select an existing Tracking to be a component of this Group:

Select a Component: ASTER to GSFC After

Place this Component: Above the X-axis

Select a Color: Auto Mixed

Add Component | Reset Values

← Can have components above and below X-axis

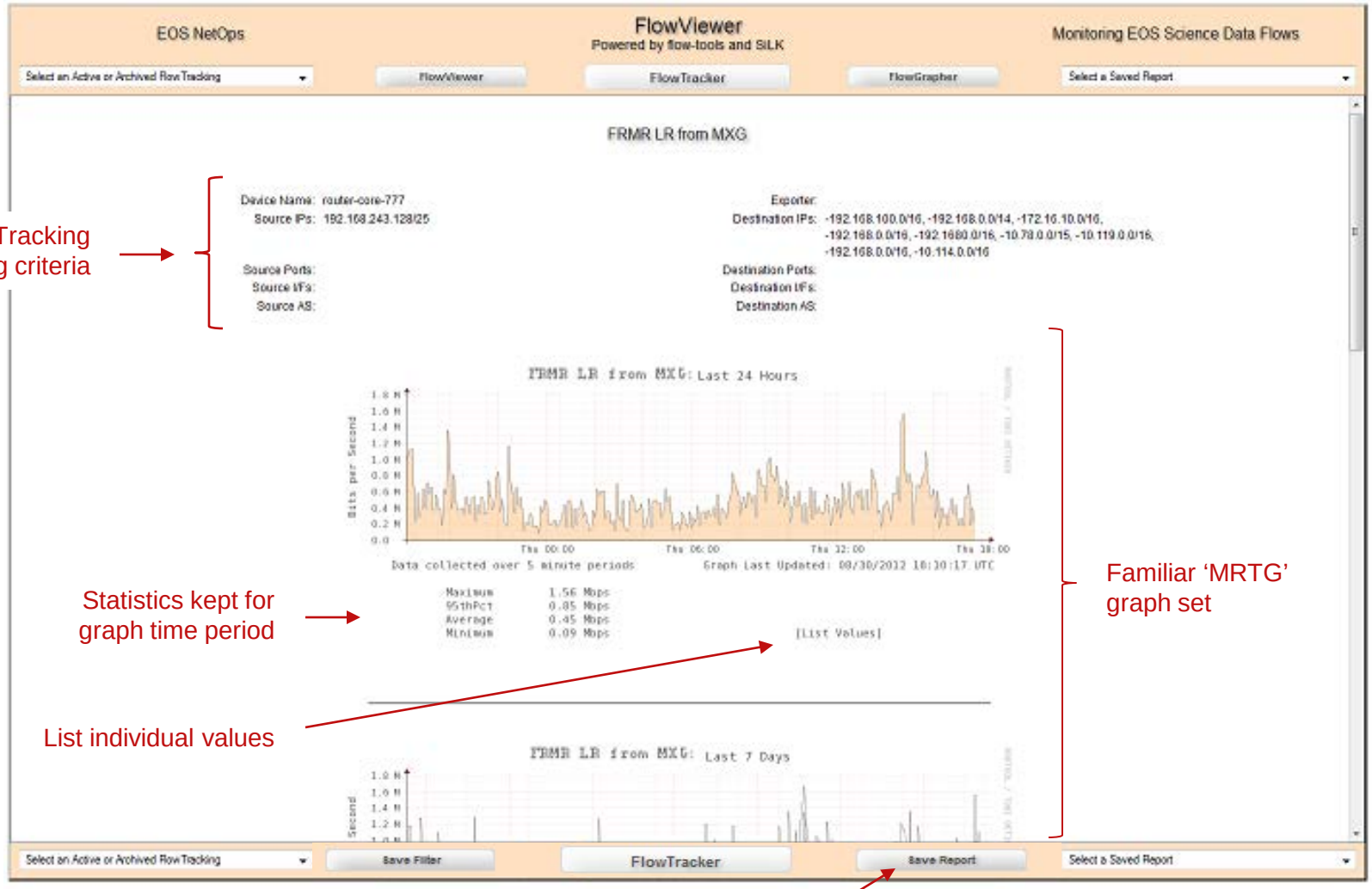
This group is composed of these components:

Adjust Group components →

Netflow exports to Collector 15b	Above 02	auto mixed2	New Color:	auto mixed2	Move:	Leave Alone
Netflow exports to Collector	Above 01	auto mixed1	New Color:	auto mixed1	Move:	Leave Alone

Adjust the Group | Reset Values

FlowTracker Report – General Example



FlowTracking filtering criteria

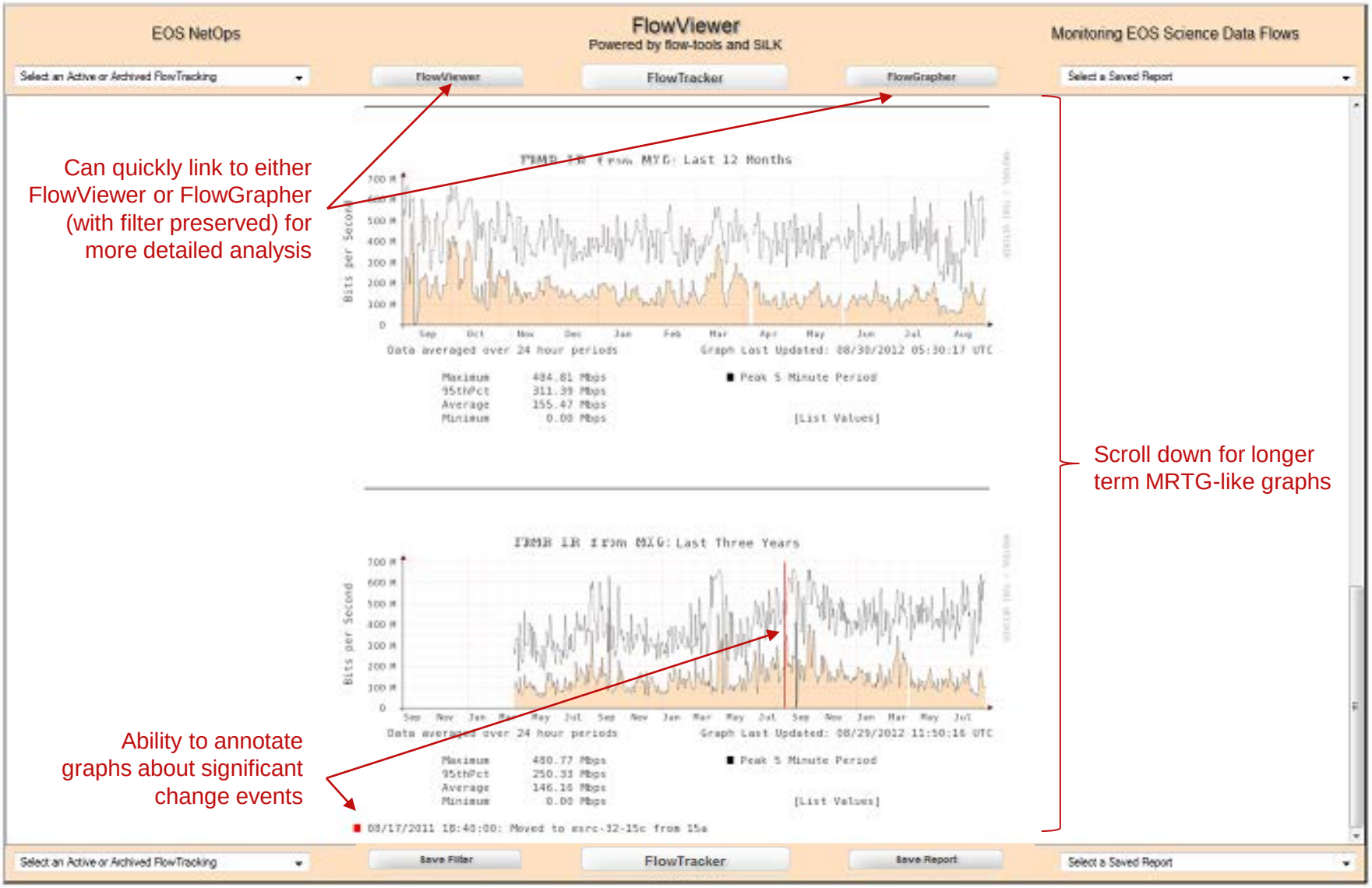
Statistics kept for graph time period

List individual values

Familiar 'MRTG' graph set

Save Report

FlowTracker Report – General Example, cont.



FlowTracker Report – Group Example

This FlowTracking documents the delivery of NPP data to the University of Wisconsin. One can see a switch from two (Atmospheric Science) servers to one only, and then all to the other of the pair.

Access to all saved reports

This is an example where you might want to save a FlowTracking

FlowTracker Report – Group Case Study

Each legend item 'hyperlinks' back to the Individual FlowTracking

This example depicts a situation where traffic shaping was invoked to manage limited network resources. This FlowTracking Group helps identify if perhaps there is one 'big player' for which a different network arrangement might mitigate the problem

Pulldown of all FlowTrackings

Listing of all FlowTrackings

Ability to ‘Revise’, ‘Rename’, ‘Archive’, ‘Remove’, and ‘Restore’ FlowTrackings

Case Studies

Components of an Interface

Satellite data in

Science data out

To service provider

Gray line preserves highest 5-minute measurement over the longer term graphs

These graphs help NASA monitor an expensive high-rate circuit between a polar ground station in Norway and the GSFC in Maryland. The circuit is shared with other Federal agencies through the use of MPLS tunnels.

This depression of peak values indicates that there may be an issue with the network or the MPLS tunnel (or the servers, or software or, …)

EOSDIS NASA's Earth Observing System Data and Information System

EOS NetOps

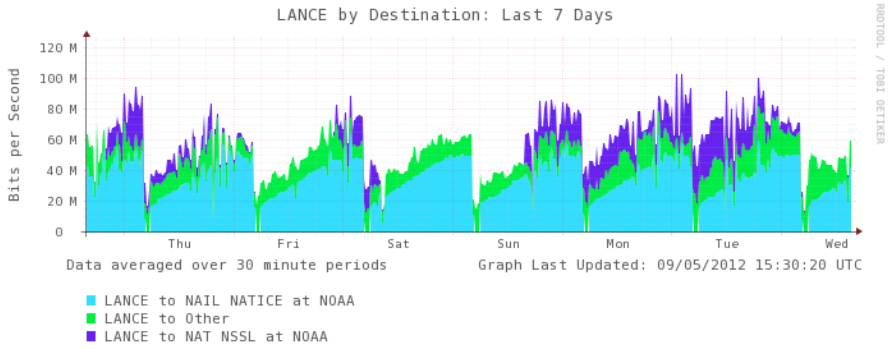
FlowViewer
Powered by flow-tools and SiLK

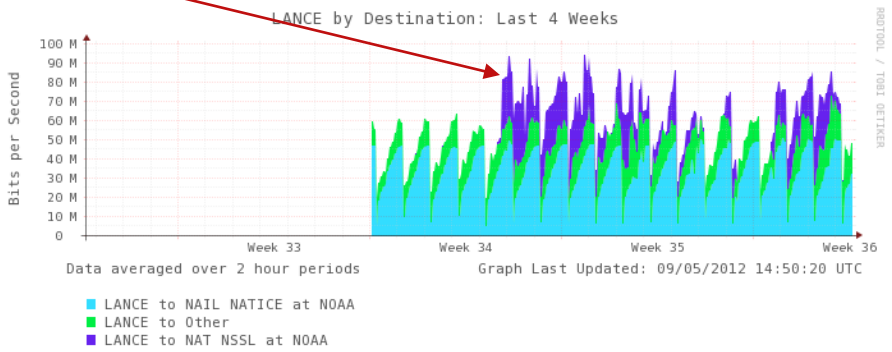
Monitoring EOS Science Data Flows

Select an Active or Archived FlowTracking | FlowViewer | FlowTracker | FlowGrapher | Select a Saved Report

Around the time of last summer's hurricane Isaac, Land, Atmosphere Near-Real-Time Capability for EOS (LANCE-MODIS*) system managers noted a sharp increase in traffic.

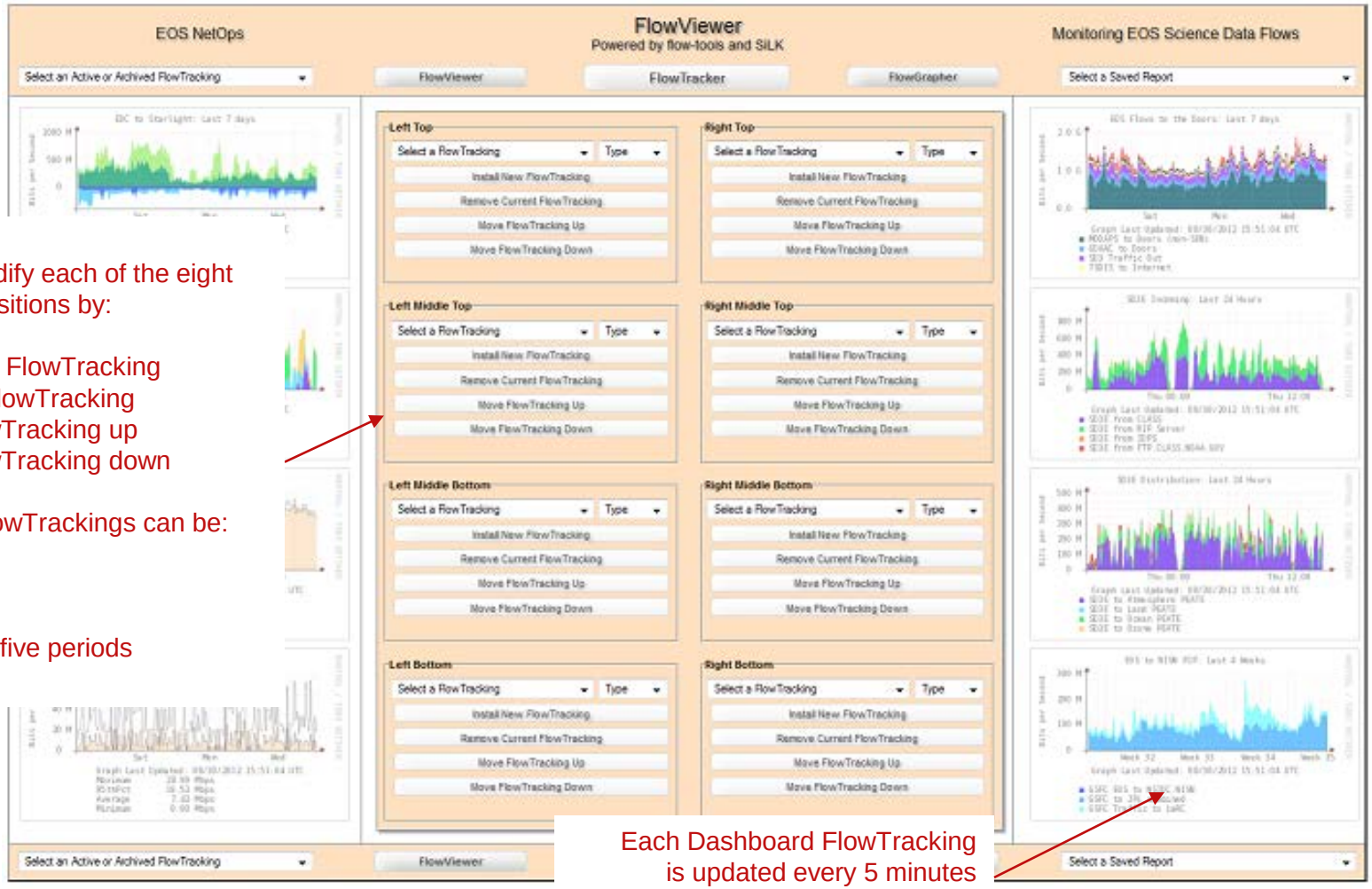
The FlowTracker Re-create capability was invoked to create a FlowTracking Group which isolated the new user that had come on line: the National Severe Storms Laboratory.





* MODIS - Moderate Resolution Imaging Spectroradiometer

Select an Active or Archived FlowTracking | FlowViewer | FlowTracker | FlowGrapher | Select a Saved Report



Users can modify each of the eight Dashboard positions by:

1) Install new FlowTracking
2) Remove FlowTracking
3) Move FlowTracking up
4) Move FlowTracking down

Dashboard FlowTrackings can be:

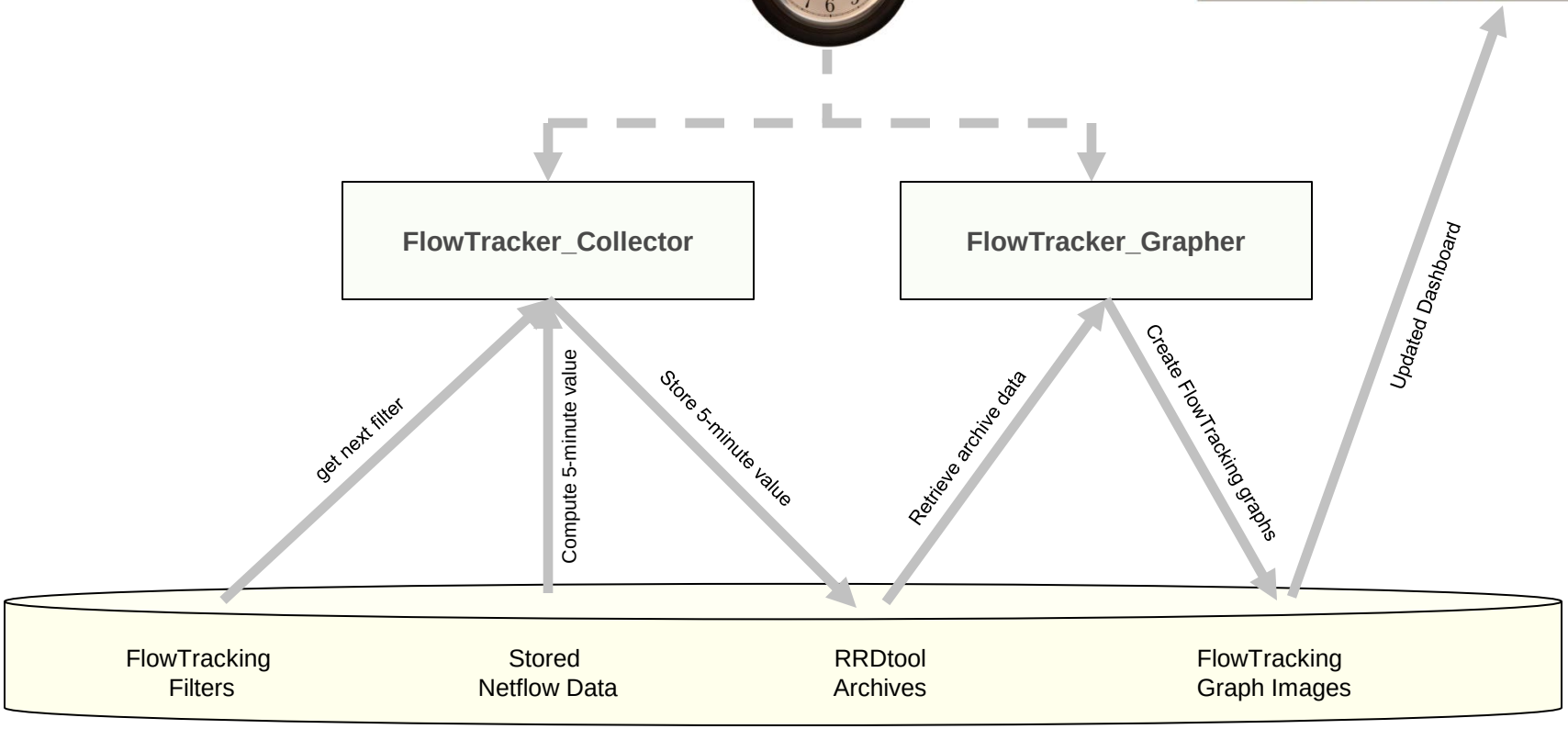
1) Individual
2) Group
3) Any of the five periods

.

Each Dashboard FlowTracking is updated every 5 minutes

Each Dashboard graph links back to the original FlowTracking

Upon FlowViewer installation, the FlowTracker_Collector and FlowTracker_Grapher scripts are placed in the Linux background. They will “wake up” every five minutes and collect a 5-minute value for each active FlowTracking. The FlowTracking and Dashboard graphs are updated with the latest data point.



- FlowViewer distribution includes “analyze_netflow_packets” utility
- FlowViewer has supported flow-tools for over five years; but is new to SiLK
- Integration with SiLK may not be optimized as a result
- Would welcome SiLK related improvement suggestions
- At the same time … some ‘requests’ of SiLK ☺. Please include:
  - IPFIX Information Element (IE) [5]: ipClassOfService
  - IPFIX Information Element (IE) [16]: bgpSourceAsNumber
  - IPFIX Information Element (IE) [17]: bgpDestinationAsNumber
  - IPFIX Information Element (IE) [70]: mplsLabelStackSection
  - IPFIX Information Element (IE) [71]: mplsLabelStackSection2
  - IPFIX Information Element (IE) [72]: mplsLabelStackSection3

Thank You

Joe Loiacono
Network Engineer, CSC
jloiacon@csc.com

http://earthdata.nasa.gov/esdis

NASA Official: Kevin Kranacs
Manager, ESDIS Networks

FlowViewer is available from:
https://sourceforge.net/projects/flowviewer

Considerations for Scan Detection Using Flow Data.

REDJACK

REDJACK

Overview

- Scans and scan detection – goals and objectives
- A review of Threshold Random Walk
- Real time vs. Flow based approaches
- Bi-flows and Oracles
- Extensions
  - to ICMP and UDP
  - indeterminate reduction to improve benign detection
- Beyond detection – actionable intelligence
- Comparisons with `rwscan`
- Conclusions and future directions

Scans and scan detection goals and objectives

REDJACK

- At one time 90% of internet traffic was scanning
  - Now about 10% or so, so why do we care
- Still a viable propagation mechanism for malware
  - many newly compromised machines scan locally
  - scanning of entire internet happens, e.g. sip server
    - *Analysis of a "/0" Stealth Scan from a Botnet* – CAIDA
- Scan detection provides situational awareness
  - What is sought, who is looking on a global level
- Responses provide local inventory
- Interactions with scanners can identify compromise
  - actionable in many cases

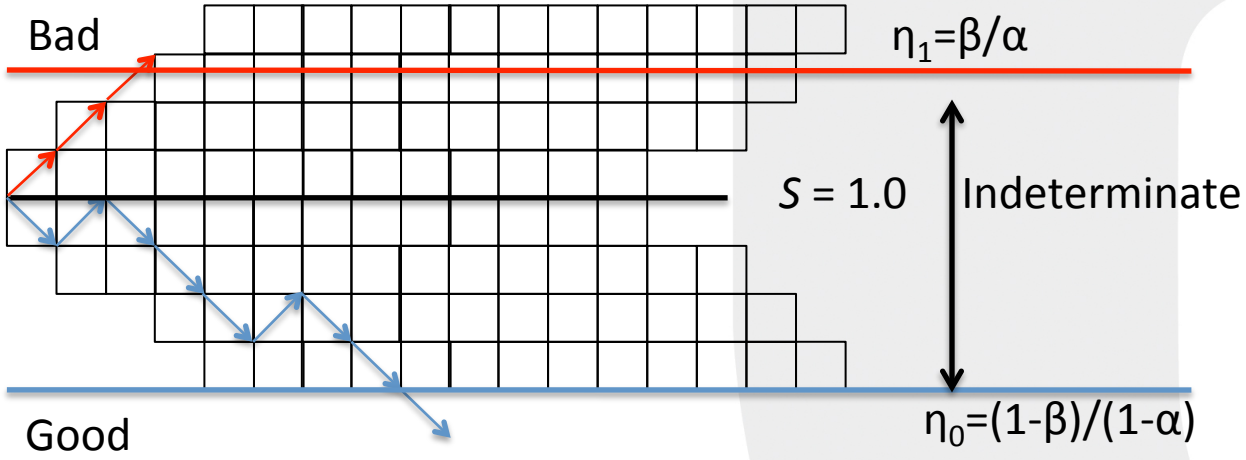
Scan detection
Threshold Random Walk (TRW)

REDJACK

- Assumptions
  - good guys connect, bad guys don't (mostly, for both)
  - bad guys behavior random, targets random (hah! / huh?)
- Model both behaviors
  - analyze connection attempt sequence
  - choose between good guy / bad guy hypothesis
- Need probabilities for models
  - θ_0 – good guy connects
  - θ_1 – bad guy connects
- Score *S* starts at 1.0 (indeterminate)
  - Successful connection multiplies score by θ_1 / θ_0
  - Failed connection multiplies score by $(1-\theta_1) / (1-\theta_0)$

REDJACK

TRW scoring and classification



- α is the desired false positive rate (0.01 often used)
- β is the desired detection rate (0.99 often used)
- $\eta_1=\beta/\alpha$ and $\eta_0=(1-\beta)/(1-\alpha)$ set the bad and good thresholds for the score *S*
- For a given set of parameters, possible to calculate min *all hit* counts for good and min *all miss* counts for bad

REDJACK

TRW and oracles

- In real time, hit/miss determination hard / expensive
  - Scan may be over before you can score
  - Use an *oracle* to predict connections
- An *oracle* tracks internal network services
  - Updated dynamically by outgoing traffic (or static)
    - For *ex post facto* analysis, oracle can be calculated from outbound traffic for an epoch, prior to inbound scan detection
    - Analysis of inbound traffic can be used to create an oracle if bi-directional traffic is not available
    - Both are effective with flow
  - Used to evaluate connection attempts
    - Works through temporary outages reducing false misses

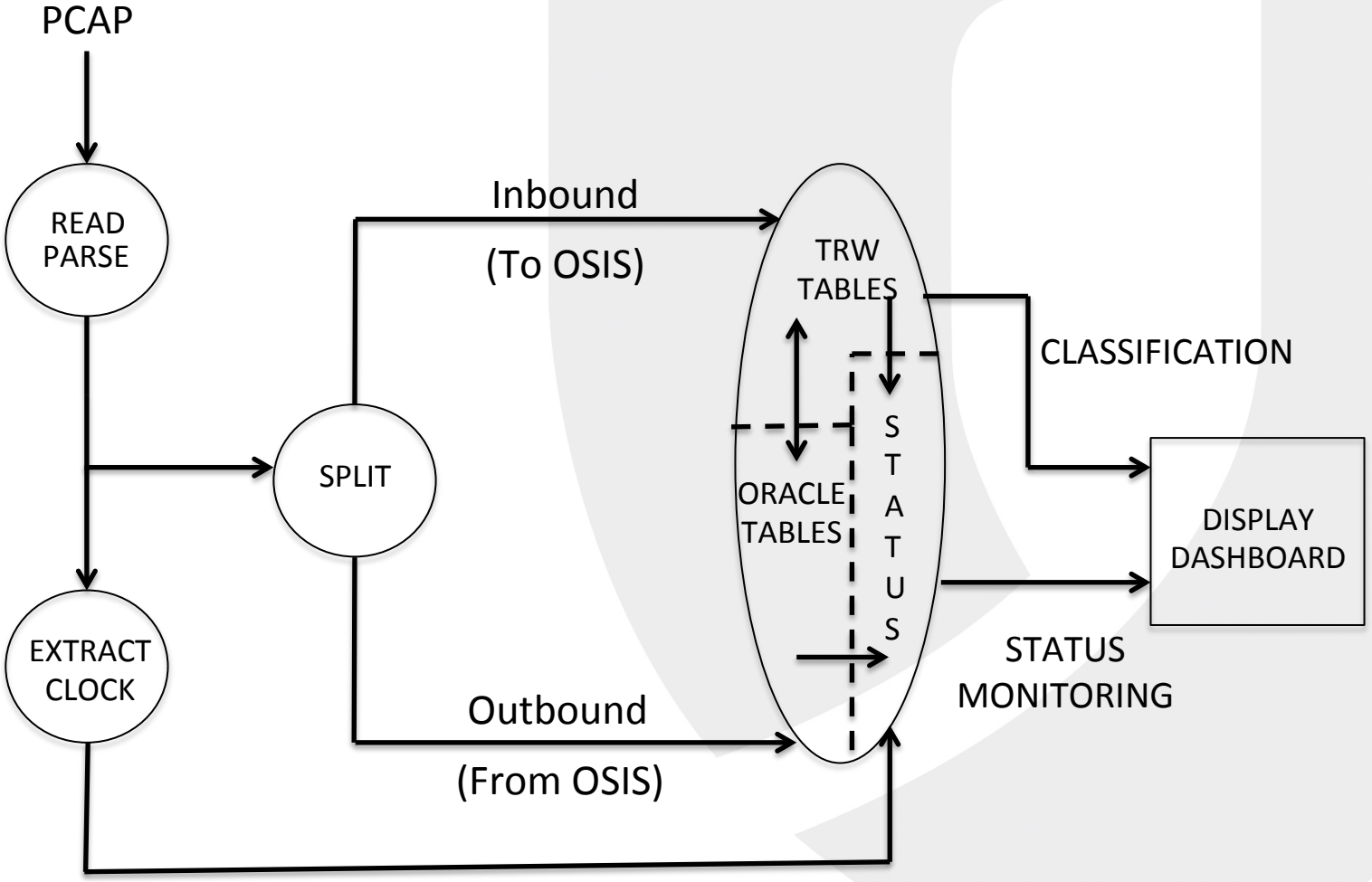
Multiple oracles for multiple scan modes

REDJACK

- TRW primarily used for TCP scans
  - Service oracle from sources that lead with SYN/ACK
  - Include service (responsive port) for precision
  - Can deal with things like passive mode FTP
- UDP oracle possible, as well
  - Can infer UDP service ports over time
- ICMP (ping) oracle trivial from ping response flows
- Adding ports improves precision
  - Detects vertical scans / mixed mode scans
  - Host only oracles for non-SYN TCP, etc. work too.
  - Computation of appropriate θ_1 is interesting
    - Randomness assumptions probably not correct

REDJACK

"Real time" TRW workflow for prerecorded pcap data



Flow is liberating (somewhat)

REDJACK

- Can separate oracle maintenance and scan analysis
  - Can pre-compute oracle for analysis epoch
  - In the absence of outbound data, can infer consensus oracle from multiple complete connections
  - With enough state, can detect very slow scans
    - Can even detect distributed scans with a bit of thought
- TRW computation simplified with oracles
  - Per host target lists most difficult part
    - Cuckoo sets for {source, target, service [, mode]}
    - Bloom filters to eliminate duplicates
    - Short, linear, list of targets (indeterminates with many targets are very rare – can be special cased)
    - Sorted data (as with `rwscan`)

REDJACK

The dirty truth about indeterminates

- TRW requires minimum target count to classify a source
  - Lots of sources have all hits to too few targets
    - Regular users of your primary web site (nothing else) OK
  - Lots of singletons (one target, hit or miss, never again)
    - Can probably forget about them (or aggregate off line)
  - Partial results from multiple locations / epochs compose
    - Could put partial results in a DBMS & periodically compose
    - Detect very slow scans this way composing on source
    - Detect distributed scans composing on service
      - Look for aggregates with good coverage
  - The epoch over which the initial analysis is done sets the detectability threshold.
    - Probably want a continuous process with table maintenance

REDJACK

Beyond detection – what now?

- TRW in real time can be an active defense
  - Block scanners before they learn about you
  - With flow, it is too late (even in the pipeline)
- *Ex post facto* detection can
  - Identify possibly compromised machines
    - Significant exchanges between scanner / scanee bad sign
    - Even small exchanges are a danger sign
      - Link target service to vulnerabilities and prioritize fixes
  - Characterize scan targets to see “what’s hot”
    - Fix vulnerable machines based on scan interest
    - Whether machine has been successfully scanned or not.
  - Trends over time – repeat scanners, modes, services

REDJACK

Comparison with rwscan (I)

- Flow data from 14 months of a /22 in Canada
  - oracle is set of all active hosts
  - Implementation using cubags
    - Span bag – all inbound sources w active interval as data
    - Hit bag – all src/dst pairs w dst in oracle (# flows as data)
    - Miss bag – ditto for dst not in oracle
    - Project dst off hit / miss bags and roll up to dst counts
    - Join projected bags, span bag to give src, hit / miss counts
    - Compute TRW score and classify.
  - We took 0:13, `rwscan` took 3:15 (`malloc` ???)
    - Found 8000 more scanners, 75,000 more benign than `rwscan`
    - 400,000 indeterminate, mostly too few flows, some single target with many repeats and lots of flows (5% of total flows)

Comparison with rwscan (II)

REDJACK

- IARPA (OSIS) data from PREDICT
  - Streaming pcap implementation for comparison
    - No timings: different platforms and demo stream slowed
    - Flow at 1 pkt/flow from rwptoflow
  - Separate oracles for Hosts, TCP, ICMP
  - Results for background data (scenario 5b5)

	rwscan	Host	TCP	ICMP
Scanner	14	16	26	5
Benign	-	329	39	0

  - Host includes 1 UDP, 1 ping + 14 detected by rwscan
  - TCP includes 12 vertical, 2 mixed + 10_1 + 2_2 by rwscan
  - Only 1 ICMP detected by rwscan. Others less than 32 flows (Minimum for missile component)

REDJACK

Observations

- Stopping analysis on classification only good in real time
  - Can take action (block, whitelist, etc.) in real time
  - In batch mode lose information on volume, targets
- Benign classifications are important
  - Useful to know nice as well as naughty
    - Detect behavior changes
- Multiple oracles very useful.
  - oracle data is a cheap dynamic system inventory
- Confounding scan detection with backscatter analysis, etc. is not useful.
  - This is not an “either / or” case

Future Directions

REDJACK

- Refinement of θ parameters
  - oracle allows tightening of θ_0 (closer to 1.0)
  - What is the actual target density (θ_1)
- State maintenance for continuous operation
  - Management / pruning of indeterminate hosts
- oracle maintenance
  - Might link removal to DNS ttl?
  - New services / transient ports
- Consequences of scanning
  - Compromised host detection
  - Prioritization of patching – CVE/NVD linkage
- Distributed scans might be tractable

REDJACK

Conclusions

- Scan detection is still important
- Most useful in real time, but *ex post facto* is useful
- Can be done with flow – has some advantages
- Predictive oracles better than traffic matching
  - A miss should be a hit sometimes
  - Multiple oracles for multiple scanning modes work
- Management of “indeterminates” is important
- Diagnosing “benigns” is important
- `rwscan` needs to be replaced
  - Scan database needs more information
  - Need to feed operationally useful actions

REDJACK

Questions / Discussion

John McHugh
Senior Principal
RedJack, LLC
john.mchugh@redjack.com

I'll be around for the rest of the meeting.
Come talk to me.

REDJACK

Questions?

Situational Awareness Metrics from Flow and Other Data Sources

Soumyo D. Moitra
SEI CERT NetSA

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE 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.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

Introduction

Need a more flexible set of metrics for
network situational awareness

- Aggregate (over large IP sets)
- Composite (multiple measures or counts)
- Can detect changes in traffic patterns
- Amenable to visualization
- Fast and scalable (simple algorithms)

Overview

Propose some new metrics for SA

Uses Flow Data

Some require additional data:

- Information on Assets
- Organizational Level Data
- Elicited Data
- Various Lists of Sites/Hosts/Domains

- DNS
- Topology

Proposed metrics

Threats - Risks - Impacts

A) Mainly flow data and results from SIMS

N(attack category) – N(method of operation) – N(system or host)

Estimate (N): <TP> + <FN> | exercises & pen tests

B) Flow and other data

Match attack sources with malicious domain lists

- intersection of the IP sets

Implementation levels & Compliance levels

- priority of patches or tasking orders
- criticality of hosts

Probabilities of success * Expected damage | Attack category

Flow-based Metrics: Threats

N(i) | {I} = distribution of attacks by category i in set {I}
= attack scenario

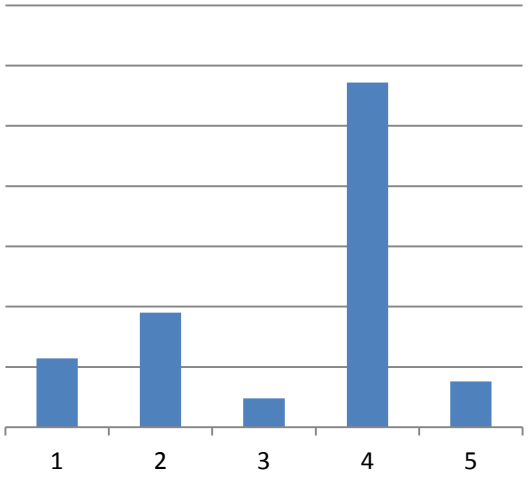
[w(i)*N(i) | $\underline{w}$] = seriousness-weighted attack distribution
= attack intensity

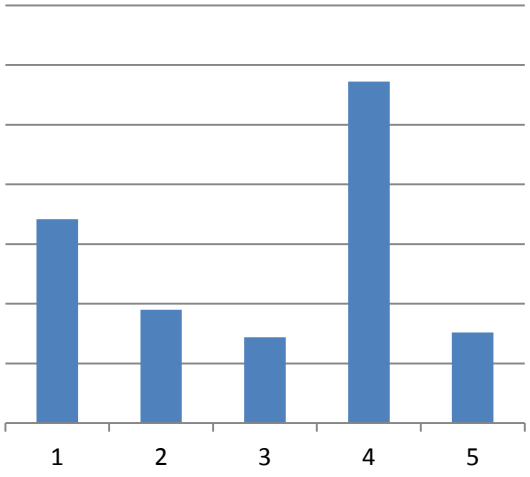
Σ_i[w(i)*N(i)] = Overall seriousness > Trends

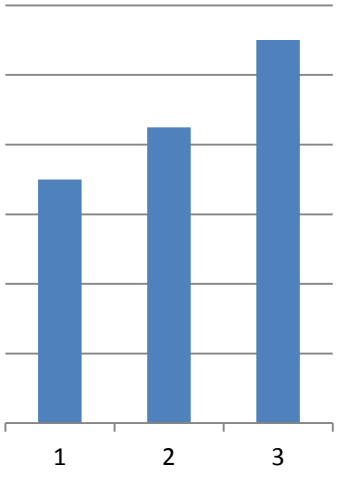
$[\Sigma_i w(i) = 1]$

Soumyo D. Moitra
NetSA/CERT SEI

Illustrative Example: Numbers, Intensity, Trends







Soumyo D. Moitra
NetSA/CERT SEI

Flow-based Metrics: Risks

N(m) = Distribution of attacks by “method of operation”

[s(m)*N(m)|s] = Severity-based risk scenario

Σ_m[s(m)*N(m)] = Overall severity metric

$[\Sigma_m s(m) = 1]$

Flow-based Metrics - Impacts

N(h) = Distribution of attacks by system/host “h”

[v(h)*N(h)|v] = Value-weighted impact of attacks

Σ_h[v(h)*N(h)] = Overall value of network assets
that are being attacked by their attack rate

Other Data Needed

{Attack Categories} and {Relative Seriousness}

{Taxonomy for MOs} and {Potential Severity}

{Classification of Network Assets – value & criticality}

{True Positives | Alerts & Verification} and {False Negatives}
<- Exercises and Testing

{Lists of Malicious Domains/Ips} [Some exist]

{Status of Assets w.r.t. Compliance: Yes/No |patch or TO}

{Success rate of attacks by category | recent reports}

{Expected damage from successful attacks}

Soumyo D. Moitra
NetSA/CERT SEI

Estimation of metrics based on non-flow data

Maliciousness of Attacker Set:
{A} ~ attacker set
{M} ~ lists of known malicious hosts
{A} ∩ {M} = Degree of Attribution by Maliciousness

Risk ≡ Non-compliance (+ other factors) >> Compliance level
$\Sigma_h J(h,p)*c(h) / \Sigma_h Y(h,p)*c(h) = I(p)$ = Implementation level of patch p
$J \supset Y$
$\Sigma_p u(p)*I(p)$ = Compliance level with respect to patching
Compliance by criticality & urgency

Impact:
Likelihood * Consequence
= π * D = {probability of successful attack * expected damage}

{π(k) * D(k); by level of damage k}

Benefits for Situational Awareness

New metrics to supplement current measures

Additional aspects of SA

Identification of important data to be collected

Fast estimation procedures

Can track changes over time

Summary and Conclusions

Summary

Set of SA metrics: ***Threats-Risk-Impact***
Properties and interpretation of the metrics
Flow data and additional data (as identified)
Benefits from applying these metrics

Key Challenges

A processing and analysis layer between queries & reporting
Data availability
Problems with the numbers (NATs, Proxies, inconsistencies, etc.)

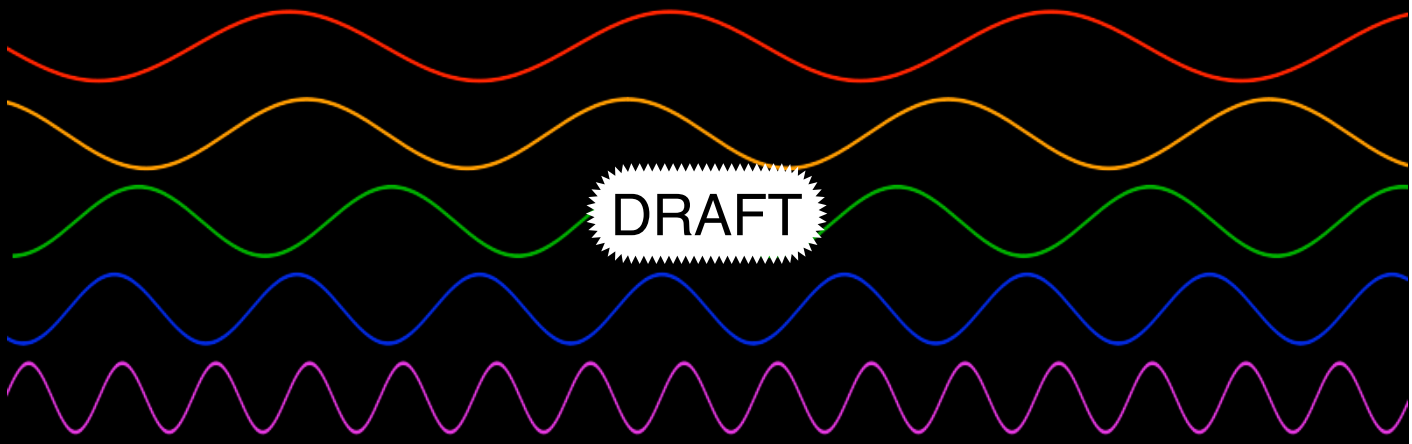
Future Work in Brief

Develop, validate & interpret these metrics
Collect the needed data systematically
Include the intermediate analytics capabilities

THANK YOU!

smoitra@cert.org

Statistical analysis of flow data using Python and Redis



FLOCON 2013

Kevin Noble
Terraplex@gmail.com

Overview

- 1. Beacon description
- 2. Beacons as used by attackers
- 3. Considerations for beacon classification
  - a. periodicity in time series analysis
    - i. Considerations to evaluate periodicity
- 4. Visualize beacons
  - a. Factors of classification useful to detect beacons
- 5. Beacon Bits, an analytical tool set and workflow to detect beacons
  - a. Demo
  - b. Extracting data from flows
  - c. Storing timing data
  - d. Statistical analysis and evaluation of beacon properties
- 6. Result
- 7. Code / Discussion / Q&A

http://www.mcafee.com/us/resources/white-papers/wp-global-energy-cyberattacks-night-dragon.pdf

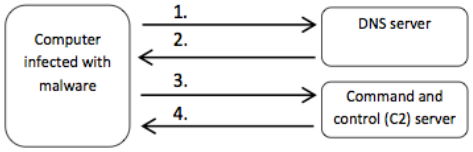
Network Communications

Network communications are relatively easy to detect because the malware uses a unique host beacon and server response protocol. Each communication packet between the compromised host and the C&C server is signed with a plain text signature of "hW$." (or "\x68\x57\x24\x13") at the byte offset 0x42 within the TCP packet.

The backdoor begins its beacon at approximately five-second intervals with an initial packet that may be detected with the pattern: "\x01\x50[\x00-\xff]+\x68\x57\x24\x13."

http://www.commandfive.com/papers/C5_APT_C2InTheFifthDomain.pdf

The malware used in the attack was programmed to communicate with several 'callback' domains. The malware located its C2 server(s) by resolving these domains into IP addresses using the ubiquitous Domain Name System (DNS) [3] protocol. These communications are depicted in Figure 1.



1. Using the Domain Name System (DNS) protocol, the computer asks a DNS server for directions to the callback domain.
2. The DNS server advises that the callback domain is located at IP address x.x.x.x.
3. The malware communicates with the C2 server located at IP address x.x.x.x to obtain C2 instructions and/or to send a response.
4. The C2 server provides additional C2 instructions to the malware.

After sending the basic beacon, the compromised computers waited for a response from the server, then closed the connection when they had not received a response from the server within five seconds.

Both of the compromised computers reattempted the communications approximately every eight seconds. On some days the high frequency of the beacon activity resulted in over 10000 connection attempts per victim in a 24 hour period.

Beacons

1. Beacons manifest as repetitious communication attempts in the form of packets
   - a. Most beacons are not malicious
   - b. Malicious beacons are sourced from infected host where the malware repeatedly attempts remote connectivity
   - c. Beacon events are discernible
2. Detection
   - a. The more frequent a beacon, the easier to detect
   - b. Beacons that are consistent in time series are easier to detect
   - c. Beacons events lend themselves to time series analysis

Timing is a signature

PROTOCOL	TYPICAL BEACON INTERVAL* (SECONDS)
LURK	26
X-Shell C601	36
Update?	1 to 13, 12±3, 16, 104±3 or 200 ±15
Murcy	11
Oscar	12±2, 13, 15, 16, (55 or 155±5), (7.5, 8.5 or 15) , (45, 55, 106)
BB	8
DB	4 to 92
Qdigit	60

* Commas indicate that the interval changed between victims. Brackets indicate that a variety of intervals were observed from a single computer.

TABLE 7: INTERVAL BETWEEN COMMUNICATIONS

http://www.commandfive.com/papers/C5_APT_C2InTheFifthDomain.pdf

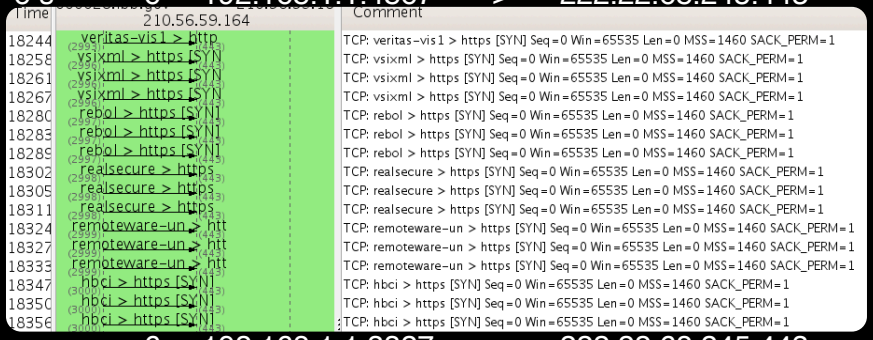
Sample Beacon as viewed in flow for network and timing properties

Present all the characteristics and properties for known beacons

Avoid payload analysis (except perhaps size)

beacon/testset$ ra -nnr beacon_test_extract.arg - host 222.22.68.245

StartTime	Flgs	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts	TotBytes	State
13:00:58.783986	e s	6	192.168.1.1.3719		->	222.22.68.245.443		2	124	REQ
13:31:52.667327	e s	6	192.168.1.1.3208		->	222.22.68.245.443		2	124	REQ
14:01:53.659479	e s	6	192.168.1.1.2665		->	222.22.68.245.443		2	124	REQ
14:32:00.062273	e s	6	192.168.1.1.2152		->	222.22.68.245.443		2	124	REQ
15:02:55.611042	e s	6	192.168.1.1.1962		->	222.22.68.245.443		2	124	REQ
15:33:52.663009	e s	6	192.168.1.1.1524		->	222.22.68.245.443		2	124	REQ
16:03:52.602414	e s	6	192.168.1.1.4867		->	222.22.68.245.443		2	124	REQ
16:33:57.090316	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
17:04:52.558100	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
17:34:59.598407	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
18:05:56.669750	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
18:36:53.968150	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
19:06:56.229070	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
19:37:53.975195	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
20:08:53.685264	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
20:38:54.173905	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ
21:10:09.140943	e s	6	192.168.1.1.3327		->	222.22.68.245.443		2	124	REQ
21:40:52.834383	e s	6	192.168.1.1.2808		->	222.22.68.245.443		2	124	REQ
22:10:57.850103	e s	6	192.168.1.1.2231		->	222.22.68.245.443		2	124	REQ
22:41:55.148182	e s	6	192.168.1.1.1718		->	222.22.68.245.443		2	124	REQ
23:12:58.582524	e s	6	192.168.1.1.1244		->	222.22.68.245.443		2	124	REQ
23:43:52.478378	e s	6	192.168.1.1.4999		->	222.22.68.245.443		2	124	REQ



GOAL: Surface malicious beacons for inspection by examining Network traffic

Inspecting traffic flows for beacons

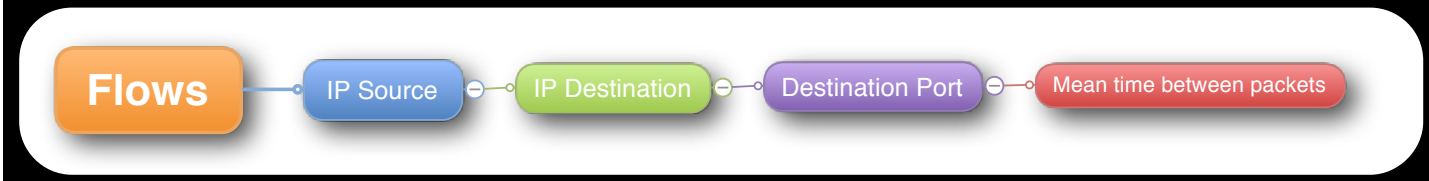
Flow based tools have a limited facility to detect beacons alone.

Flow tools are ideal for the collection and verification of beacons.

Flow based tools do provide counts and summaries and quantizing (bins) in some cases.

Quantize time to seconds (sub-seconds complicate the details) appears to be useful.

Timing is the key to detection followed by verification by inspecting the host.

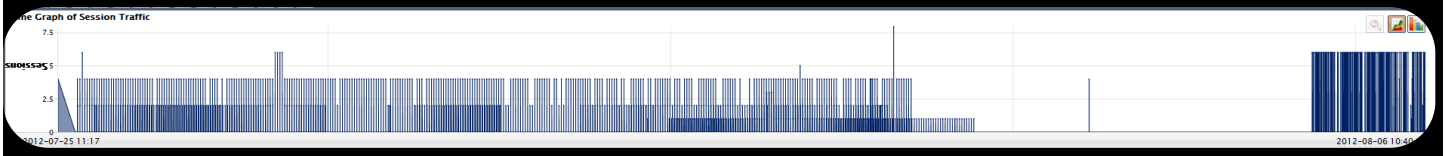


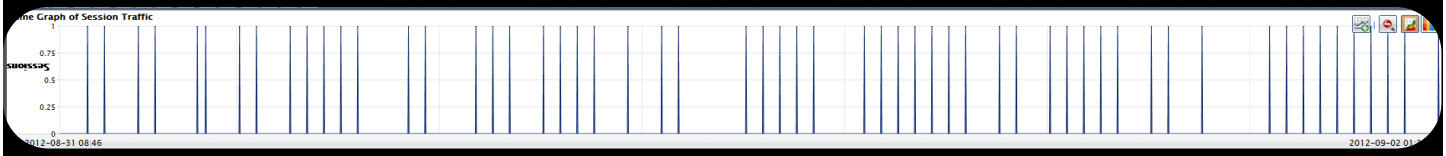
Beacon p0rn

Visual timing as a graph

Produces an instant visual representation of a beacon.

Graphing does not scale to allow analyst to inspect everything.





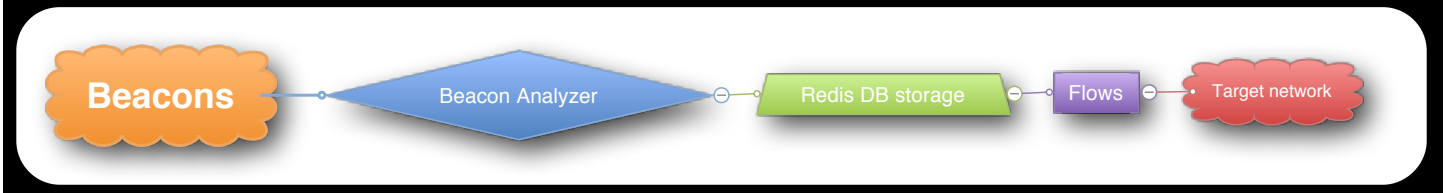
[1854, 1801, 1807, 1855, 1857, 1800, 1805, 1855, 1807, 1857, 1857, 1803, 1857, 1860, 1801, 1843, 1805, 1858, 1863, 1854, 1801, 1863, 1859, 1857, 1801, 1859, 1802, 1858, 1802, 1802, 1856, 1800, 1800, 1800, 1860, 1804, 1858, 1863, 1859, 1857, 1804, 1802, 1854, 1804, 1856, 1802, 1859, 1812, 1847, 1808, 1853, 1867, 1851, 1800, 1800, 1806, 1801, 1854, 1801, 1800, 1865, 1861, 1861, 1850, 1800, 1800, 1801, 1864, 1858, 1857, 1803, 1804, 1853, 1801, 1864, 1859, 1802, 1859, 1858, 1857, 1803, 1808, 1849, 1804, 1857, 1800, 1808, 1853, 1863, 1861, 1854, 1802, 1858, 1865, 1857, 1865, 1855, 1802, 1856, 1800, 1803, 1862, 1859, 1858, 1801, 1800, 1859, 1806, 1853, 1859, 1801, 1804, 1801, 1855, 1812, 1803, 1844, 1800, 1802, 1858]

Graphing every session does not scale

Beacon detection

Beacon Bits

- 1. Parse from FLOW
  - a. IP Source
  - b. IP Dest
  - c. Port Dest
  - d. Time (from Source)
- 2. DataStore
  - a. Native Python
  - b. Redis
- 3. Analysis
  - a. Python
- 4. BEACONS



```
IP source   1.1.1.1
IP dest     210.215.10.254  "NEXONASIAPACIFIC"
dst port    443
pair_count  8432
mean        121
Standard Deviation: 0.026849474628 169643.0
compensated_variance: 2542
online_variance: 20548
online_variance_n: 20546
web_std_dev (0.002493930934161027, 0.22931978029843433)
seconds     1020272     minutes     17004 hours 283
    days    11
src_count   10809
dst_count   8432
traffic with source and dest:
'SET:1.1.1.1:210.215.10.254:443:2012810'
'SET:1.1.1.1:210.215.10.254:443:2012811'
'SET:1.1.1.1:210.215.10.254:443:2012812'
'SET:1.1.1.1:210.215.10.254:443:2012813'
'SET:1.1.1.1:210.215.10.254:443:2012814'
'SET:1.1.1.1:210.215.10.254:443:2012815'
'SET:1.1.1.1:210.215.10.254:443:2012816'
'SET:1.1.1.1:210.215.10.254:443:2012817'
'SET:1.1.1.1:210.215.10.254:443:2012818'
'SET:1.1.1.1:210.215.10.254:443:2012819'
'SET:1.1.1.1:210.215.10.254:443:2012820'
'SET:1.1.1.1:210.215.10.254:443:2012821'
'SET:1.1.1.1:210.215.10.254:443:2012822'
'SET:1.1.1.1:210.215.10.254:443:multi']
[21, 223, 21, 223, 21, 222, 21, 223, 21, 223, 21, 223, 21, 222, 21, ….]
```

Beacon Classification and expression

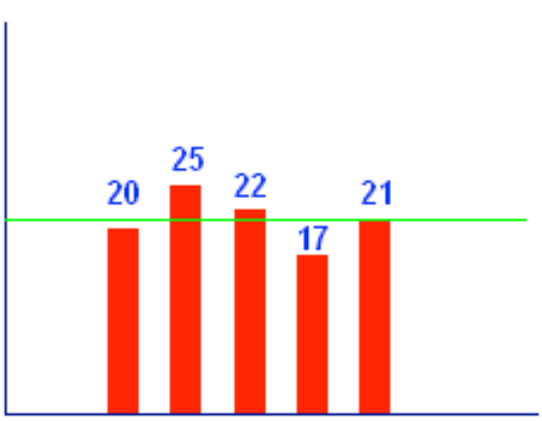
Execution condition	Frequency	Interval / Mean	Packet Proto	Packet Dest	Port	Payload	Payload Size
Continuous	Consistent	Static	Single	Single	Single	Consistent	Static
conditional	Transient	Dynamic	Multiple	Multiple	Multiple	Transient	Dynamic
transient						none	

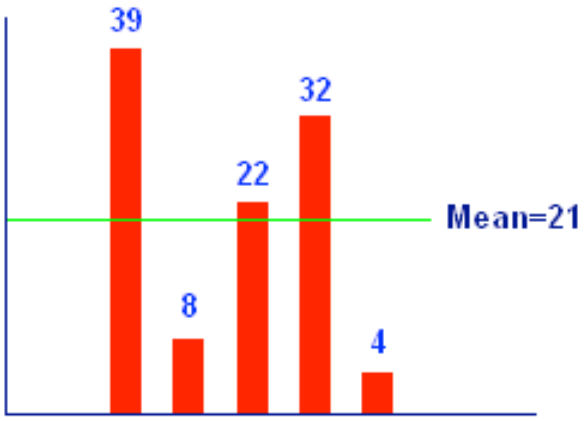
Beacon expression as a combination of conditions

Continuous and consistent TCP packets at 300 second intervals
TCP packet over a single port 80 every 900 seconds continuously
7 packets, 5 minutes apart, every 3 days using TCP or UDP to one of of 5 host over one of these 3 ports, with the following payload
1 TCP packet, every 30 day to one of 30 possible host

Malicious Beacons top characteristics used in the analysis process

- 1. Unconnected beacons
  - a. Low Varience
  - b. Low Standard Deviation
  - c. Limited number of host attempting to Connect
  - d. At least 3 packets
  - e. At least 15 minutes of 'total' time in the analysis
- 2. Connected beacons
  - a. Similar as unconnected
  - b. Payload is a factor
    - i. Strings / offsets / atomic





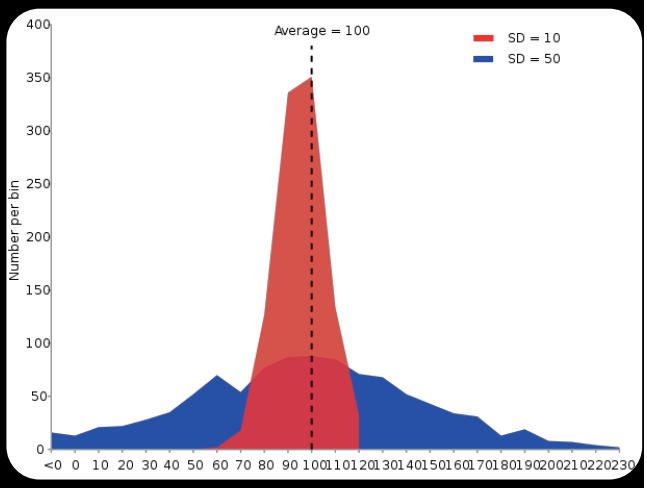
Histograms

- 1. Limited usefulness if used exclusively
- 2. Histograms value factors:
  - a. Large sample population
  - b. Combined with varience
  - c. Combined with static classifications (previous slides)
- 3. Dropped from analysis based on performance of other factors

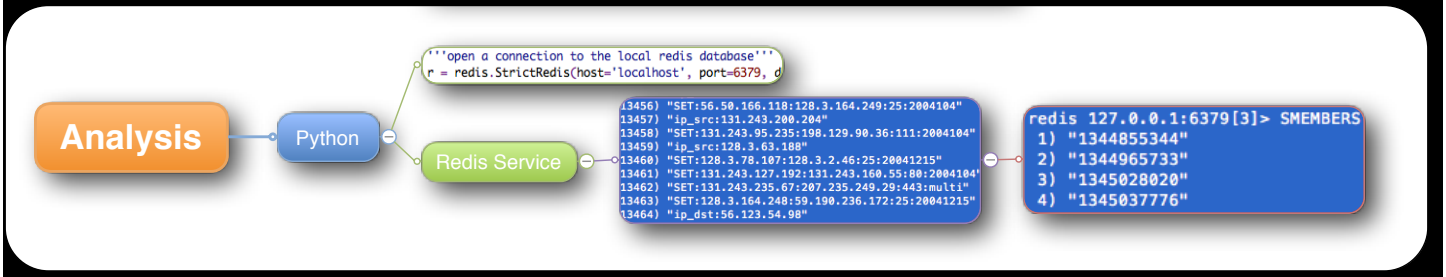
```
$ rahisto -H dur 120 -r htest.arg
N =         122
mean =      2.927606
stddev =    0.382476
max =       3.113874
min = 0
median
95% = 3
mode = 0.000000
```

Flow conversion to mysql
rasqltimeindex -r argus.file -w mysql://user@host/db

Class	Interval	Freq	Rel.Freq	Cum.Freq
108	2.782000e+00	0	0.0000%	1.6393%
109	2.808000e+00	1	0.8197%	2.4590%
110	2.834000e+00	2	1.6393%	4.0984%
111	2.860000e+00	3	2.4590%	6.5574%
112	2.886000e+00	8	6.5574%	13.1148%
113	2.912000e+00	16	13.1148%	26.2295%
114	2.938000e+00	25	20.4918%	46.7213%
115	2.964000e+00	19	15.5738%	62.2951%
116	2.990000e+00	20	16.3934%	78.6885%
117	3.016000e+00	10	8.1967%	86.8852%
118	3.042000e+00	5	4.0984%	90.9836%
119	3.068000e+00	8	6.5574%	97.5410%
120	3.094000e+00	3	2.4590%	100.0000%



Enumerate over keys
Should be able to move through the millions of keys quickly
Evaluate traffic based on timing properties in a statistical sense
Some assumption include host might be up during working hours
No more then 4 host would be infected




```
sets_sub = each.split(':')
set_src_ip = sets_sub[1]
set_dst_ip = sets_sub[2]
set_dst_port = sets_sub[3]
set_date = sets_sub[4]
src_count = r.get('ip_src:'+set_src_ip)
dst_count = r.get('ip_dst:'+set_dst_ip)
```

Variance

1. http://en.wikipedia.org/wiki/Algorithms_for_calculating_variance
   - a. **Algorithms for calculating variance** play a major role in statistical computing. A key problem in the design of good algorithms for this problem is that formulas for the variance may involve sums of squares, which can lead to numerical instability as well as to arithmetic overflow when dealing with large values.
   - b. Several Algorithms tested, settled on using three:
     - i. Compensated Variance
     - ii. Online variance
     - iii. Kurtosis

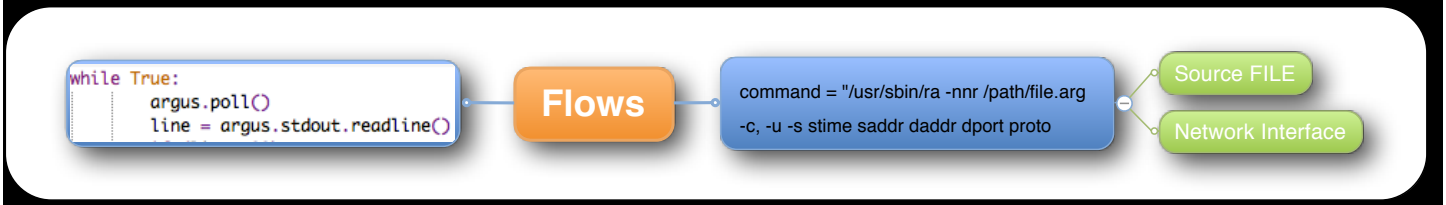
Standard Deviation

- 1. Little ‘dispersion’ for each set
- 2. Minimum population distance from the mean
- 3. Using a MODIFIED version of Standard Deviation that would be considered a WEIGHT
  - a. Tolerance increase with frequency (reverting to normal standard deviation for final release)

SOURCE IP	DEST IP	DEST PORT	DATE	STDDEV
100.0.5.230	1.0.20.5	8888	2012913	0.045732737
100.0.5.230	1.0.20.5	8888	2012914	0.044662676
100.0.5.230	1.0.20.5	8888	2012915	0.04343173
100.0.5.230	1.0.20.5	8888	2012916	0.042813404
100.0.5.230	1.0.20.5	8888	multi	0.019851071

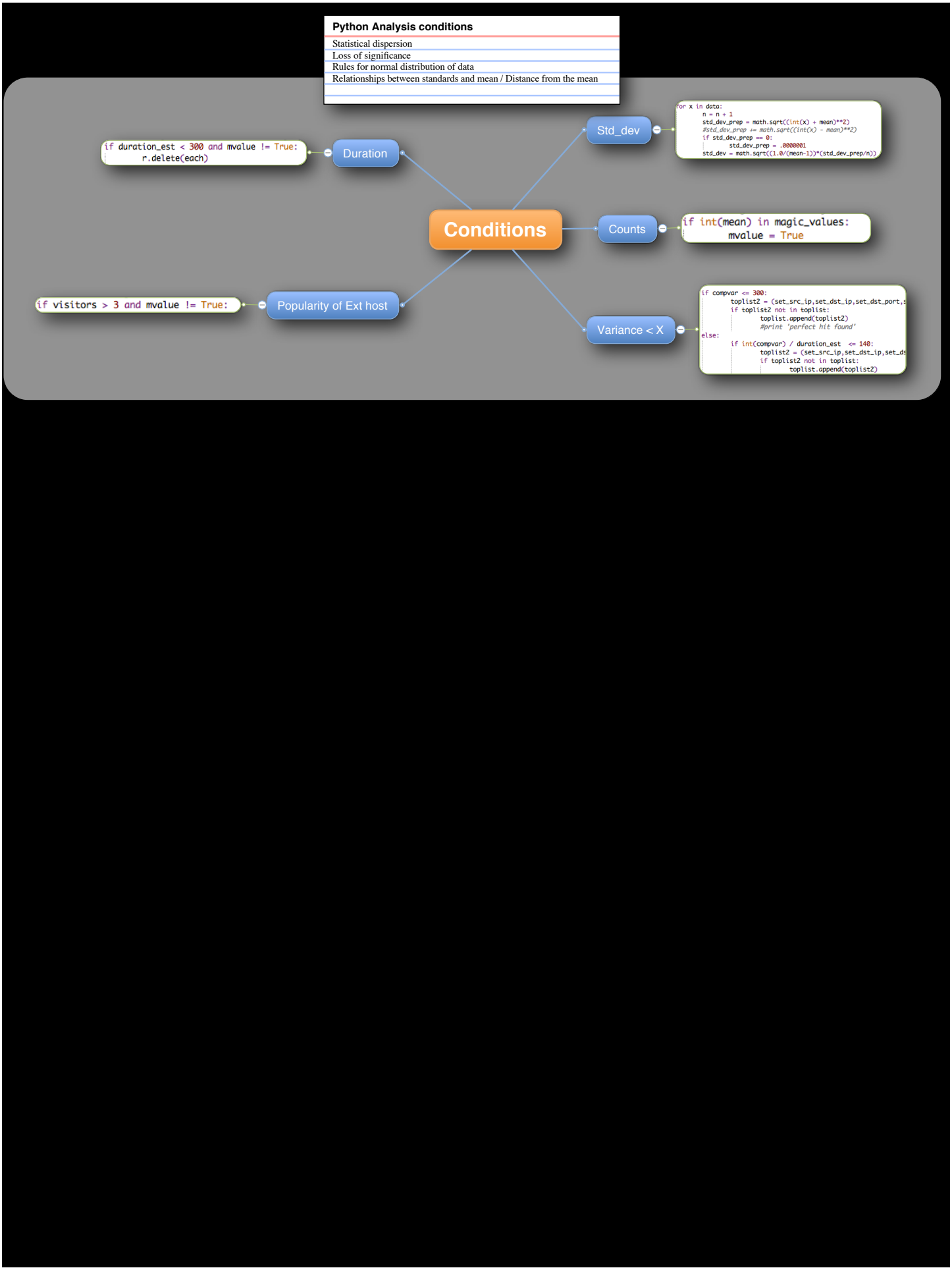
Extracting from Flows

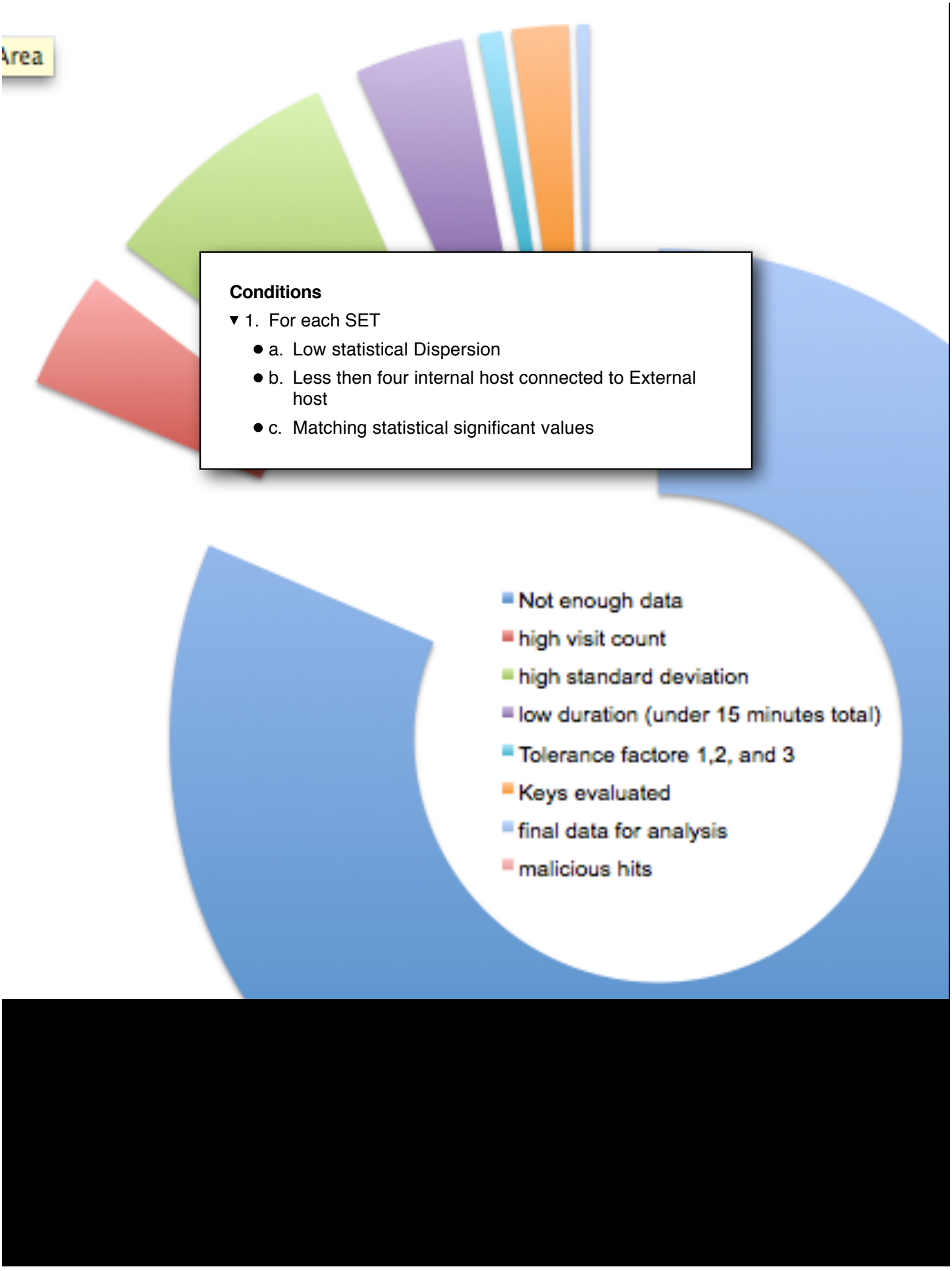
Can we tabulate timing for traffic as a means to detect beacons?
TCP SYN
Isolated to traffic sources from the network we seek to defend
Traffic destined to external network (avoid internal to internal packets)
Exclusion of trusted and authorized host and networks (if possible)
Limited totTrack timing properties



Polling
Using Python to compile a dataset is a process of conversion from binary parsed to text, formed into sets.
The largest sample set took 54 minutes to consume and held traffic for 16 days.
Python handles the sets fairly well but does not facilitate continuous analysis.

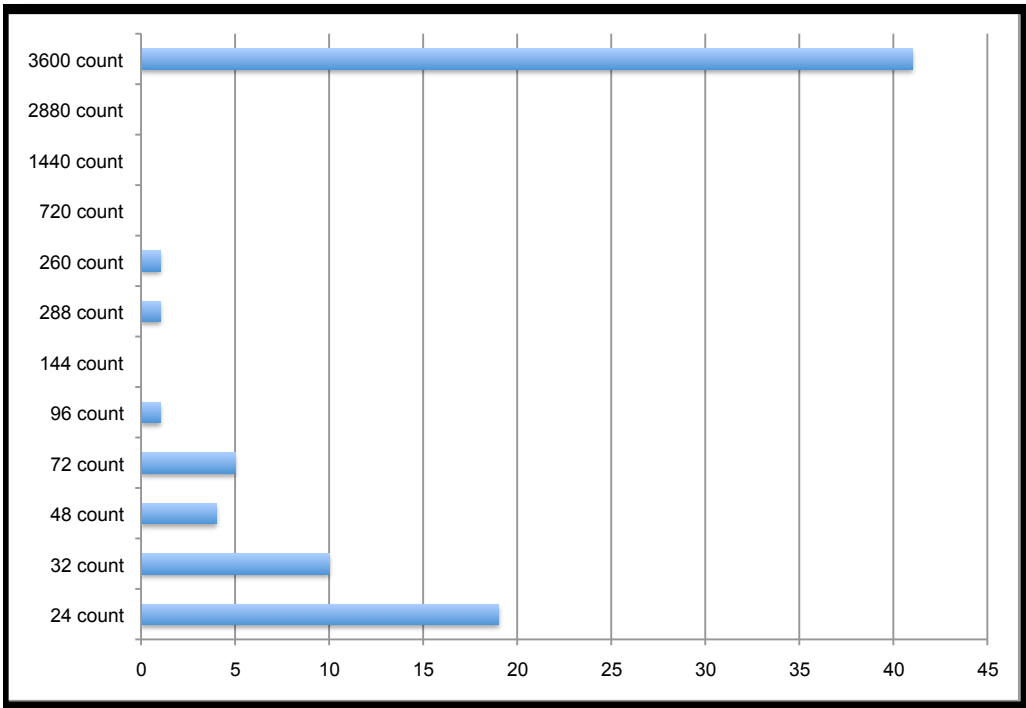
Analysis considerations




Area
Conditions
1. For each SET
a. Low statistical Dispersion
b. Less then four internal host connected to External host
c. Matching statistical significant values
Not enough data
high visit count
high standard deviation
low duration (under 15 minutes total)
Tolerance factore 1,2, and 3
Keys evaluated
final data for analysis
malicious hits

Divisible by 60 seconds?
Beacons generally resolve to set intervals in minutes
Connected sessions also maintain a connected state set in minutes
Most basic Remote Administration Tools
False positive are frequent
Evaluating Interval count alone still produces a useful set
Excluding trusted networks is useful

seconds in a day	Interval in minutes	Count
86400	0.5	2880
86400	1	1440
86400	2	720
86400	4	360
86400	5	288
86400	10	144
86400	15	96
86400	20	72
86400	30	48
86400	45	32
86400	60	24



Interval	Count
0.5	30
1	60
2	120
4	240
5	300
10	600
15	900
20	1200
30	1800
45	2700
60	**3600**
40	2400
30	1800
20	1200

Introduction to Redis

Redis is an open source, advanced **key-value store**. It is often referred to as a **data structure server** since keys can contain strings, hashes, lists, sets and sorted sets.

You can run **atomic operations** on these types, like appending to a string; incrementing the value in a hash; pushing to a list; computing set intersection, union and difference; or getting the member with highest ranking in a sorted set.

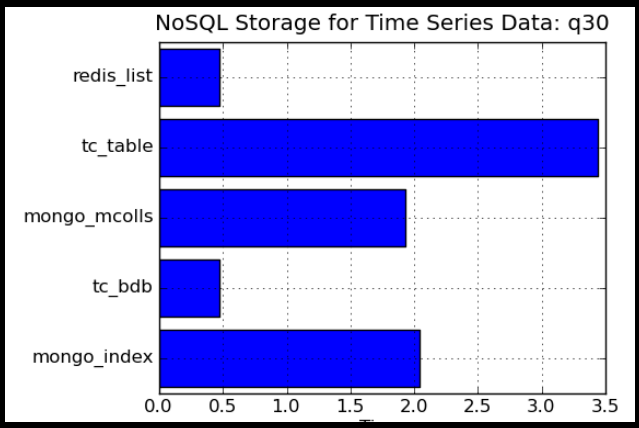
In order to achieve its outstanding performance, Redis works with an **in-memory dataset**. Depending on your use case, you can persist it either by dumping the dataset to disk every once in a while, or by appending each command to a log.

Redis also supports trivial-to-setup master-slave replication, with very fast non-blocking first synchronization, auto-reconnection on net split and so forth.

Other features include a simple check-and-set mechanism, pub/sub and configuration settings to make Redis behave like a cache.

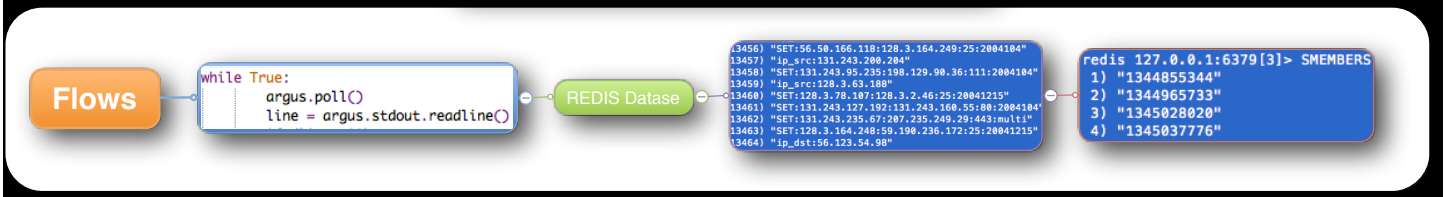
You can use Redis from most programming languages out there.

Redis is written in **ANSI C** and works in most POSIX systems like Linux, *BSD, OS X without external dependencies. Linux and OSX are the two operating systems where Redis is developed and more tested, and we **recommend using Linux for deploying**. Redis may work in Solaris-derived systems like SmartOS, but the support is *best effort*. There is no official support for Windows builds, although you may have some options.



Source: https://github.com/yinhm/nosql-tsd-benchmark

Collection
Tracking SETS with timing information
Tracking Source IP activity by count
Tracking Destination activity by count
Redis manages duplicates
Redis can handle the size
Memory is ideal for the transaction rate and the type of data being managed




```
beacon/testset$ ra -nnr beacon_test_extract.arg  - host 222.22.68.245
      StartTime  Flgs  Proto       SrcAddr  Sport  Dir       DstAddr  Dport  TotPkts  TotBytes  State
 13:00:58.783986  e s      6  192.168.1.1.3719   ->  222.22.68.245.443      2       124  REQ
 13:31:52.667327  e s      6  192.168.1.1.3208   ->  222.22.68.245.443      2       124  REQ
 14:01:53.659479  e s      6  192.168.1.1.2665   ->  222.22.68.245.443      2       124  REQ
 14:32:00.062273  e s      6  192.168.1.1.2152   ->  222.22.68.245.443      2       124  REQ
 15:02:55.611042  e s      6  192.168.1.1.1962   ->  222.22.68.245.443      2       124  REQ
```

Simplistic data schema

1. For Each IP Source, IP Dest, Dest Port, Date
    - a. Unix Time (String)
2. Counts
    - a. Increment counter
        - i. Source
        - ii. Destination
3. Date and Multiple
    - a. Supports differential analytical output
    - b. Statistical significance might be represented over multiple days
    - c. Statistical significance might be represented on a single day
4. Expiring keys
    - a. Necessary for production
5. White List
    - a. Useful for production
    - b. Requires care and feeding

Demonstration

- 1. start redis server and client
- 2. Populate redis database from flow file
- 3. collect timing data form flow file
- 4. launch analyzer
  - a. show redis db post analyzer
- 5. launch graph view

Significance

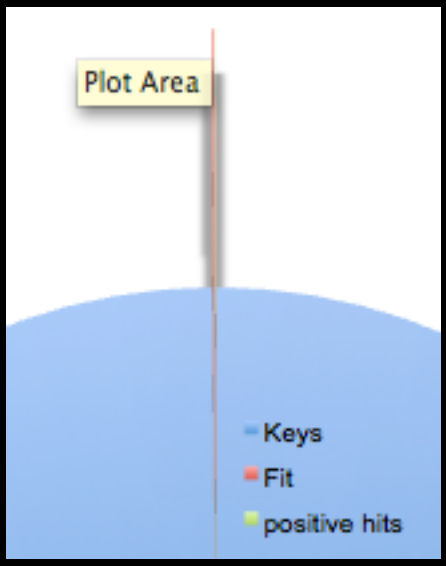
Parsing through 3 days of traffic yields beacons.

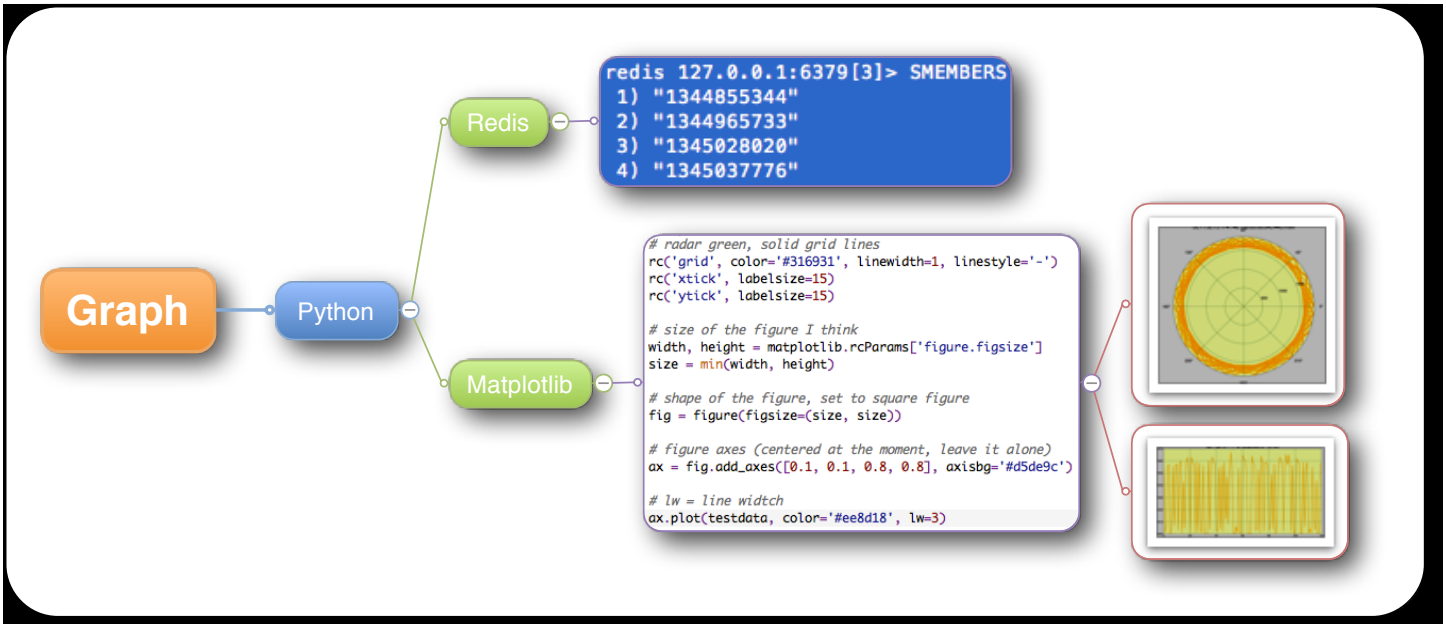
The number of beacons depends on the test conditions

The most statistically significant data included malicious beacons

Pulling the most significant results with flows and full packet capture is useful

Host inspection is the best verification of results




Graph
Python
Redis
redis 127.0.0.1:6379[3]> SMEMBERS
1) "1344855344"
2) "1344965733"
3) "1345028020"
4) "1345037776"
Matplotlib
# radar green, solid grid lines
rc('grid', color='#316931', linewidth=1, linestyle='-')
rc('xtick', labelsize=15)
rc('ytick', labelsize=15)
# size of the figure I think
width, height = matplotlib.rcParams['figure.figsize']
size = min(width, height)
# shape of the figure, set to square figure
fig = figure(figsize=(size, size))
# figure axes (centered at the moment, leave it alone)
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8], axisbg='#d5de9c')
# lw = line width
ax.plot(testdata, color='#ee8d18', lw=3)

Graph / Plot (text view)

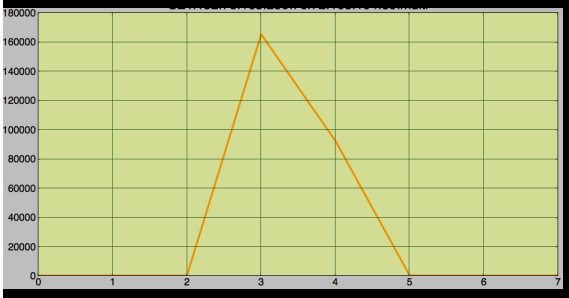
1. Specific results can be examined in detail
2. The timing data can be put into an array for a graphical display

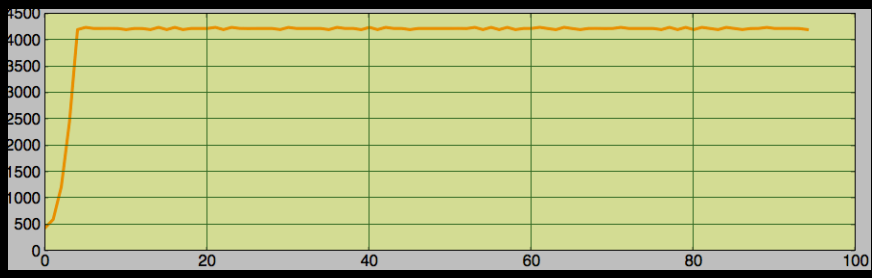
```
seconds between query: 21 Standard Deviation: 0.0229724128309 425812.0
IP source       100.0.9.95
IP dest         1.0.9.25        "APNIC Debogon Project"
dst port        80
pair_count      1896
mean            224
seconds         424704  minutes         7078 hours 117  days    4
src_count       1899
dst_count       3792
traffic with source and dest ['SET:100.0.9.95:1.0.9.25:80:2012912', 'SET:100.0.9.95:1.0.9.25:80:2012913', 'SET:100.0.9.95:1.0.9.25:80:2012914',
'SET:100.0.9.95:1.0.9.25:80:2012915', 'SET:100.0.9.95:1.0.9.25:80:2012916', 'SET:100.0.9.95:1.0.9.25:80:2012917', 'SET:100.0.9.95:1.0.9.25:80:multi']

compensated_variance: 31178
online_variance: 156444
online_variance_n: 156362
web_std_dev (0.01108764481808884, 0.48254481793081905)
nslookup details: Server:               208.67.222.222
Address:        208.67.222.222#53
Finished

** server can't find 25.9.0.1.in-addr.arpa.: NXDOMAIN



[18, 21, 21, 837, 21, 22, 21, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21,
837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 22, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21,
21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21,
21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 20, 21, 21, 838, 21, 21, 21, 837,
21, 21, 21, 837, 20, 22, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 20, 21, 21, 838, 21, 21, 21, 837, 20, 22, 21, 836, 22, 21, 21,
836, 21, 21, 22, 836, 22, 21, 21, 836, 22, 21, 21, 836, 21, 21, 21, 837, 22, 21, 21, 836, 21, 21, 21, 837, 21, 22, 21, 836, 21, 22, 21, 836, 21, 22,
21, 836, 21, 22, 21, 836, 21, 24, 18, 837, 21, 22, 21, 837, 20, 21, 22, 836, 21, 21, 21, 837, 22, 21, 21, 836, 21, 21, 21, 837, 21, 22, 20, 837, 21,
21, 21, 837, 21, 21, 22, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 22, 21, 836, 21, 22, 21, 836, 21, 22, 21, 836, 21, 21, 21, 837, 21, 21, 21, 837,
21, 21, 21, 837, 21, 21, 22, 836, 21, 22, 21, 836, 21, 21, 21, 837, 21, 21, 22, 836, 21, 22, 21, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21,
837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 22, 21, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 22,
21, 836, 21, 21, 22, 836, 21, 21, 22, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 22, 836, 21, 21, 21, 837, 21,
21, 22, 836, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 24, 834, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837,
21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 838,
21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21, 837, 21, 22, 21, 836, 21, 21, 21, 837, 21, 21, 21, 838, 21, 21, 21, 837, 21, 21, 21, 837, 21, 21, 21,
```

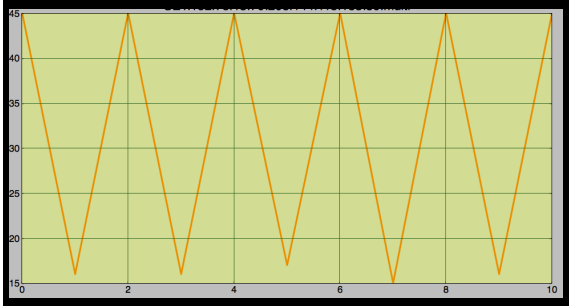
Plot Text OUTPUT example

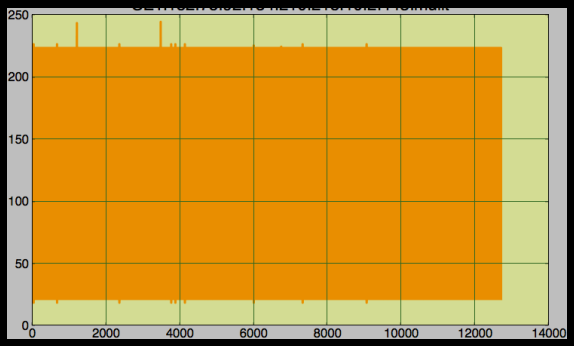
Graphing 1

Findings
Dialing the tolerances to each network is important
If you open the tolerance to include traffic just outside the statistical significant will leads to interesting results

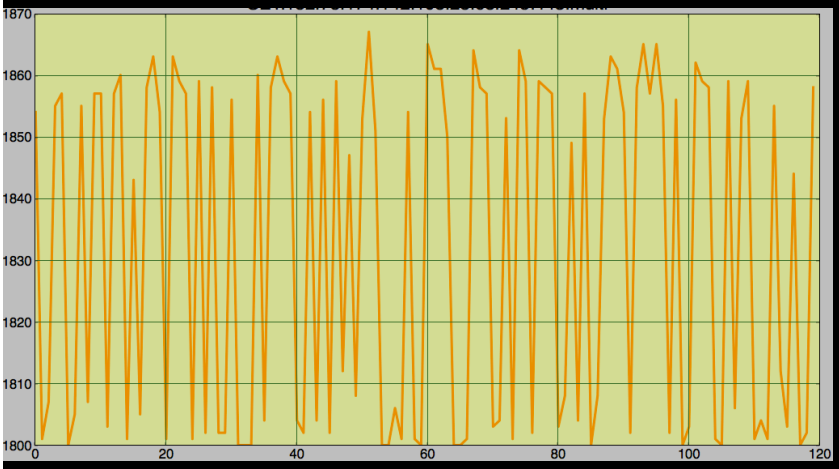


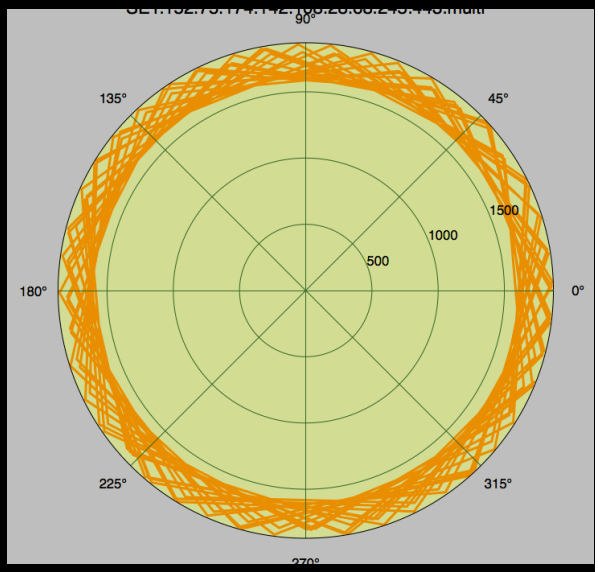






timing of a sample beacon





Considerations

- 1. Tune variables to a specific network
  - a. Host count
  - b. vistors
- 2. Outlier reject may exclude useful results
- 3. Results should include domain results
- 4. Excluding trusted sources saves time
  - a. Trusted list requires management
- 5. Continuous collection and periodic analysis needs more testing
  - a. Expiration of data (production)
  - b. Scheduled analysis
    - i. Output top list
      - 1. include graphical output
  - c. Require periodic flush of the database

Conclusion

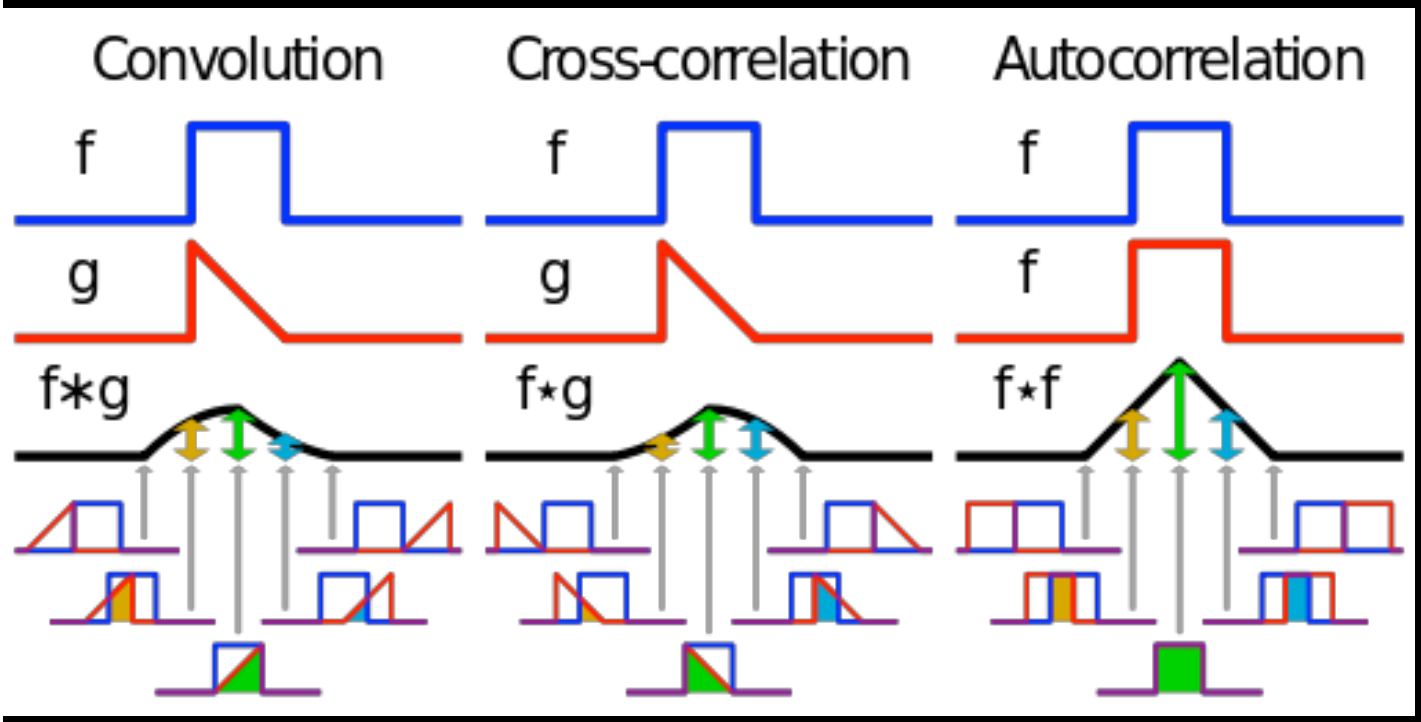
1. Timing is a signature
2. Expanding beacon detection to include payload analysis seems useful
3. Full packet capture can assist in validating threats
4. Host inspection is the best way to validate threats
5. Expand tracking to include DNS
6. Variable timing is difficult but not impossible to include in the analysis
7. Easy to include nslookup and whois results in our dataset

Tools

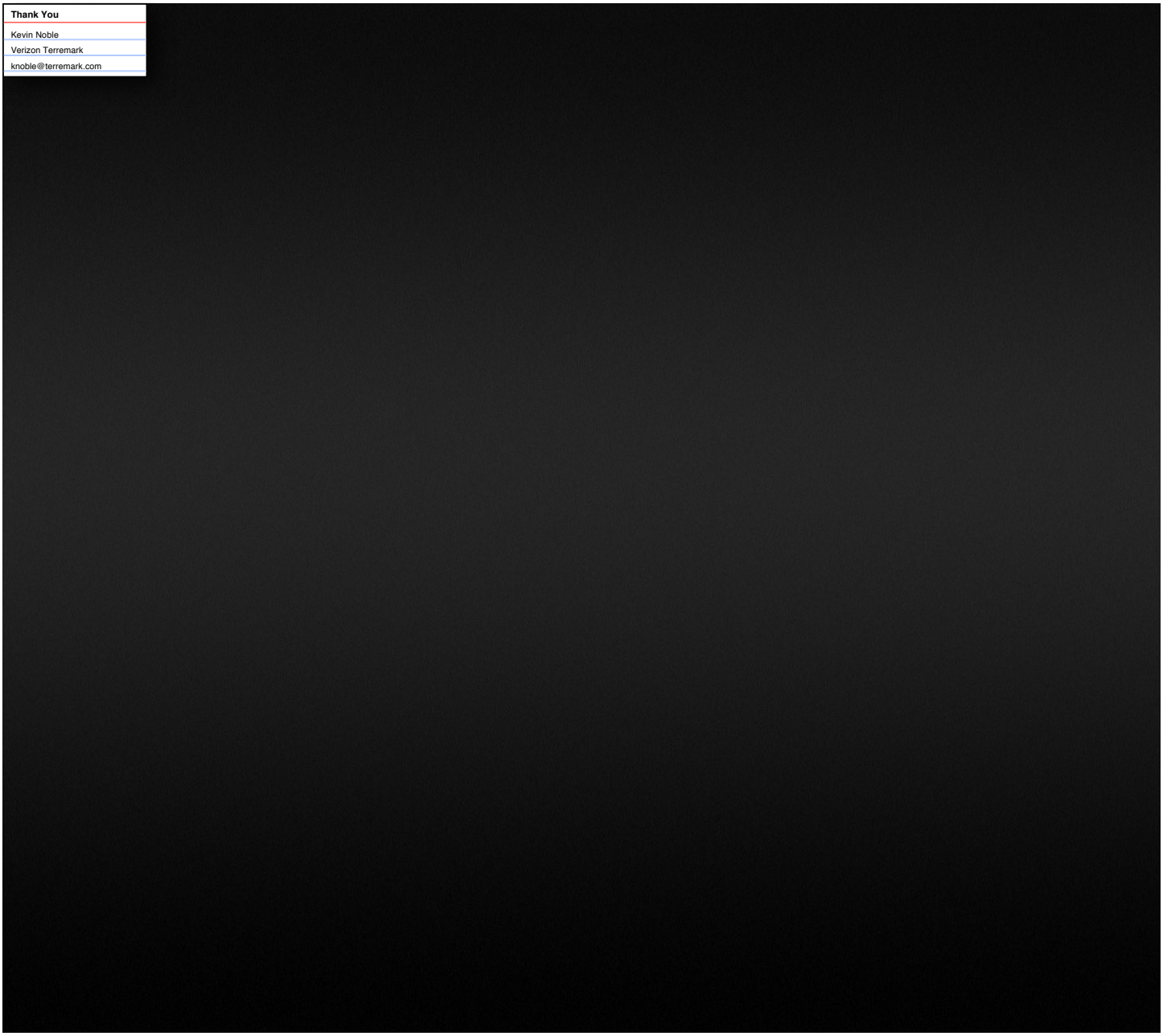
- 1. Flow collection
  - a. Code http://code.google.com/p/beaconbits
  - b. ARGUS
    - i. http://www.qosient.com/argus/
- 2. Dev Code
  - a. Python 2.7.1
    - i. Library for Redis
      - 1. https://github.com/andymccurdy/redis-py
    - ii. Library for Stats
      - 1. http://www.jstor.org/stable/1266577
      - 2. NUMPY
      - 3. MATPLOTLIB
  - b. IDE editor
    - i. Komodo IDE V2
- 3. Database
  - a. Redis 2.5.11 (00000000/0) 64 bit
    Running in stand alone mode
    http://redis.io
- 4. Presentation
  - a. CURIO
    - i. http://www.zengobi.com/products/curio

Future considerations

1. Release a production capable version (with enough public interest)
2. Release a stand alone version (no redis required, just reads flows and outputs)
3. Include the use of exclusion list (trust / clean list)
4. Time series analysis with autocorrelation



Untitled


Thank You
Kevin Noble
Verizon Terremark
knoble@terremark.com

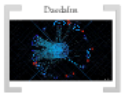
Pitfalls and Problems
Many ways to get visualization wrong.
Data Saturation: Our Data is Hard! (credit to Mr. Marty).
Graphics bring issues which command line tools avoid.

Tablets: Keytar of Security Visualization?
Huge push for tablets, from two directions:
Youth and Executives. Very potent combination.
Seems inevitable that we will be showing security imagery on a tablet, soon.

What about other senses? May we engage hearing or touch to better experience the data?
Visualization isn't about seeing, it's about understanding

Thank You.

@securitytim
tray@21ct.com



Credit To The Giants

Rafael Marty -- "Applied Security Visualization"
Book about the specifics of security visualization.
Ben Shneiderman -- Too many publications to easily mention, user interface pioneer since 1980s.
Eight Golden Rules of Interface Design (google them!)
Shneiderman's Mantra -- Overview, zoom and filter, details on demand.

Makers and Clancy -- "Whatever You Say, Say Nothing"
Best network security song ever.

Where are we going?
3D? Has to be a valid use case, but it could happen.
More automation and grouping w/o human intervention.

Heat Maps

Dashlets

Tim Ray

Amplify FCU

Department of Information Resources, State of Texas

21CT

@SecurityTim

Origin Systems (EA)

Credit To The Giants

Rafael Marty -- "Applied Security Visualization"

Book about the specifics of security visualization.

Ben Shneiderman -- Too many publications to easily mention, user interface pioneer since 1980s.

Eight Golden Rules of Interface Design (google them!)

Shneiderman's Mantra -- Overview, zoom and filter, details on demand.

Makem and Clancy -- "Whatever You Say, Say Nothing"

Best network security song ever.

Why Visualize?

Dense Information -- Humans process visual information very fast and have discriminatory power hard to replicate in a machine.

Internal Marketing

- Must convince non-technical folks
- Security as elevator pitch

Art is Acquisition, Content is Retention

Overview, Zoom and Filter, Details on Demand

Lesson from Games:
Art is Acquisition,
Content is Retention

We did this backwards, IMO

It applies to everything, even netflow security.

Magazine or Broken iPad?

Overview, Zoom and Filter, Details on Demand

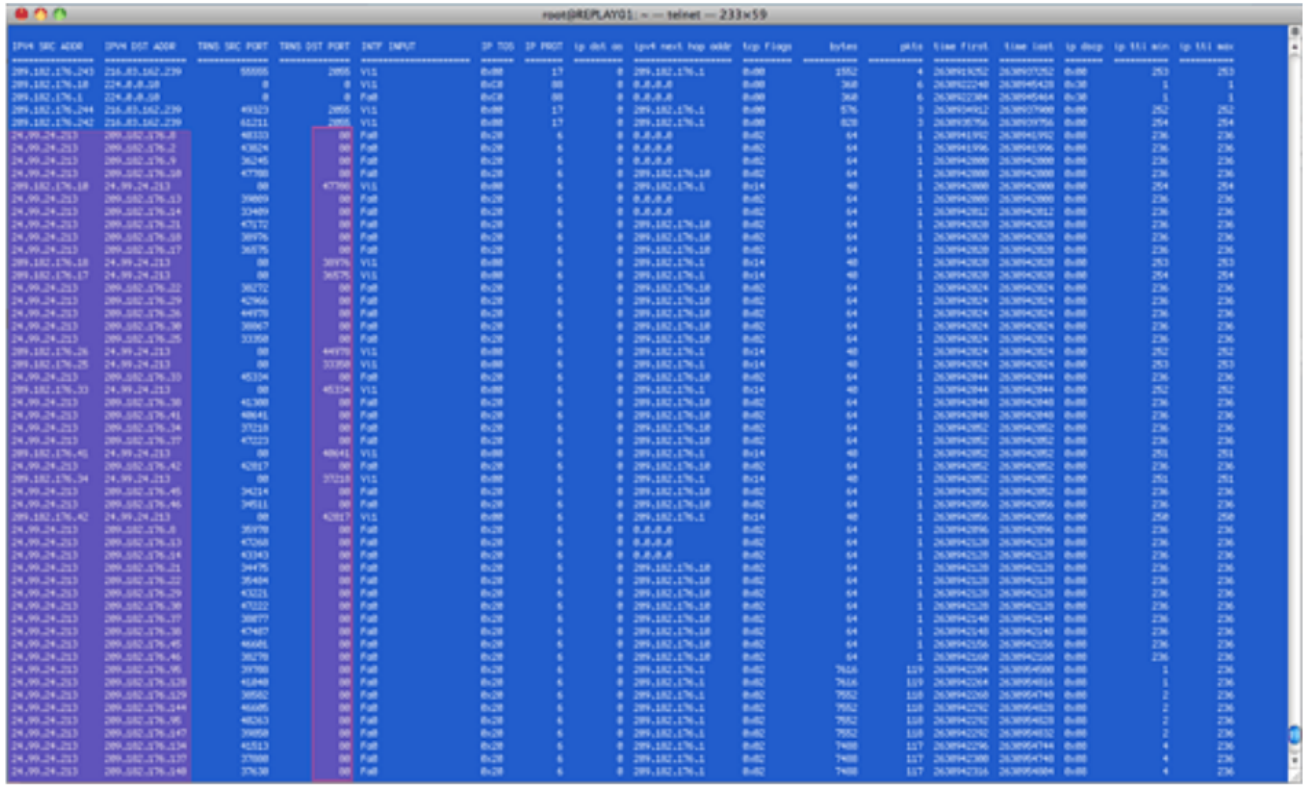
--Ben Shneiderman

The Way We Were
What did we do before?
Lots of spreadsheets and text files.

Not that there's anything wrong with that…

Why was that not enough?
Database search advantage.
Need to search for more than one thing at a time.
Closer to real time is better.
Note: It may be enough in your environment!

Src IP	Dst IP	Appln	Src Port	Dst Port	Protocol	DSCP	TCP FLAGS	Flow Rate	Traffic	Packets	NextHop	FNF NbarApp
192.168.10.	192.168.13.1	compressnet	2	169	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	15.80.39.28	-
192.168.10.	192.168.13.1	compressnet	2	564	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	190.23.69.213	-
192.168.10.	192.168.13.1	compressnet	2	741	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	95.55.61.159	-
192.168.10.	192.168.13.1	compressnet	281	2	TCP	AF12	UAP SF	1.0 Kbps	1.0 KB	2	223.191.78.79	-
192.168.10.	192.168.13.1	compressnet	165	3	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	64.15.70.93	-
192.168.10.	192.168.13.1	compressnet	424	3	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	149.191.44.148	-
192.168.10.	192.168.13.1	compressnet	822	3	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	73.7.175.22	-
192.168.10.	192.168.13.1	rje	800	5	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	74.157.168.220	-
192.168.10.	192.168.13.1	discard	9	714	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	1.209.56.6	-
192.168.10.	192.168.13.1	daytime	13	252	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	232.237.195.100	-
192.168.10.	192.168.13.1	daytime	960	13	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	22.88.95.248	-
192.168.10.	192.168.13.1	msp	18	116	TCP	AF12	UAP SF	1.0 Kbps	1.0 KB	2	81.203.252.131	-
192.168.10.	192.168.13.1	msp	18	735	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	18.50.17.126	-
192.168.10.	192.168.13.1	ftp-data	20	513	TCP	AF12	UAP SF	0 Kbps	1.0 KB	2	240.7.195.4	-



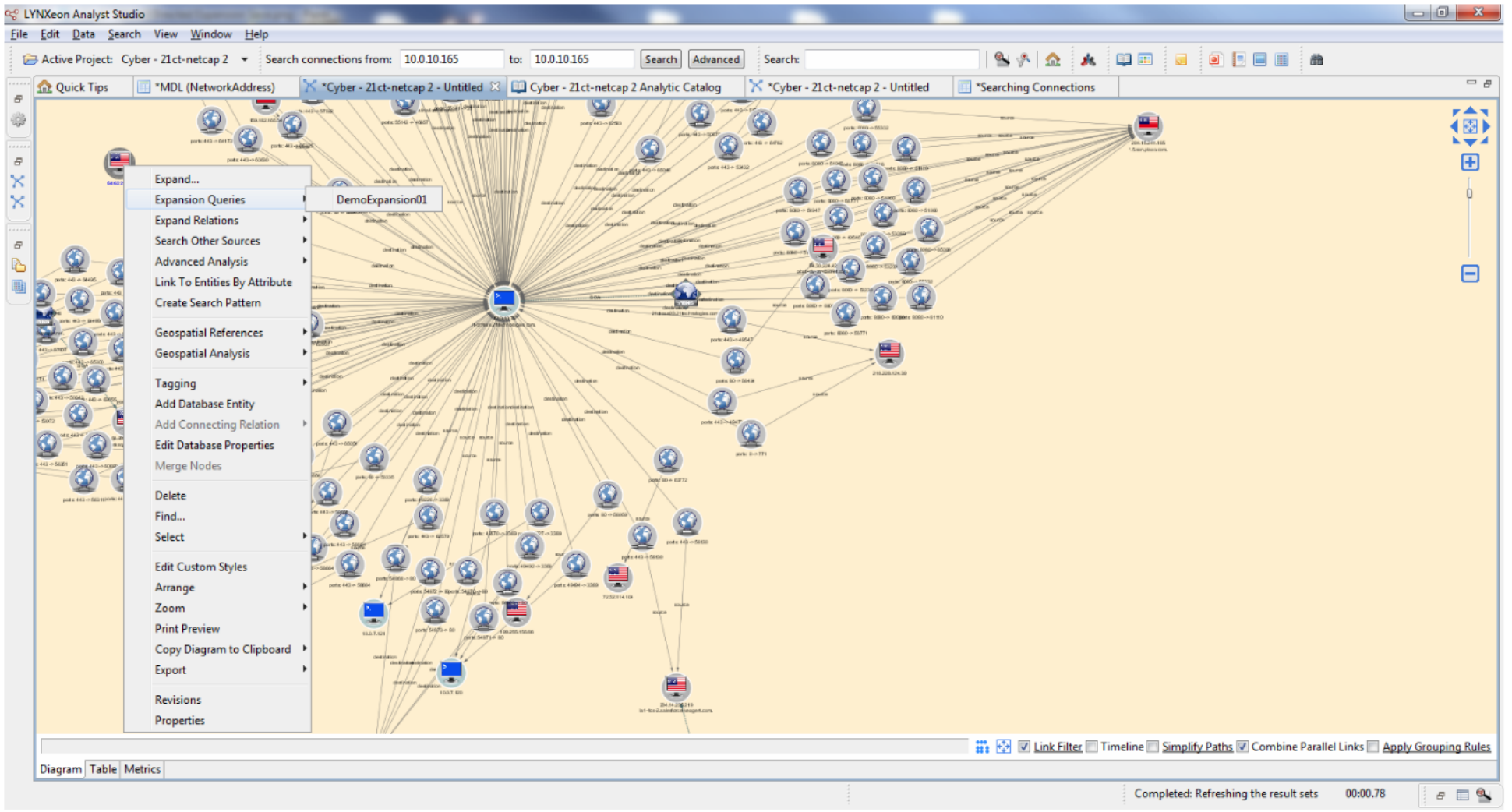
The Way We Were
What did we do before?
Lots of spreadsheets and text files.

Not that there's anything wrong with that…

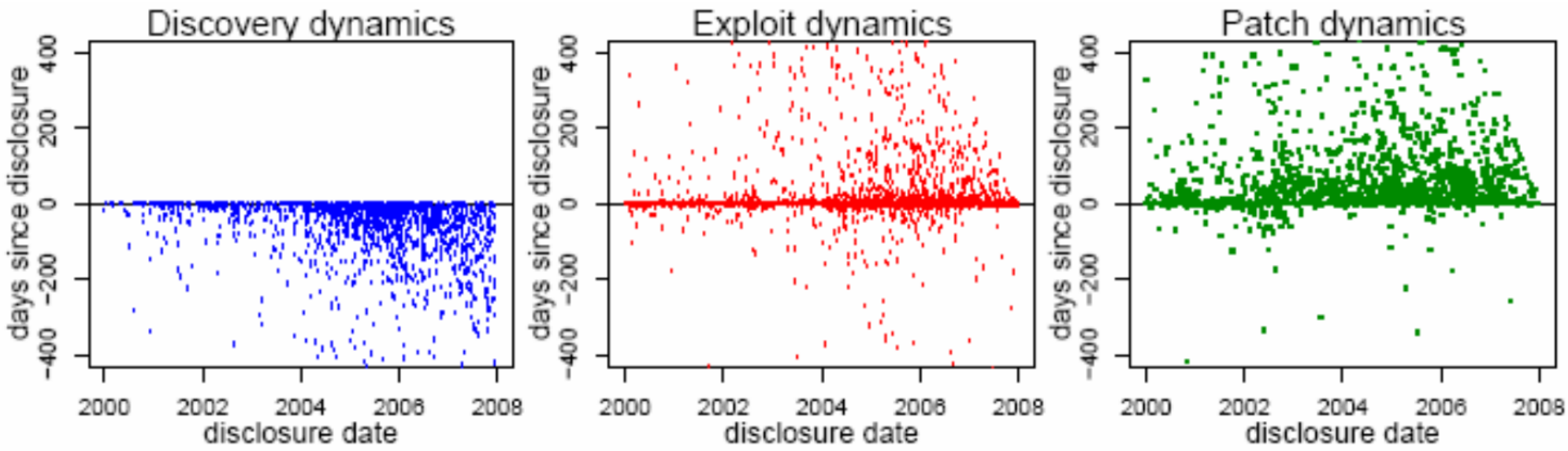
Why was that not enough?
Database search advantage.
Need to search for more than one thing at a time.
Closer to real time is better.
Note: It may be enough in your environment!

What are we Doing Now?

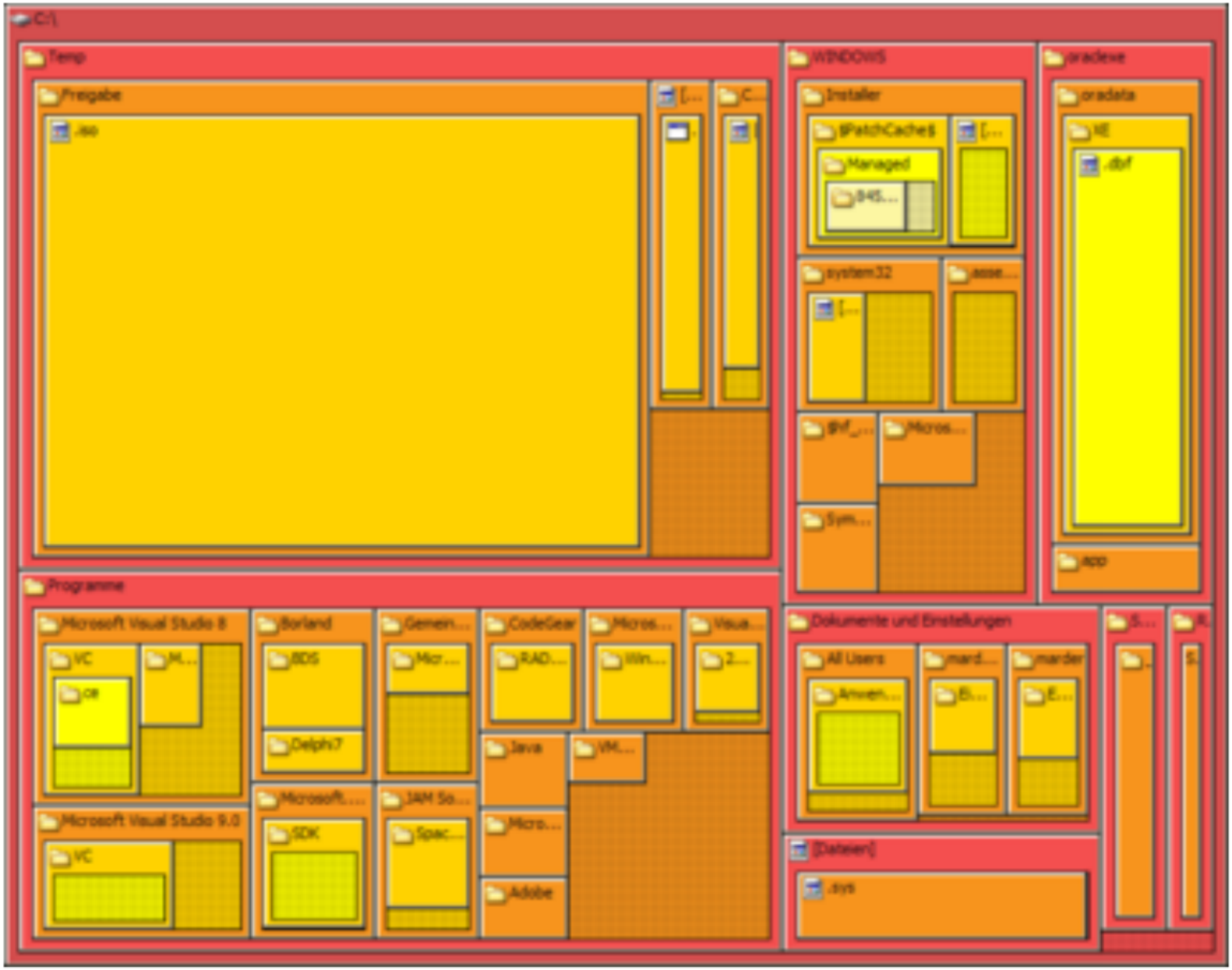
Graphs


LYNXeon Analyst Studio
File Edit Data Search View Window Help
Active Project: Cyber - 21ct-netcap 2
Search connections from: 10.0.10.165
to: 10.0.10.165
Search
Advanced
Search:
Quick Tips
*MDL (NetworkAddress)
*Cyber - 21ct-netcap 2 - Untitled
Cyber - 21ct-netcap 2 Analytic Catalog
*Cyber - 21ct-netcap 2 - Untitled
*Searching Connections
Expand...
Expansion Queries
DemoExpansion01
Expand Relations
Search Other Sources
Advanced Analysis
Link To Entities By Attribute
Create Search Pattern
Geospatial References
Geospatial Analysis
Tagging
Add Database Entity
Add Connecting Relation
Edit Database Properties
Merge Nodes
Delete
Find...
Select
Edit Custom Styles
Arrange
Zoom
Print Preview
Copy Diagram to Clipboard
Export
Revisions
Properties
Link Filter
Timeline
Simplify Paths
Combine Parallel Links
Apply Grouping Rules
Diagram
Table
Metrics
Completed: Refreshing the result sets
00:00.78

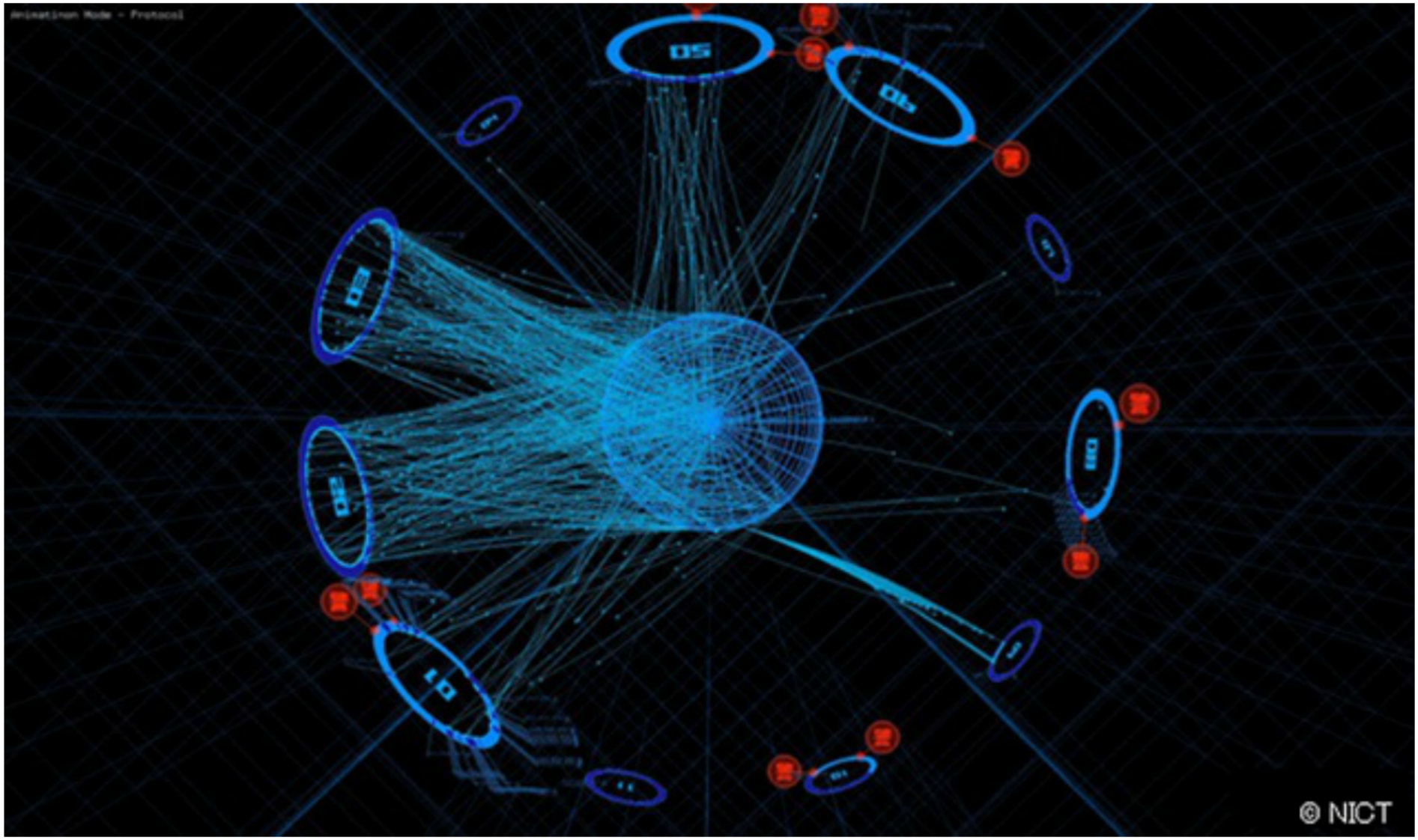
Scatter Plots



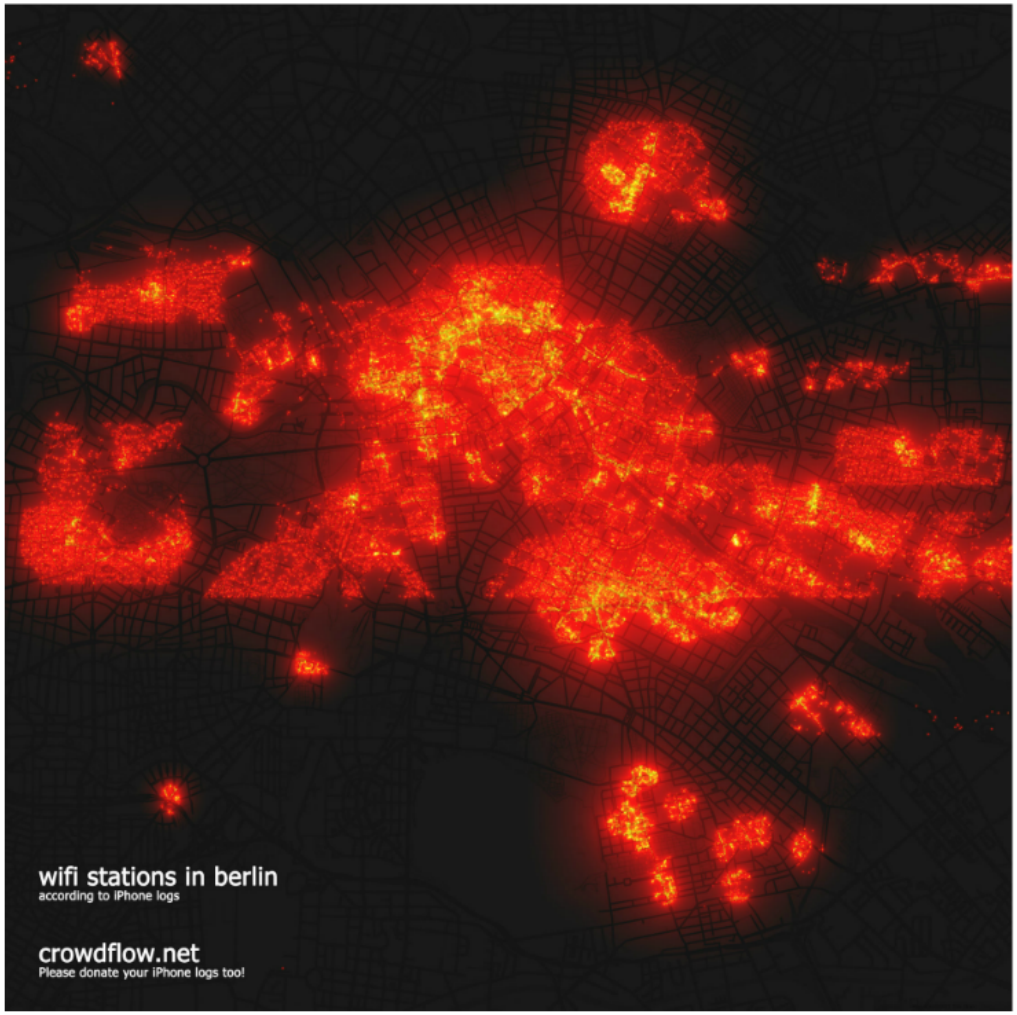
Treemaps


C:\
Temp
Freigabe
.iso
WINDOWS
Installer
$PatchCache$
Managed
system32
oracleve
oradata
.dbf
app
Programme
Microsoft Visual Studio 8
VC
Borland
BDS
Delphi7
Microsoft Visual Studio 9.0
VC
SDK
Java
Dokumente und Einstellungen
All Users
[Dateien]
.sys

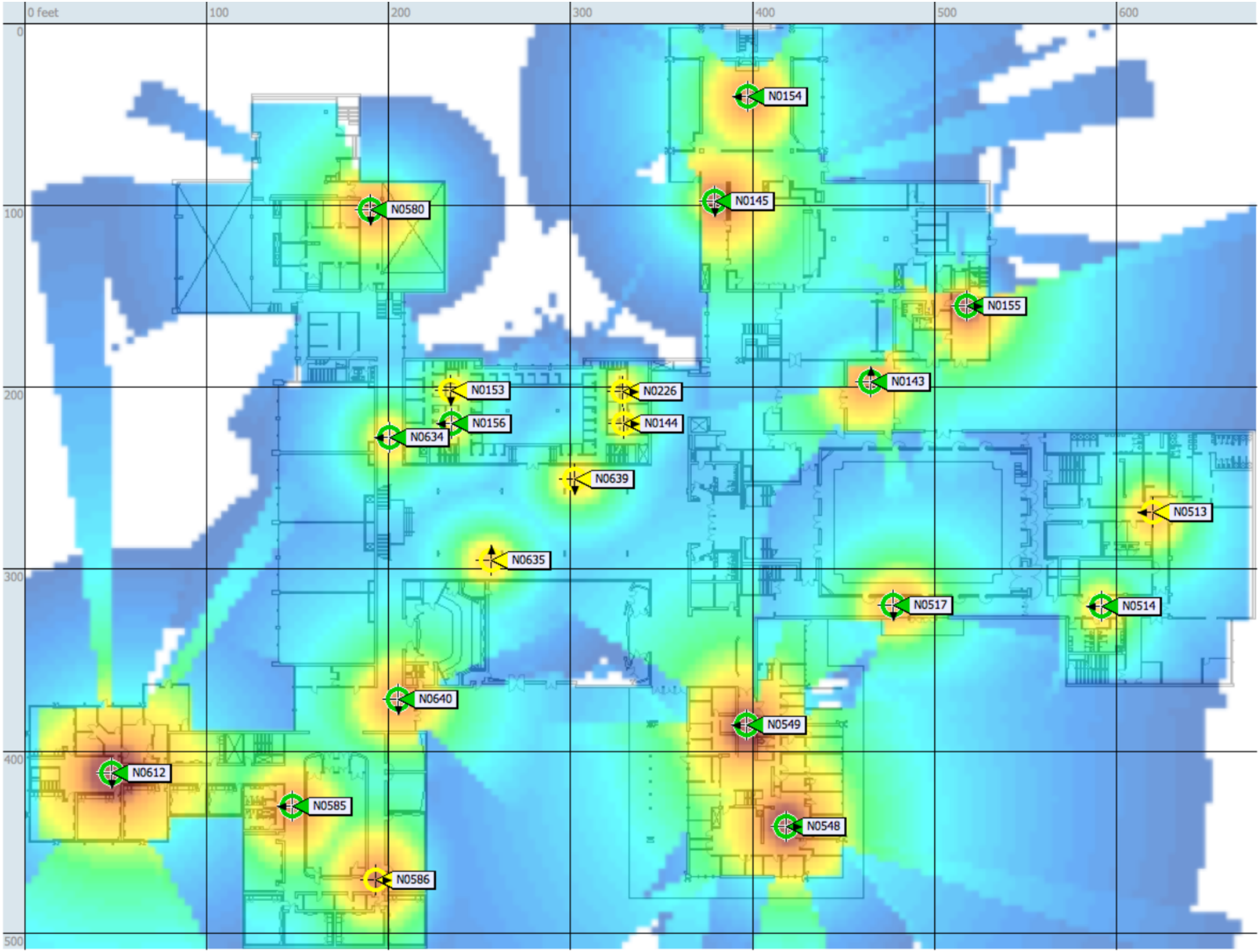
Daedalus



Heat Maps



Ok, little too hot...


0 feet
100
200
300
400
500
600
0
100
200
300
400
500
N0154
N0145
N0580
N0155
N0143
N0153
N0226
N0156
N0144
N0634
N0639
N0513
N0635
N0517
N0514
N0640
N0549
N0612
N0585
N0548
N0586

Where are we going?

3D? Has to be a valid use case, but it could happen.

More automation and grouping w/o human intervention.

Tablets: Keytar of Security Visualization?
Huge push for tablets, from two directions:
Youth and Executives. Very potent combination.
Seems inevitable that we will be showing security imagery on a tablet, soon.

What about other senses? May we engage hearing or touch to better experience the data?
Visualization isn't about seeing, it's about understanding


AT&T 3G
3:30 PM
Wednesday
23
Messages
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Notes
Utilities
iTunes
App Store
Settings
Phone
Mail
Safari
iPod
iPhone


TETRIS
CNN
HBO GO
epi
GUITAR HERO


Roland
Roland

Pitfalls and Problems

Many ways to get visualization wrong.

Data Saturation: Our Data is Hard! (credit to Mr. Marty).

Graphics bring issues which command line tools avoid.

Thank You.

@securitytim

tray@21ct.com

Introduction to Anomaly Detection

Char Sample <csample@cert.org>
George Jones <gmj@cert.org>
CERT/NetSA
FloCon 2013

Standard CERT Disclaimer

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE 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.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation 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.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

Where We Are Going

- **Introduction, Definitions, and Usage**
- Anomaly Collection and Classifications
- Anomaly Detection: Profiles & Attention Focusing
- Conclusion

Introduction

Assumption “Attacks exhibit characteristics that are different than those of normal traffic” (Denning, 1987).

Assumption validity

Introduction

Why do we care?

- In spite of following “best practices” vulnerabilities are still being discovered and exposed.
- Signature based solutions are failing miserably - new malware has < 10% detection rate by certain signature products.
- Fuzzing technologies make it easier for attackers to create their own 0 day attack. Fuzzing technologies work by automating the process of creative inputs, this in turn makes it easier for hackers to create their own 0 day attack.

Introduction

Why do we care?

- Anomaly detection provides an alternate approach than that of traditional intrusion detection systems. Jung et al., suggests modeling both normal and malicious behavior. (Jung, Paxson, Berger and Balakrishnan, 2004).
- Not all anomalies are malicious acts. (Sommer & Paxson, 2010)
- Most compelling reason: Anomalies have the potential to translate into significant critical and actionable information. (Chandola, Banerjee, & Kumar, 2009)
- AD is gaining popularity, this introductory presentation provides information and insight for deciphering the terms.

Introduction

The value of AD?

- AD represents an opportunity to see everything.
  - Good:
    - Capture 0 day attacks.
    - Define new analytics.
    - Gain a greater understanding of the network environment.
    - Proactive security posture.
    - Ability to better understand own environment.
    - Ability to complement existing solutions.
  - Bad:
    - Information overload.
    - Potential for improper use of models.
    - False positives are costly and incident handling is not easy nor automated.
    - Intrusion Detection has been shown to have fundamental differences from other areas where machine learning has been applied (Sommer & Paxson, 2010).

Definitions

Anomaly Definition

- A deviation from the norm; strange condition, situation or quality; an incongruity or inconsistency.
- Examples of network traffic anomalies:
  - IP address changes – New IP addresses appearing on sources and/or destinations found in logs.
  - Destination port changes – New destination ports showing up, especially combined with new destination addresses.
  - Command changes – sudden use of rarely used commands (e.g. Debug command, in HTTP or any other service).
  - Volume changes – sudden increases in service volume, destination volume.
  - Protocol anomalies – ssh over port 80, odd TCP flags, etc.

Definitions

Operational Profile

The operational profile of a system is defined as the set of operations that the software can execute along with the probability with which they will occur. An operation is a group of runs that typically involve similar processing (Lyu, 2002).

The Role of Profiles

Profiles are used to determine the norm, usual or expected behavior. They represent “baseline behavior”.

More on how we obtain profiles when we discuss collection and classifiers.

Anomaly Detection Usage

Uses for Anomaly Detection

- Detect precedent attack behavior. (CERT 2010)
  - APT assistance.
- Zero day attack detection.
- Intrusion detection.
- Insider threat detection
- Situational awareness.
- Validate and assist with signature data.

Anomaly detection can be considered the thoughtful process of determining what is normal and what is not.

Where We Are Going

- Introduction, Definitions & Usage
- **Anomaly Collection and Classifications**
- Anomaly Detection: Profiles & Attention Focusing
- Conclusion

Anomaly Collection

Machine Learning

- Un-Supervised learning
  - Gather information on the network passively, determine normal, build profile, then set decision boundaries.
    - Collects and builds.
    - Fast collection increase time spent on categorization.
- Supervised learning
  - Uses training data in order to learn the environment.
  - Provides groupings of learned categories.

Regardless of the learning method, the operational profile is the result of this step.

Anomaly Classification

Classifiers (decision support for uncertainty)

- Classifiers provide ways to organize the data.
- Commonly referenced models in anomaly classification:
  - Decision Tree
  - Bayes
  - Fuzzy
  - Certain types of clusters*

Where We Are Going

- Introduction. Definitions & Usage
- Anomaly Detection Usage
- Anomaly Collection and Classifications
- **Anomaly Detection: Profiles & Attention Focusing**
- Conclusion

Operational Profile Candidates

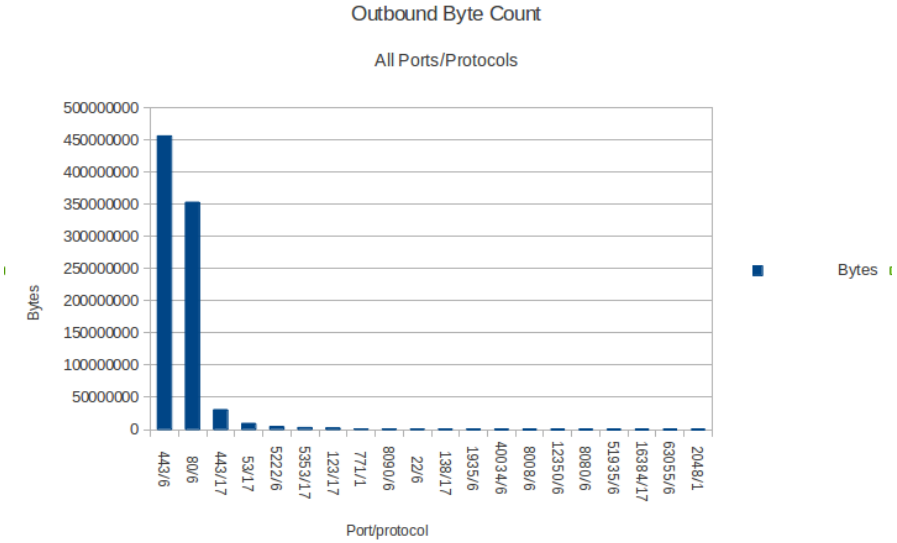
Here are a few candidates for operational profiles:

- Netflow (using SiLK names for fields)
  - sIP, dIP, sPort, dPort, pro, packets, bytes, flags, sTime, dur, eTime, sen, in, out, nhIP, scc, dcc, cla, type, sTime+msec, eTime+msec, dur+msec, iTy, iCo, initialF, sessionF, attribut, appli
- External Data Sources
  - DNS, ASN, WHOIS, GeoIP, blacklists, reputation
- Full Packet Data and Logs
  - IDS alerts, extracted URLs, extracted DNS responses, authentication logs, email headers, AV data…
- Application data, User behavior, Policy Violations
- Combinations of any of the above

Example Operational Profile
Outbound Bytes per Port

Operational Profile

- Statistical breakdown of outbound calls by service (proto+port)
- First data below shows top 13 services, %99.87 of all bytes
- Second data below shows bottom services. The interesting things are often in the noise.
- Graph shows the first data set.



pro	dPort	Bytes	%Bytes	cumul_%
6	443	452535897	52.718420	52.718420
6	80	353117060	41.136567	93.854987
17	443	29619475	3.450537	97.305524
17	53	9202986	1.072107	98.377631
6	5222	4347436	0.506457	98.884088
17	5353	3030322	0.353019	99.237107
17	123	2416876	0.281555	99.518662
1	771	982035	0.114403	99.633065
6	8090	642693	0.074871	99.707936
6	22	531313	0.061896	99.769831
17	138	415305	0.048381	99.818213
6	1935	303337	0.035337	99.853550
6	40034	223559	0.026044	99.879594

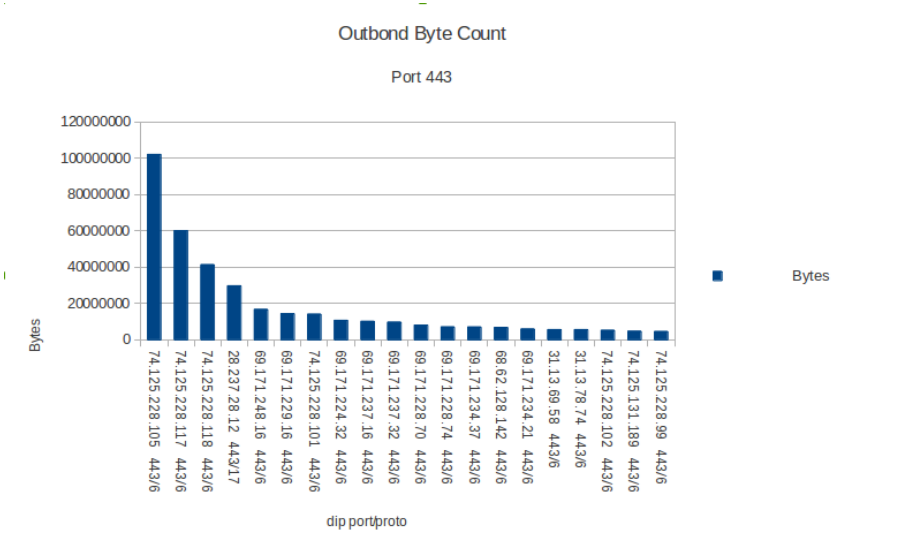
pro	dPort	Bytes	%Bytes	cumul_%
6	1935	303337	5039	99.858457
6	40034	208713	3424	99.880346
6	8080	192331	873	99.900517
6	8008	146607	853	99.915893
6	12350	110309	1830	99.927462
6	993	75351	1171	99.935365
6	51935	74724	1437	99.943202
17	16384	70248	104	99.950569
6	63055	58396	1123	99.956693

Example Operational Profile
Drilling down on one service: 443

Operational Profile

We know what services are normal, now we must find what is normal for the services.

- Drill down on outbound port 443
- Look at total bytes to destinations
- First data below shows top dests
- Second data shows bottom dests
- Graphic shows first data.
- Caveat: this data is cooked for the slides. There are inconsistencies.



pro	dIP	dPort	Bytes	%Bytes	cumul_%
6	74.125.228.105	443	99184312	20.571027	20.571027
6	74.125.228.117	443	60136869	12.472508	33.043535
6	74.125.228.118	443	41171738	8.539102	41.582637
17	128.237.28.12	443	28945131	6.003279	47.585916
6	69.171.248.16	443	16700906	3.463802	51.049718
6	69.171.229.16	443	14444553	2.995830	54.045547
6	74.125.228.101	443	14141718	2.933021	56.978568
6	69.171.224.32	443	10705577	2.220358	59.198926
6	69.171.237.16	443	10206140	2.116774	61.315701
6	69.171.237.32	443	9759443	2.024128	63.339829

pro	dIP	dPort	Bytes	%Bytes	cumul_%
17	157.55.235.146	443	46	0.000008	99.999943
17	111.221.77.154	443	46	0.000008	99.999951
17	111.221.77.162	443	46	0.000008	99.999959
17	157.55.235.148	443	46	0.000008	99.999968
17	157.55.235.161	443	46	0.000008	99.999976
17	64.4.23.140	443	46	0.000008	99.999984
17	111.221.74.16	443	46	0.000008	99.999992
17	111.221.77.152	443	46	0.000008	100.000000

Where We Are Going

- Introduction. Definitions & Usage
- Anomaly Detection Usage
- Anomaly Collection and Classifications
- Anomaly Detection: Profiles & Attention Focusing
- **Conclusion**

Conclusion

AD is gaining in popularity.

There are many different components of AD and the ones discussed represent only a portion, not a complete picture.

Understanding how the profile is built and what it represents is vital to understanding how the results were obtained.

It is important to how attention focusing is being directed.

References

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 15.

Denning, D.E., 1987, "An intrusion-detection model", *Software Engineering IEEE Transactions on,* (2):222-232.

Jung, J., Paxson, V., Berger, A.W., and Balakrishman, H., 2004, "Fast Portscan Detection Using Sequential Hypothesis Testing", *Security and Privacy 2004. Proceedings 2004 IEEE Symposium on* (pp. 211-225). IEEE.

References

Lyu, M., 2002, “Software Reliability Theory”, *Encyclopedia of Software Engineering.*

Sommer, R., & Paxson, V., (2010), “Outside the closed world: On using machine learning for network intrusion detection.”, In *Security and Privacy (SP), 2010 IEEE Symposium on,* pp. 305-316, IEEE, 2010

Thank you!

Char Sample & George Jones
CERT/NetSA
October 2012

TRALSE POSITIVE

Simple Methods for Confirming IDS/IPS Alerts

Introduction

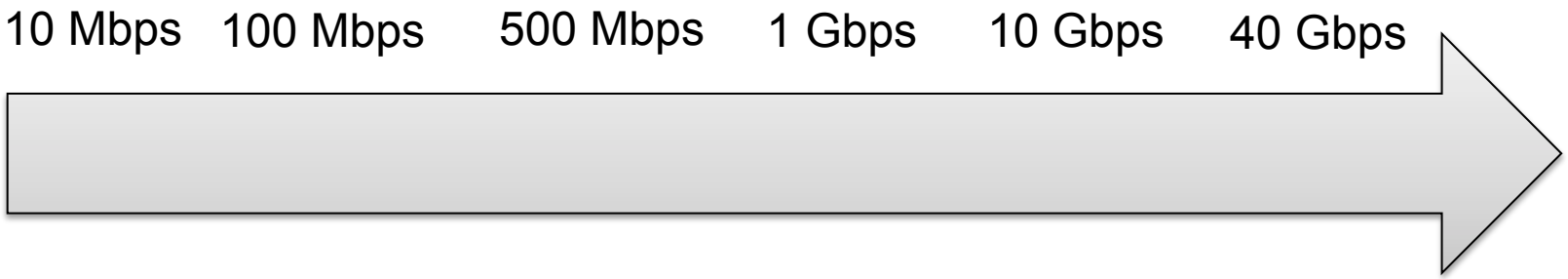
- Geoffrey Serrao
- Currently Employed at Sourcefire, Inc.
  - Tier I Technical Support Engineer
- Typical work day for a Tier 1
  - Hardware questions
  - Configuration questions
  - False positive analysis

IDS/IPS Alerts

- Big Three
  - Snort
  - Suricata
  - Bro IDS
- IDS/IPS systems generate alerts based on:
  - Signatures
  - Network Anomalies
- We will be dealing mostly with signature based events today

A Trend

- More data is being analyzed
- More events are being generated
- What do we do with all of these events?



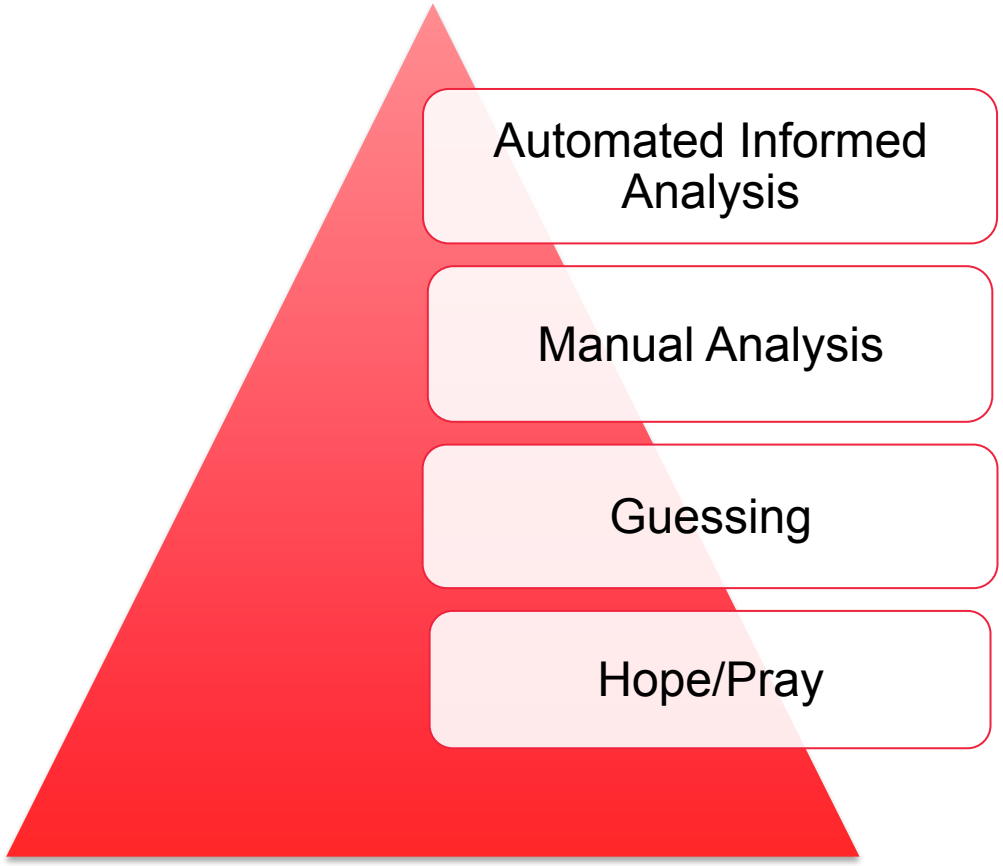
SOURCEfire®

Current Incident Handling Process

- Preparation
- Detection and Notification
- Investigation And Qualification
- Communication
- Containment and Recovery
- Lessons Learned

Existing Techniques

Best



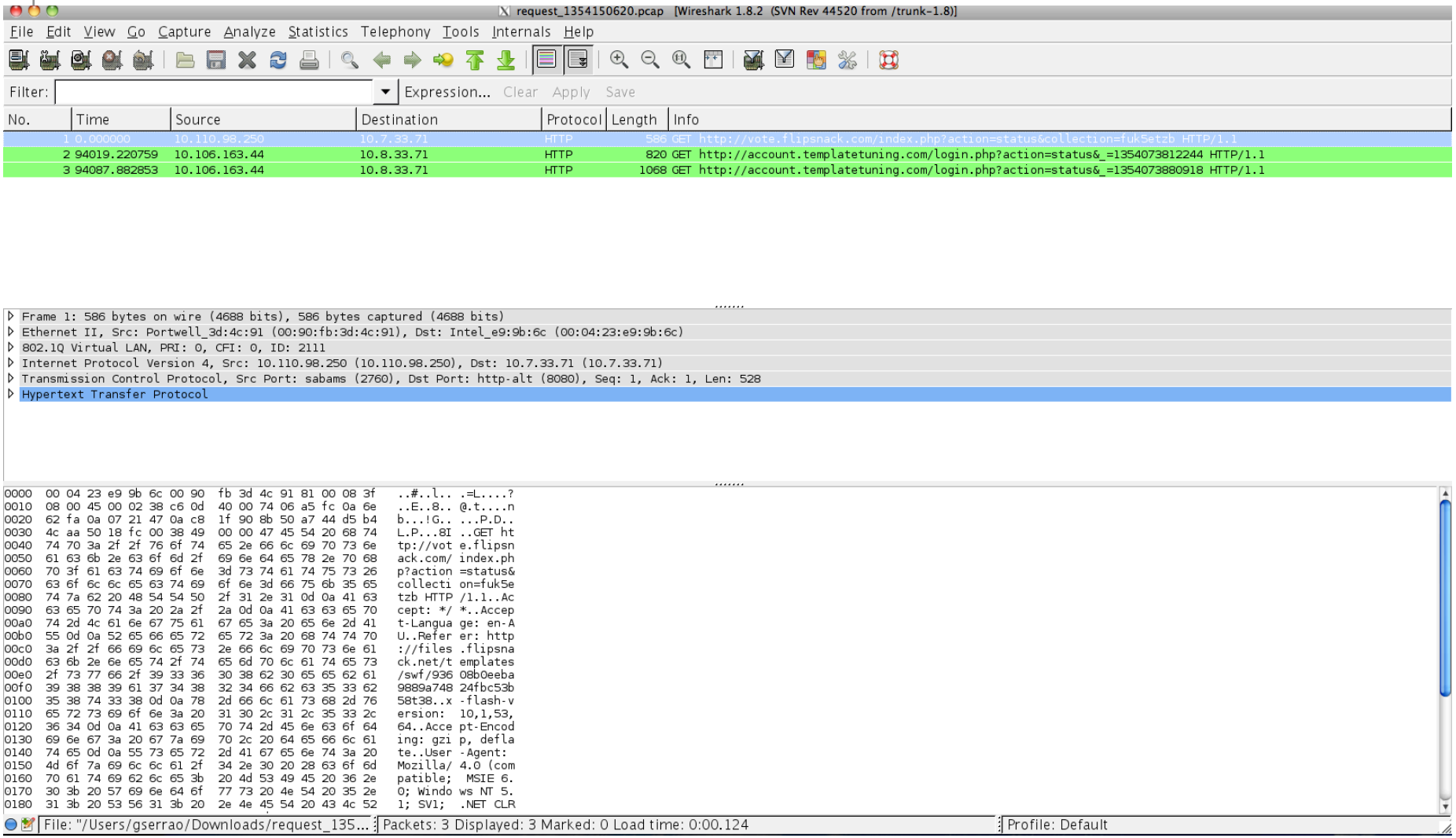
Worst

SOURCEfire®

The Current Method

- Step 1: Verify Rule Context
  - Rule Header
  - Content Matches
- Step 2: Verify Endpoints
  - Who's talking
- Step 3: Verify Conversation
  - What's being said – gets technical
- Step 4: Verify Operational Context
  - How does this type of attack affect my network deployment? – also gets technical

A Happy Example



Drawbacks of the Current Method

- Limited by the amount of information available to the analyst at the time
- Time intensive
- Tedious
- Reactive approach

Real World Example

traffic.pcap [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
607	35.484069	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
608	35.484072	72.21.195.15	10.8.0.4	TCP	294	[TCP segment of a reassembled PDU]
609	35.484087	10.8.0.4	72.21.195.15	TCP	54	50489 > http [ACK] Seq=853 Ack=117693 Win=523896 Len=0
610	35.484185	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
611	35.484280	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
612	35.484306	10.8.0.4	72.21.195.15	TCP	54	50489 > http [ACK] Seq=853 Ack=120613 Win=524280 Len=0
613	35.484376	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
614	35.484494	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
615	35.484518	10.8.0.4	72.21.195.15	TCP	54	50489 > http [ACK] Seq=853 Ack=123533 Win=524280 Len=0
616	35.484576	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
617	35.484682	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
618	35.484684	72.21.195.15	10.8.0.4	TCP	294	[TCP segment of a reassembled PDU]
619	35.484695	10.8.0.4	72.21.195.15	TCP	54	50489 > http [ACK] Seq=853 Ack=126693 Win=523896 Len=0
620	35.484794	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
621	35.484899	72.21.195.15	10.8.0.4	TCP	1514	[TCP segment of a reassembled PDU]
622	35.484900	72.21.195.15	10.8.0.4	HTTP	343	HTTP/1.1 200 OK (JPEG JFIF image)
623	35.484913	10.8.0.4	72.21.195.15	TCP	54	50489 > http [ACK] Seq=853 Ack=129902 Win=523848 Len=0
624	35.509065	10.8.0.4	10.8.0.187	TCP	78	50490 > netbios-ssn [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=425440370 TSecr=0 SACK_PERM=1
625	35.512143	10.8.0.187	10.8.0.4	TCP	74	netbios-ssn > 50490 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=11948705 TSecr=4254
626	35.512168	10.8.0.4	10.8.0.187	TCP	66	50490 > netbios-ssn [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=425440370 TSecr=11948705
627	35.519204	10.8.0.4	10.8.0.187	NBSS	138	Session request, to 10.8.0.187<20> from GSERRAO_MAC<00>
628	35.520890	10.8.0.187	10.8.0.4	NBSS	71	Negative session response, Called name not present
629	35.520924	10.8.0.4	10.8.0.187	TCP	66	50490 > netbios-ssn [ACK] Seq=73 Ack=7 Win=524280 Len=0 TSval=425440370 TSecr=11948706
630	35.520999	10.8.0.4	10.8.0.187	TCP	66	50490 > netbios-ssn [FIN, ACK] Seq=73 Ack=7 Win=524280 Len=0 TSval=425440370 TSecr=11948706
631	35.521062	10.8.0.4	10.8.0.187	TCP	78	50491 > netbios-ssn [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=425440370 TSecr=0 SACK_PERM=1
632	35.522402	10.8.0.187	10.8.0.4	TCP	66	netbios-ssn > 50490 [ACK] Seq=7 Ack=74 Win=66560 Len=0 TSval=11948706 TSecr=425440370
633	35.522752	10.8.0.187	10.8.0.4	TCP	74	netbios-ssn > 50491 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=11948706 TSecr=4254
634	35.522765	10.8.0.4	10.8.0.187	TCP	66	50491 > netbios-ssn [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=425440370 TSecr=11948706
635	35.531185	10.8.0.4	10.8.0.187	NBSS	138	Session request, to 10<20> from GSERRAO_MAC<00>
636	35.532248	10.8.0.187	10.8.0.4	NBSS	71	Negative session response, Called name not present
637	35.532348	10.8.0.4	10.8.0.187	TCP	66	50491 > netbios-ssn [ACK] Seq=73 Ack=7 Win=524280 Len=0 TSval=425440370 TSecr=11948707
638	35.532463	10.8.0.4	10.8.0.187	TCP	66	50491 > netbios-ssn [FIN, ACK] Seq=73 Ack=7 Win=524280 Len=0 TSval=425440370 TSecr=11948707
639	35.532577	10.8.0.4	10.8.0.187	TCP	78	50492 > netbios-ssn [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=425440370 TSecr=0 SACK_PERM=1
640	35.535807	10.8.0.187	10.8.0.4	TCP	66	netbios-ssn > 50491 [ACK] Seq=7 Ack=74 Win=66560 Len=0 TSval=11948708 TSecr=425440370
641	35.536057	10.8.0.187	10.8.0.4	TCP	74	netbios-ssn > 50492 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=11948708 TSecr=4254
642	35.536101	10.8.0.4	10.8.0.187	TCP	66	50492 > netbios-ssn [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=425440370 TSecr=11948708
643	35.542716	10.8.0.4	10.8.0.187	NBSS	138	Session request, to *SMBSERVER<20> from GSERRAO_MAC<00>
644	35.543786	10.8.0.187	10.8.0.4	NBSS	71	Negative session response, Called name not present
645	35.543865	10.8.0.4	10.8.0.187	TCP	66	50492 > netbios-ssn [ACK] Seq=73 Ack=7 Win=524280 Len=0 TSval=425440370 TSecr=11948708

```
0000  00 7f 28 b8 55 78 b8 8d  12 30 5d 16 08 00 45 00   ..(.Ux.. .0]...E.
0010  00 3d 8e 7c 00 00 ff 11  19 1f 0a 08 00 04 0a 08   .=.|.... ........
0020  00 01 ed e9 00 35 00 29  a3 30 19 bd 01 00 00 01   .....5.) .0......
```

File: "/Users/gserrao/Dropbox/pcapalyze/traf... | Packets: 3613 Displayed: 3613 Marked: 0 Load time: 0:00.584 | Profile: Default

SOURCEfire®

How to Improve

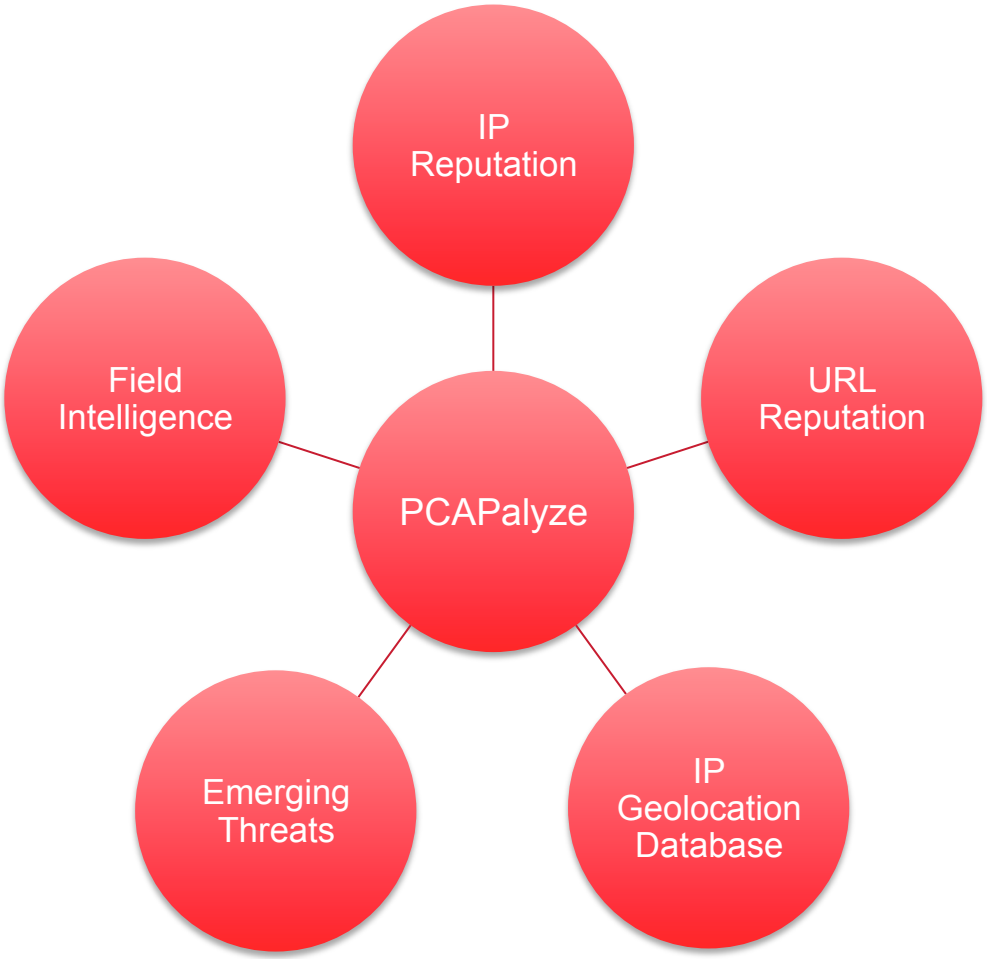
- Let's take a more proactive approach
- Increase the amount of information available to the analyst
- Increase the quality of the dissected payload
- Use automation tools
- The best methods are the most informed methods
- We need a bigger source of information

What I'd Like to See

IP's	rDNS	Verdict
…		
54.243.156.140	sourcefire.com	Clean
64.214.53.2	sf-nat.sourcefire.com	Clean
205.178.189.131	flocon.org	Clean
167.216.129.13	immunet.com	Clean
23.23.170.170	snort.org	Clean
69.43.161.180	antivirus-online21.com	+Investigate
192.88.209.252	cert.org	Clean
10.20.57.16	<none>	RFC 1918
…		

http://dns-bh.sagadc.org/domains.txt

Information Sources



Information Sources, Cont.

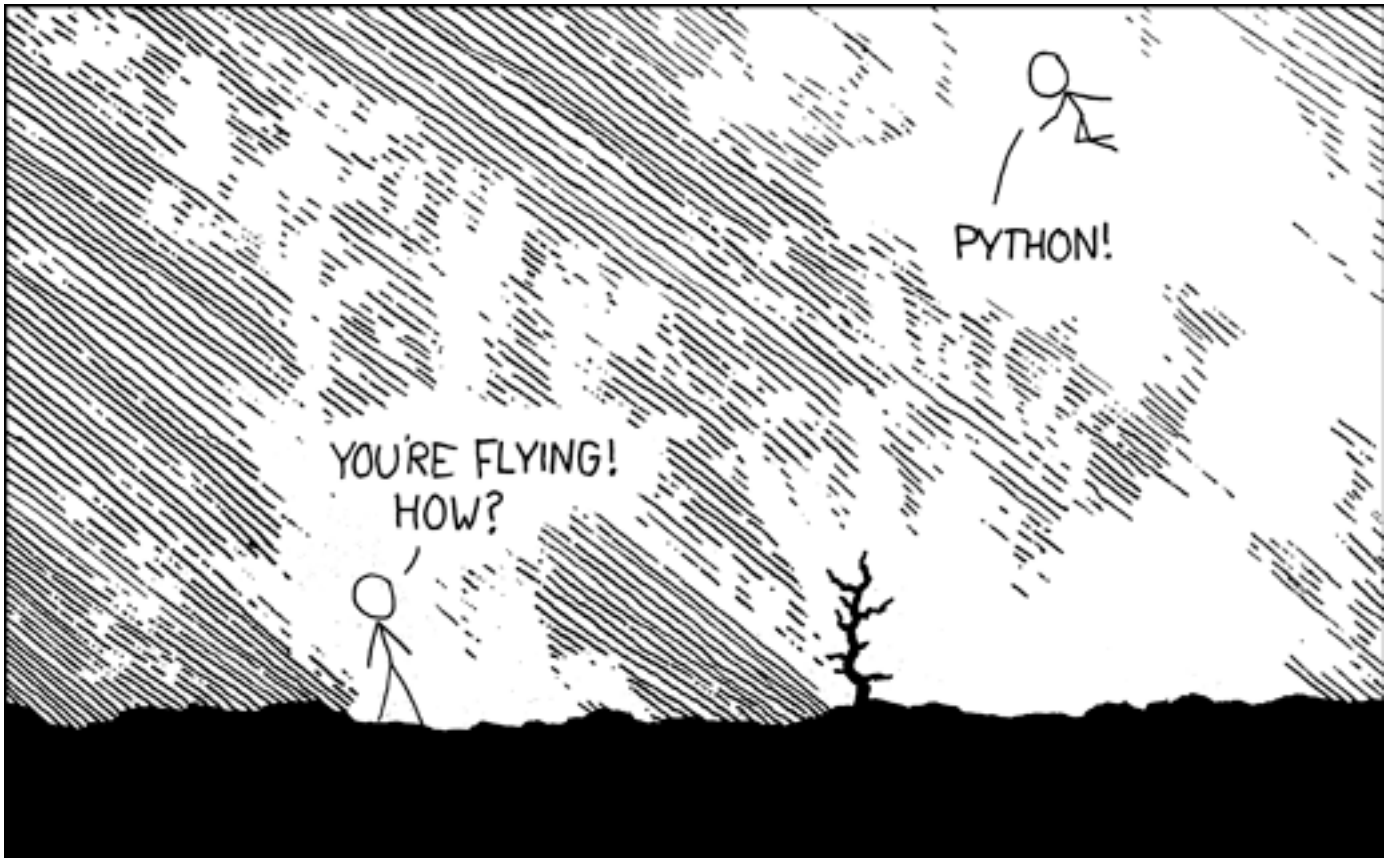
- Common
  - http://www.malwaredomains.com
  - www.mxtoolbox.com
  - https://www.dnsstuff.com/
  - http://www.siteadvisor.com/
  - https://www.phishtank.com/
- Not so common
  - Pastebin.com
  - Twitter.com

Favorite Information Source

- http://support.clean-mx.de/clean-mx/viruses

- They've been really tolerating my automated testing
- Easily encoded POST http requests for
  - IP
  - Domain

Python!



https://xkcd.com/353/

SOURCEfire®

The Code 1 of 3

```
from scapy.all import *
from scapy.utils import *
…
print "Reading PCAP(s):"
for x in range(num_pcaps):
        try:
                pkts.extend(rdpcap(caps[x]))
        except Exception, e: print e

print "Collecting IPs.."
for pkt in pkts:
        if pkt.haslayer(IP):
                if not pkt[IP].src in ip_list:
                        ip_list.append(pkt[IP].src)
                if not pkt[IP].dst in ip_list:
                        ip_list.append(pkt[IP].dst)
print len(ip_list), " unique IPs collected from pcap(s)”
…
```

The Code 2 of 3

```
for i in ip_list:
        if check_country:
                try:
                        location = str(GEOIP.lookup(i)).split('country')[1].strip('[] \n’ )
                except Exception, e:
                        print "country lookup failure.", e

        if check_hostname:
                try:
                        hostname = socket.getfqdn(i)
                except Exception, e:
                        hostname = "Couldn't find hostname", e
```

The Code 3 of 3

```
response = urlopen('http://support.clean-mx.de/clean-mx/viruses.php')
forms = ParseResponse(response, backwards_compat=False)
form = forms[0]

try:
                br = mechanize.Browser()
                …
                form['ip'] = i
                response = urlopen(form.click()).read()

                if not response.find('<br><br><div align="center"><b>For this
query is nothing recorded in our database.</b><br>') > -1:
                        reputation = "- Investigate"
                else:
                        reputation = "+ Clean"
```

Finished Output

```
-=Open proxy analysis=-
Got  248690  dangerous IPs.
Dangerous IPs matched: None

-=Full Analysis=-
6  IPs to check.
   169.10.11.239     US                    169.10.11.239        + Clean t:0.85781
    72.21.81.253     US                     72.21.81.253  - Investigate t:0.86674
 174.143.121.210     US                www.stylebistro.com       + Clean t:0.53773
   169.14.238.54     US                    169.14.238.54        + Clean t:0.87787
   169.10.22.247     US                    169.10.22.247        + Clean t:0.48822
 178.255.240.230     IT                    www.witcom.com  - Investigate t:13.33818
```

SOURCEfire®

Caveats and Pitfalls

- Customers with secure networks and tight data retention policies may not be able to take full advantage
- Working with encryption
- Tuning for accuracy

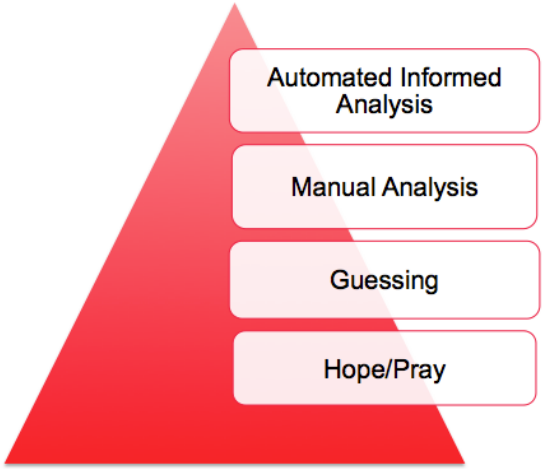
Future Development

- PCAPalyze
  - PHP web application (HTTPS) interface
  - Flask + Python back end
    - SCAPY used for extrapolating PCAP data
- Uses more sources of data
- Available for the public to use
- Works with more protocols

In Summation

IP's	rDNS	Verdict
…		
54.243.156.140	sourcefire.com	Clean
64.214.53.2	sf-nat.sourcefire.com	Clean
205.178.189.131	flocon.org	Clean
167.216.129.13	immunet.com	Clean
23.23.170.170	snort.org	Clean
69.43.161.180	antivirus-online21.com	+Investigate
192.88.209.252	cert.org	Clean
10.20.57.16	<none>	RFC 1918
…		

http://dns-bh.sagadc.org/domains.txt




```
-=Open proxy analysis=-
Got  248690  dangerous IPs.
Dangerous IPs matched: None

-=Full Analysis=-
6  IPs to check.
   169.10.11.239     US                    169.10.11.239        + Clean t:0.85781
    72.21.81.253     US                     72.21.81.253  - Investigate t:0.86674
 174.143.121.210     US                www.stylebistro.com       + Clean t:0.53773
   169.14.238.54     US                   169.14.238.54        + Clean t:0.87787
   169.10.22.247     US                   169.10.22.247        + Clean t:0.48822
 178.255.240.230     IT                    www.witcom.com  - Investigate t:13.33818
```


SOURCEfire®

Questions

Limitations of Traffic Analysis at Large Scale

Timothy J. Shimeall
CERT/NetSA
FloCon 2013

Notices

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE 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.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation 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.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

What We Will Cover

Overarching questions

What will we never know?

Analytical Limitations

Overarching Questions

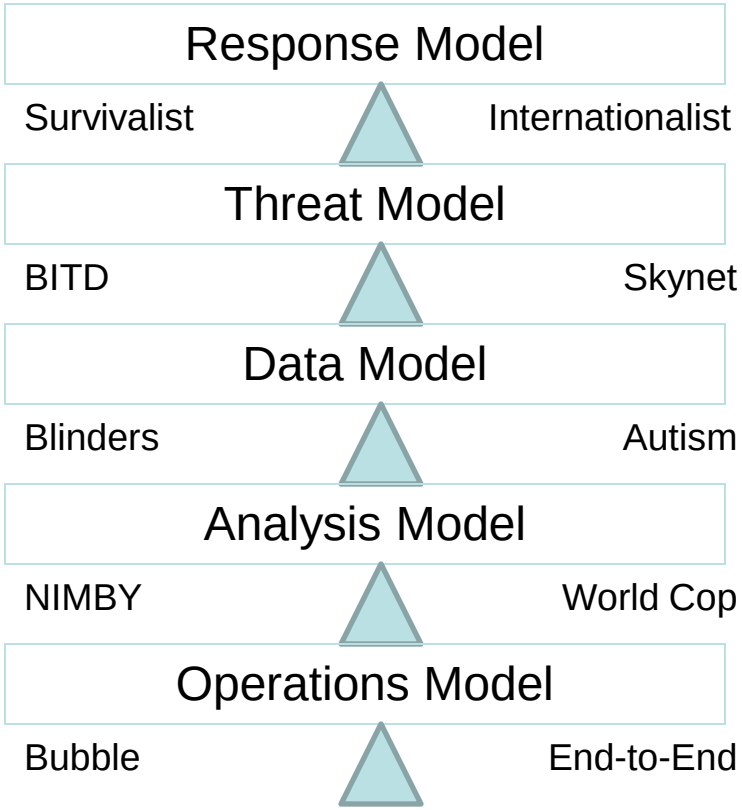
How do I know what I’m looking for?

How do I know why I’m looking?

How do I know where to look?

How do I know when it’s found?

Traffic Balance



What will you never know?

Total ground truth on a network of any size

How often is bad considered good? (false negative)

What is the next attack?

Why did they attack you?

What are your competitors seeing?

Inherently Partial Data

Technology shifts

Attacker actions

Defender actions

Managerial decisions

Network bandwidth

Correlation and Causation

Baseline in dynamic environment

Correlation vs. Causation

Implications

- Need to be cautious in kinds of conclusions
- Consider strategies for dealing with analysis gone wrong

Indication and Proof

Indication: There is reason to believe

Proof: There is no other logically defensible explanation

How much confidence do you need?

Cost of false positive?

Cost of false negative?

Conclusions

Many failure modes

Many challenges

Topic of continuing interest

Analysis of Communication Patterns in Network Flows to Discover Application Intent

Presented by:
William H. Turkett, Jr.
Department of Computer Science

WAKE FOREST
UNIVERSITY

FloCon 2013 | January 9, 2013

WAKE FOREST UNIVERSITY

Traditional Traffic Classification Techniques

Port- and payload signature-based classification techniques are increasingly less useful in modern traffic analysis.

Statistical approaches evaluating features such as packet size and interarrival times developed in response.

Traditional HTTP connection:

[src, src prt, dst, dst port, payload]

[10.1.11.58,8754, 10.19.132.45,80,

"GET /index.html"]

HTTP

Modern traffic:

[10.1.11.58,8754, 10.19.132.45,9090,

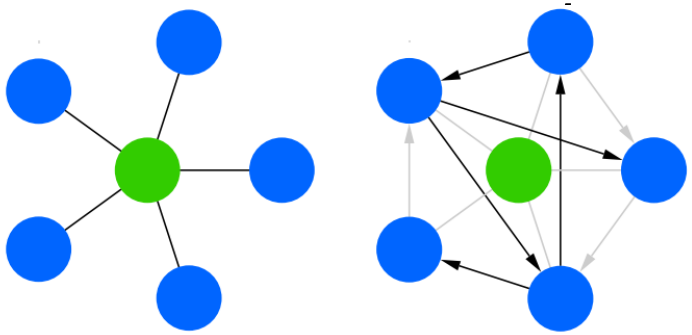
"xZvRmTTlFz"]

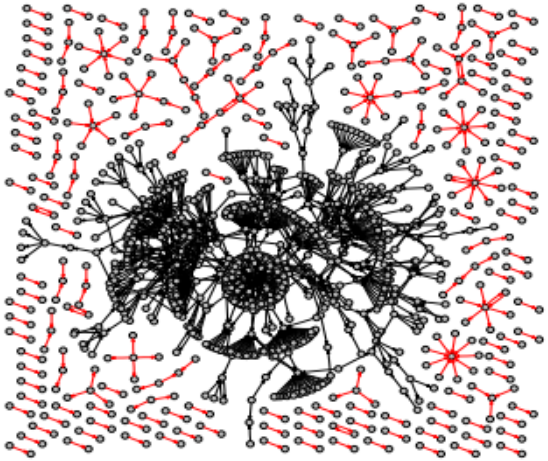
Encrypted payloads

Alternative ports/tunneling

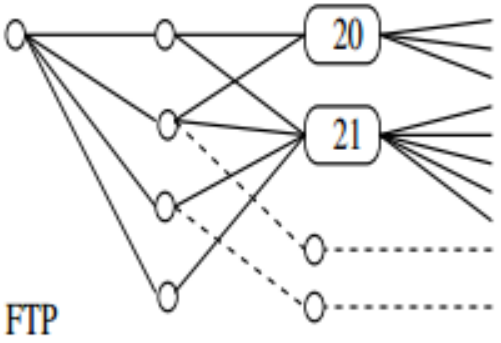
Graph Based Approaches To Traffic Classification

Graph based approaches look at the broader context of interactions (interaction networks instead of topological networks)





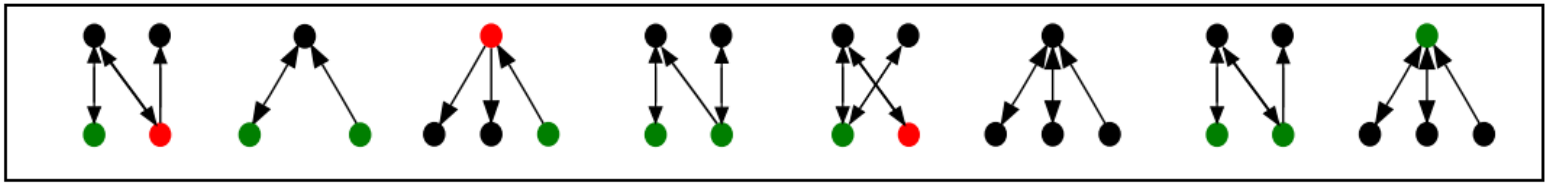
Graption – Traffic Dispersion Graph



BLINC - Graphlet

Karagiannis et al. - BLINC: Multilevel Traffic Classification In The Dark, SIGCOMM Proceedings, 2005.
Iliofotou et al. Graption: Graph-based P2P Traffic Classification At The Internet Backbone, Computer Networks, 2011

Communication Patterns And Motifs

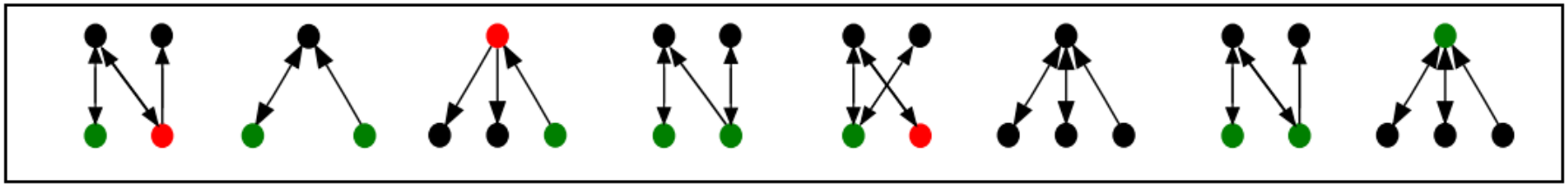


***Motifs* are patterns of interconnections occuring in networks at rates greater than expected by chance.**

Flow-level statistics can be employed to color graph nodes (hosts), allowing for *annotated motifs*

- *Bytes*: {Max, Average, Sum} bytes sent by a host over all connections host involved in
- *Duration*: {Max, Average, Sum} duration of connections host involved in
- *Node Type*: Client, server, or peer activity

Communication Patterns And Motifs



{ 1 0 0 0 1 1 0 0 }

Motif profiles **for a *host* represent in a binary vector which annotated motifs a host participates in**

FANMOD a tool for fast network motif detection

Tools such as *FANMOD* can mine graphs for motifs and determine host-level motif participation

Information Available From Flow Data

The data of interest to build graphs and color nodes is all accessible from flow data:

- Host-host interactions (Src-Dst)
- Summary-level statistics of traffic
  - Number of bytes transferred over connections
  - Duration of connections (timestamps)

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/ Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	162	/24	5	163	/24	15	10.0.23.2	1528	1745	4
Fa1/0	173.100.3.2	Fa0/0	10.0.227.12	6	40	0	2491	15	/26	196	15	/24	15	10.0.23.2	740	41.5	1
Fa1/0	173.100.20.2	Fa0/0	10.0.227.12	11	80	10	10000	161	/24	180	10	/24	15	10.0.23.2	1428	1145.5	3
Fa1/0	173.100.6.2	Fa0/0	10.0.227.12	6	40	0	2210	19	/30	180	19	/24	15	10.0.23.2	1040	24.5	14

- Assume can capture internal-to-internal and internal-to-external connections

A Deeper Problem: Discovery of Application Intent

Streaming media

Email

HTTP

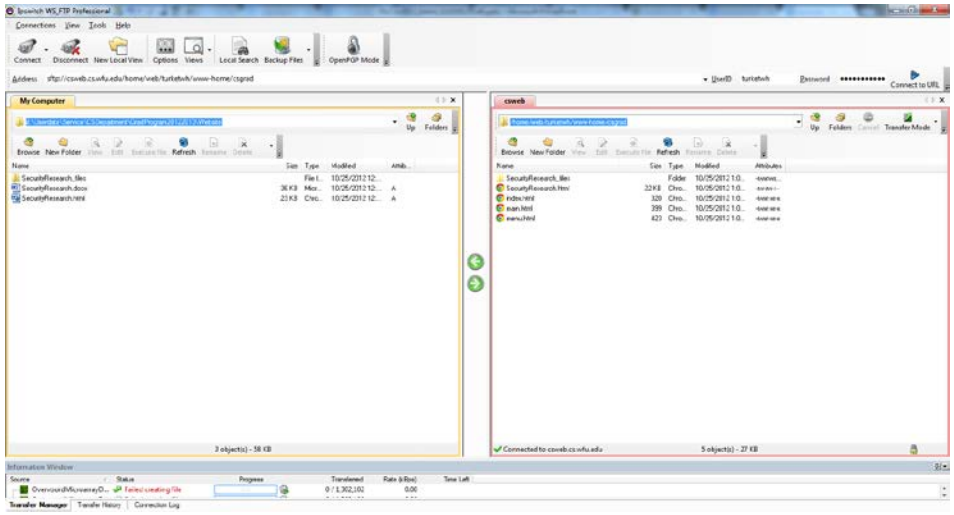
Chat

Browsing

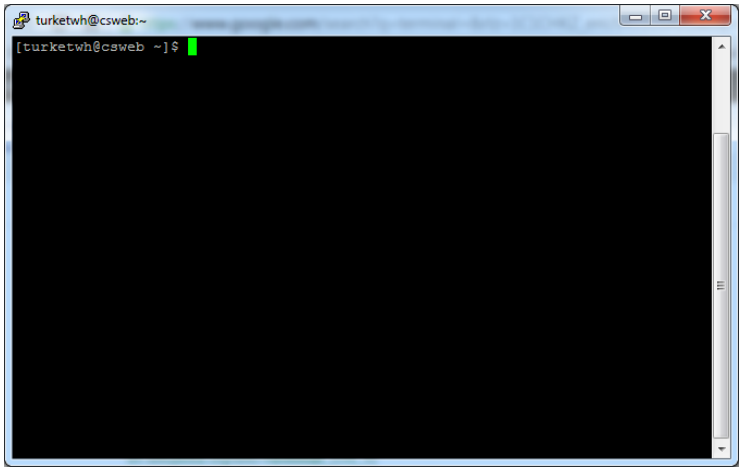
Single network protocols are now commonly employed for a variety of applications (intents)

WAKE FOREST UNIVERSITY

SSH: Application Intent



File Transfer



Terminal

Tunneling

Essence of Approach

Goal is labeling host intent from capture of a window of activity

- Potentially multiple connections within a window of activity
- Assuming that intents are used in *isolation* within a session

As designed currently, prime application is post-mortem analysis of host activity of interest.

Premise of research:

- *Annotated* and *directed* motifs capture significant information about communications
- Hypothesis: Distinct motif usage suggests distinct intent.

Traffic Classification Using Motifs: Initial Work

Our original work in this area (2009) explored separability of individual protocols, not intents.

Modeling approach consisted of:

- Construction of interactions graphs for each protocol
- Node coloring by host type (client/server/peer)
- Host motif profiles were over sets of size three or size four motifs from interaction graphs

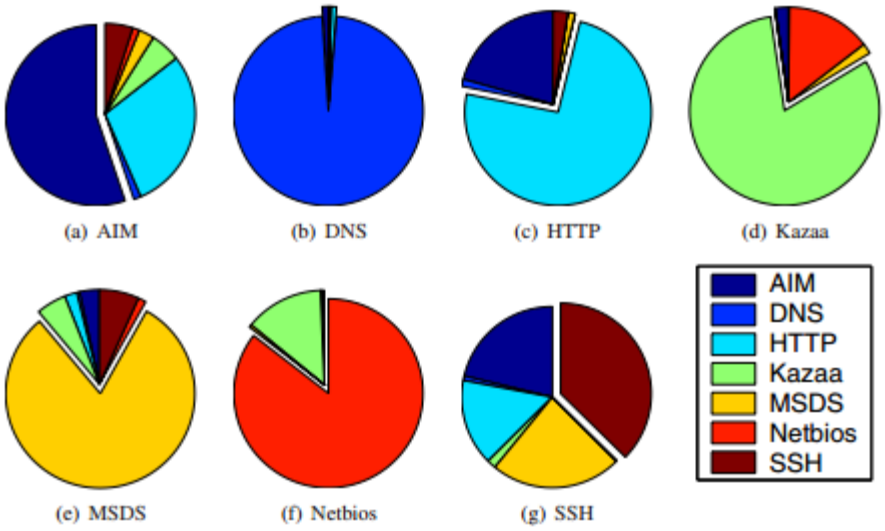
Host-protocol classification approach consisted of:

- Weighted-feature one-nearest-neighbor

WAKE FOREST UNIVERSITY

Protocol Separation Using Motifs

	True AIM	True DNS	True HTTP	True Kazaa	True MSDS	True NetBIOS	True SSH	Precision
Predicted AIM:	**298**	8	56	0	18	0	32	72.33%
Predicted DNS:	7	**632**	3	9	2	0	4	96.19%
Predicted HTTP:	120	14	**676**	0	19	3	23	79.06%
Predicted Kazaa:	5	0	1	**370**	5	34	1	88.94%
Predicted MSDS:	2	4	15	2	**269**	1	1	91.50%
Predicted NetBIOS:	0	1	0	0	2	**700**	0	99.57%
Predicted SSH:	36	0	19	1	57	2	**94**	44.98%
Recall:	63.68%	95.90%	87.97%	96.86%	72.31%	94.59%	60.65%	† **85.70%**



Data Sets For Intent Analysis

Goal is labeling host intent from capture of a window of activity

Properties of publicly available network datasets lead to difficulty in defining gold-standard datasets for training and analysis

- Privacy issues lead to IP shuffling and payload removal

Intent labeling is even harder

WAKE FOREST UNIVERSITY

Experimental Design: Flow Capture

For this work, flows were:

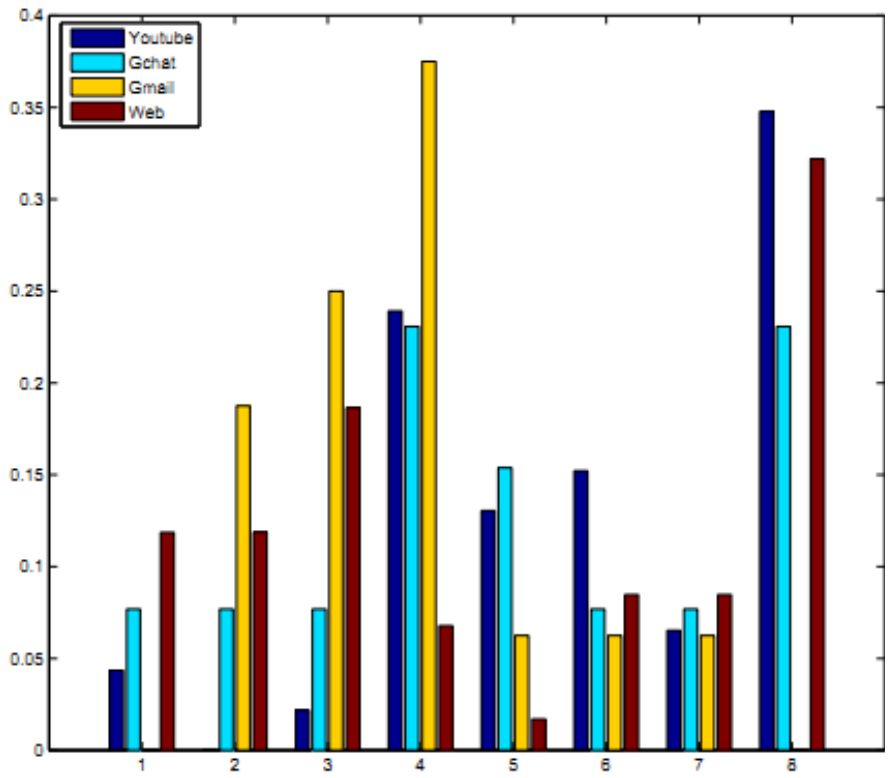
- Collected in-house
- Intents captured in isolation
- Captures automated through AutoIt scripts
- Kept any flows involved in a connection to purported HTTP host (port 80, 8080, 443)

Traffic Type	Source
Streaming media	Youtube
Email	GMail
Chat	GChat
Browsing	Yahoo random link generator

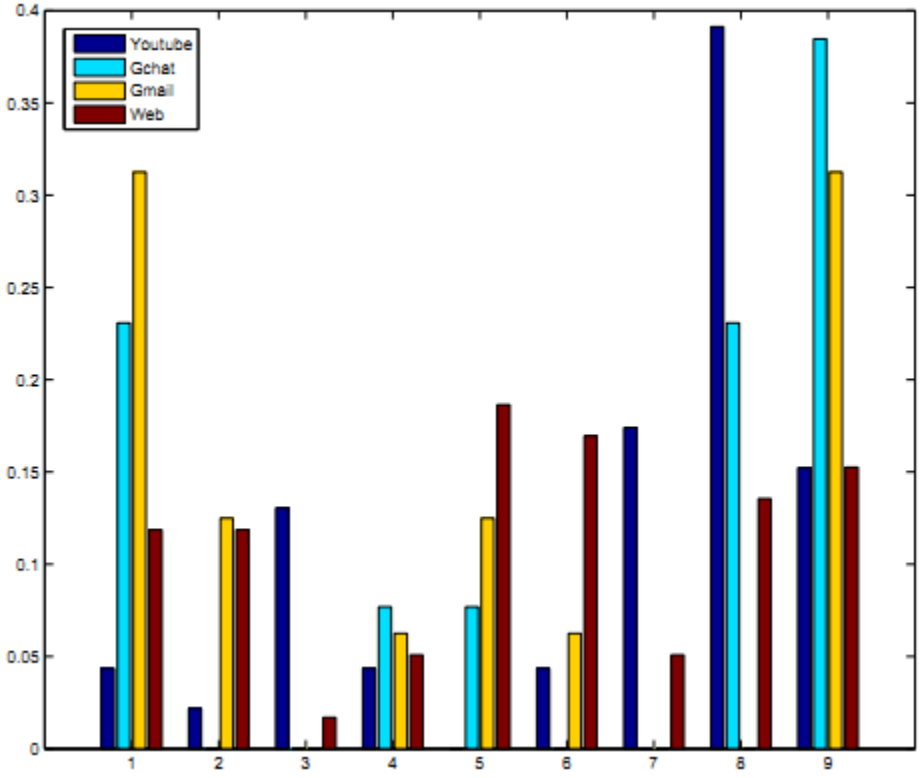
WAKE FOREST UNIVERSITY

Experimental Design: Histograms Of Annotation Statistics

No clear separation of distributions over bytes transferred or connection duration from visualization of flow statistics.



Average Bytes Transferred
(Binned, From Flow Statistics)



Average Flow Duration
(Binned, From Flow Statistics)

Experimental Design: SVM Approach and Results Summary

Support vector machine learning:

- Multiple “one-vs.-all” *support vector machine* models
- Max over model scores
- 10-fold cross validation

Accuracy across flow types (for small sample):

Truth	Total Flows	Node Type Only	Node Bytes + Type	**Node Duration + Type**
Gchat	21	0.71	1.00	**1.00**
Gmail	19	0.00	0.68	**1.00**
Browsing	71	1.00	0.97	**1.00**
Youtube	46	0.00	0.93	**0.94**

Node Duration & Type Results

Confusion matrix for model with best results – the model employing Node Duration and Type:

Label / **Truth**	Gchat	Gmail	Browsing	Youtube
Gchat	21	0	0	0
Gmail	0	19	0	0
Browsing	0	0	71	0
Youtube	3	0	0	43

Conclusions

Building evidence that subgraphs (motifs) of host interaction networks are related to type of activity (intent) being performed by hosts

Flow metrics, traditionally employed by statistical approaches to traffic analysis, can be embedded into graph structures through node coloring

Technology Transfer & Future Work

Online costs of deployment for approach:

- Building the host interaction network from network monitoring over time
- Determination of whether a host is involved in a set of motifs of interest
- Classification model scoring

Next steps:

- Refine traffic generation and collection processes
- Determine lower-limit on data required to accurately reflect a host's activity
- Remove assumption that intents are performed in isolation within a session of activity
- Understand the important motif structures

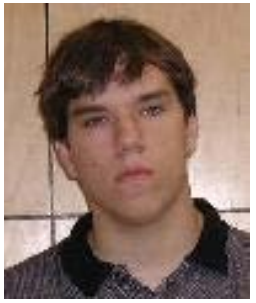
Acknowledgements

Network Security Colleagues at Wake Forest University

Dr. Errin Fulp

Brad McDanel

Lee Bailey



Tim Thomas

National Science Foundation

Grant # CNS-1018191

CyberV@R: A Model to Compute Dollar Value at Risk of Loss to Cyber Attack

FloCon 2013

James Ulrich [1]
CyberPoint Labs
julrich@cyberpointllc.com
CyberPoint International LLC

January 9, 2013

[1] with contributions from Charles Cabot, Roberta Faux, Scott Finkelstein, and Mark Raugas

Goals and Motivations

cyberpoint

- The ever-expanding threat of cyberattack presents IT administrators and CIOs with the daunting challenge of safeguarding their institutions' cyber infrastructure from breaches that could lead to catastrophic economic loss [Brenner2011], [Clarke2010], [EOPOTUS].
- Security resources remain finite, and deliberations on their wise allocation are aided by expressing risks and risk-reductions in dollar-denominated units.
- Even if we can't accurately predict overall economic loss, perhaps we can compare the relative economic benefit of alternative scenarios for resource allocation.
- So, we'd like a methodology for constructing risk models, at the organizational level, that give insight into relative, if not absolute, economic costs of cyber attack.

Proof of concept: Risk models in finance

cyberpoint

- In finance, trading desks maintain Value at Risk (VaR) models for measuring portfolio loss exposure.
- A VaR model answers the question "what is the amount of money $\$X$, such that the odds of losing more than $\$X$, over time window T, fall below some threshold of probability P?" We call this the "P-percent VaR."
- The most vanilla case (c.f. [Hull2000]) involves a portfolio of two stocks A and B. If we know (in \$) the daily volatility σ_A and σ_B of the stock prices, and the correlation coefficient ρ describing how they move relative to each other, (typically derived from historical data), then the P-percent VaR[2] is the value of X such that:

$$\frac{P}{100} = \frac{1}{\sigma_{AB}\sqrt{2\pi}} \int_{x=-\infty}^{x=X} e^{-x^2/2\sigma_{AB}} dx.$$

[2] here computed from a normal distribution with mean 0 and variance $\sigma_{AB} = \sigma_A^2 + \sigma_B^2 + 2\rho\sigma_A\sigma_B$

Can we do something similar for cyber?

Goal: perform similar calculations to obtain a distribution of possible $ losses over time, but now due to cyberattack:

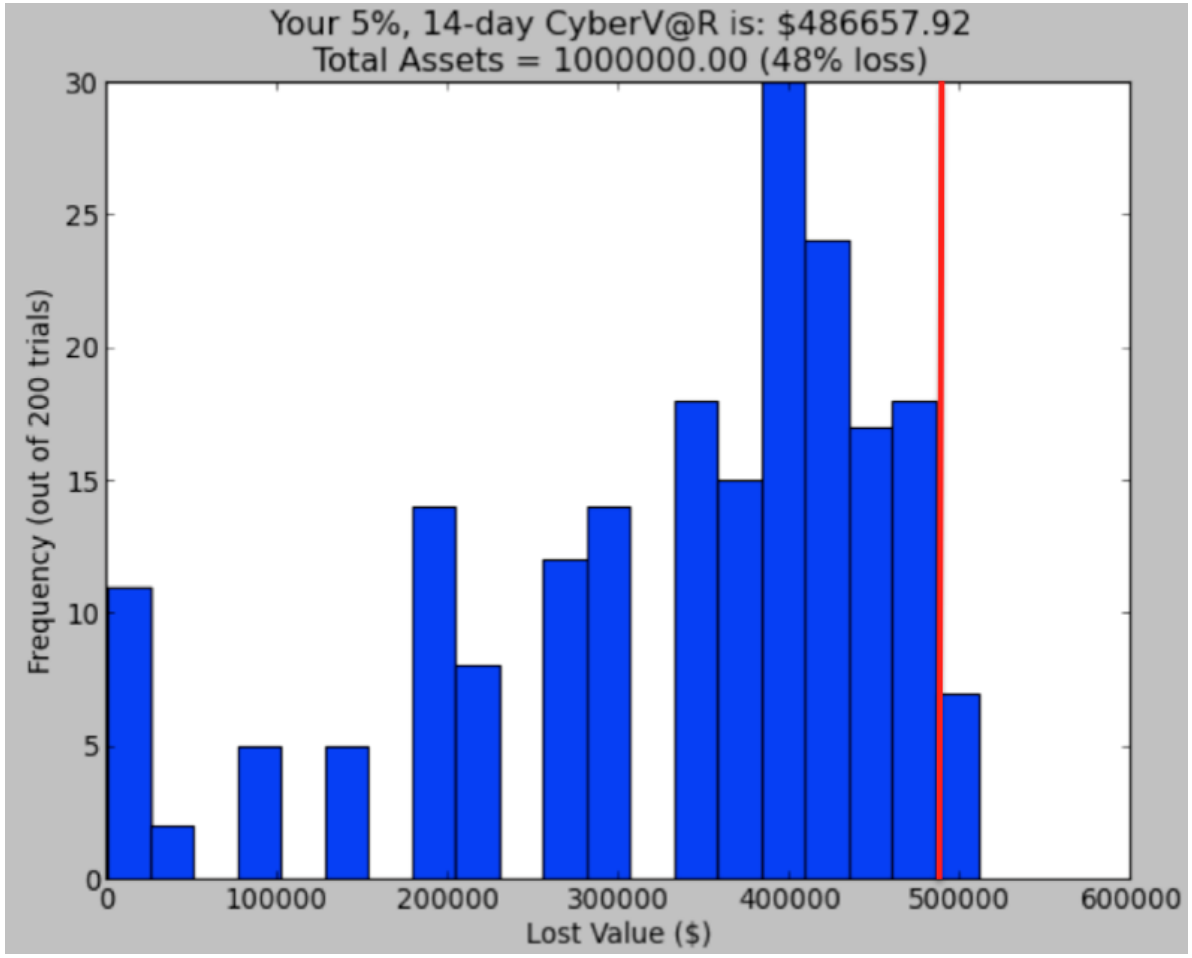


Figure 1: Loss distribution as computed by CyberV@R: red line $\approx \$X$ for P=5%. Note unlike finance example, distribution is not normal.

Yes: if we map from finance to cyber

cyberpoint

In our cyber application of the finance approach, we will make the following translations:

- Financial portfolio $\rightarrow$ networked computing infrastructure (Netflow may be a data source for this) and the assets housed there.
- Market fluctuations $\rightarrow$ threats to which the network is exposed (historical Netflow may provide this).
- Trading strategies $\rightarrow$ alternative security mitigations we may enable to reduce threats (Netflow may establish historical efficacy).
- Integration over normal distribution $\mathcal{N}(\mu, \sigma) \rightarrow$ Monte Carlo sampling over a two-slice dynamic Bayesian network [3] of attack trees (c.f. [Kol2009], [Pol2012]) representing interaction of threats, network nodes, and mitigations.

[3] a DAG B_i encoding a joint probability distribution, with a rule for transforming $B_i \rightarrow B_{i+1}$

Constructing the model (in pictures):

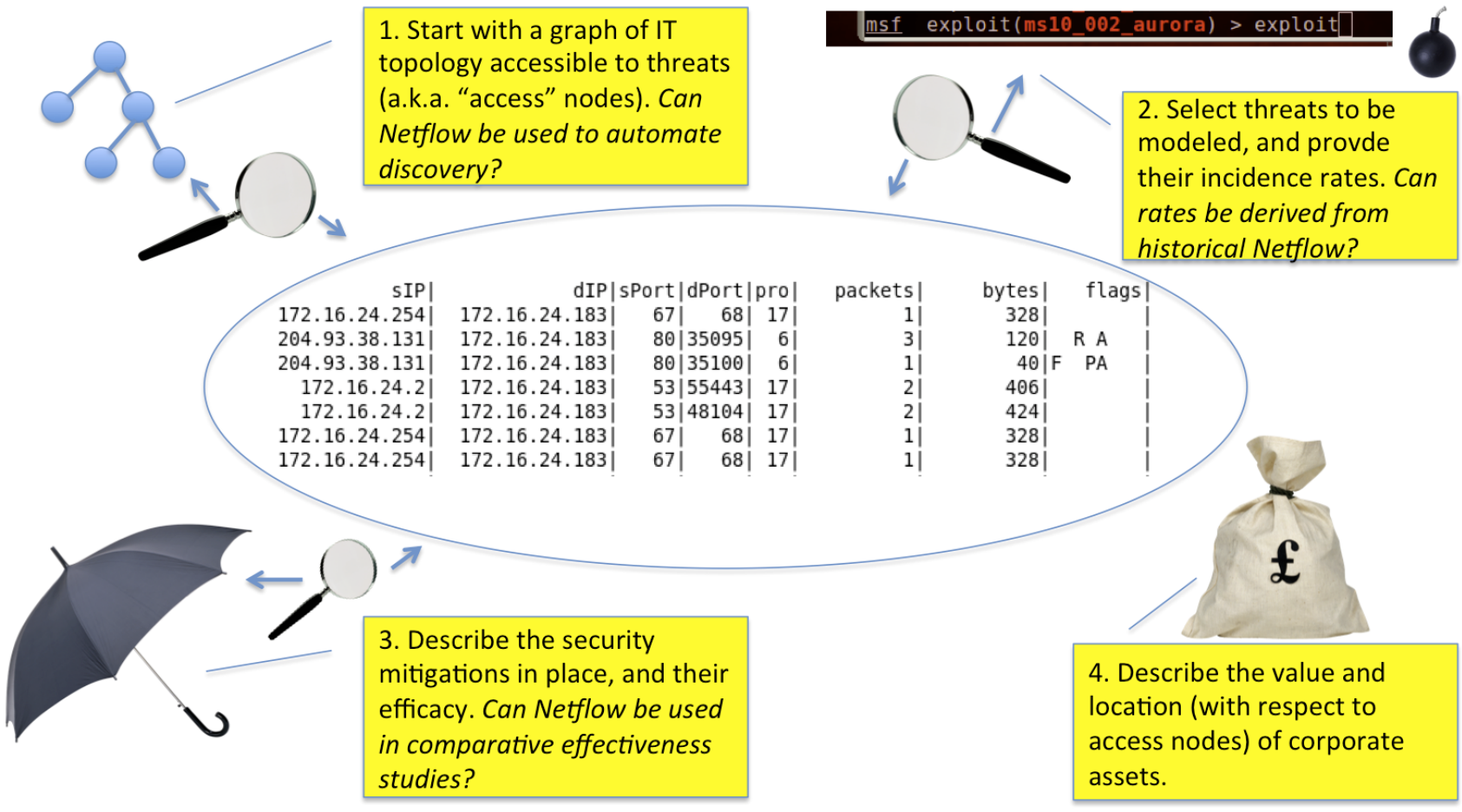


Figure 2: Model is a union of attack trees - nodes correspond to threats, security mitigations, IT infrastructure, assets of value (e.g. product designs). Each node carries a probability distribution describing its odds of being in a given state.

Constructing the model (in words):

- CyberV@R's dynamic Bayesian networks are constructed as a union of attack trees.
- Each node of each tree corresponds to a *threat stage*, a *security mitigation*, an IT element (dubbed an *access node*), or an *asset* (target of threat).
- Each node is assigned a probability distribution, conditioned on the states of its parent nodes, describing odds of the node being in a given state.[4]
- In a *trial*, the attack trees are evolved through time (via Monte Carlo sampling) to get an overall loss (value of assets reached).
- Multiple trials are conducted to produce a distribution on losses.
- The distributions are parameterized, with parameters derived empirically. Hence there is no direct training cost associated to Bayesian network construction.

[4] Threat nodes have Poisson distribution giving odds of n occurrences at any time step; mitigation nodes are Bernoulli, giving odds of thwarting any given threat stage occurrence. Access and asset nodes are two-state at each time step (reached/not reached; devalued/not devalued, respectively).

Simplest CyberV@R model (2 PCs; 1 threat)

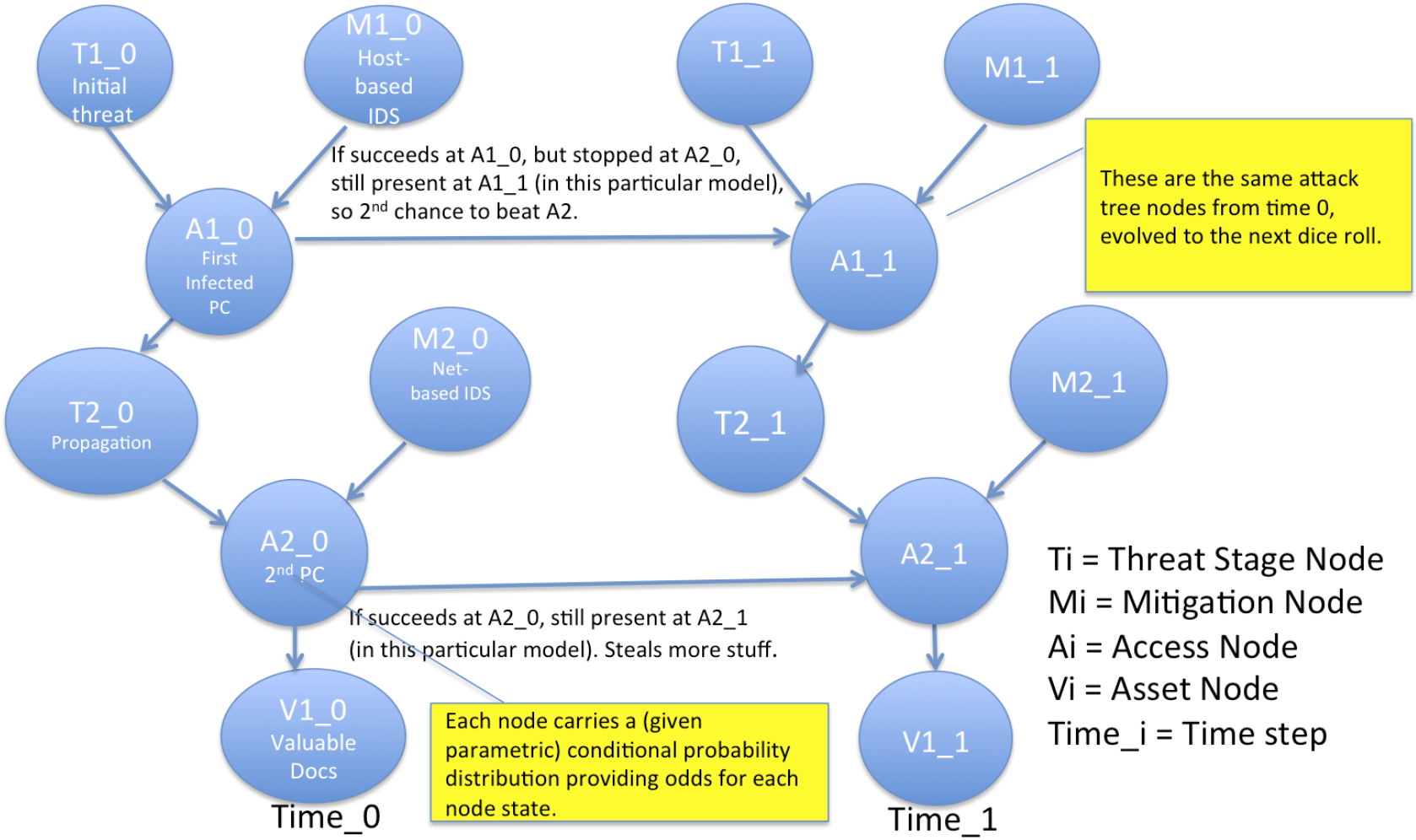


Figure 3: Time evolution of a simple CyberV@R Bayesian Network

CyberV@R in the Labs

cyberpoint

- We've constructed a CyberV@R model representing CyberPoint's internal network infrastructure at the level of routers, servers, and workstation groups ($\approx$ a dozen access nodes).
- We modeled a single threat based on Symantec's description of the Trojan.Taidoor virus (c.f. [Sym2012]).
- The model computation is implemented using CyberPoint's libPGM (see http://packages.python.org/libpgm).
- We ran the model over 100 trials, each covering a 24-month time step, in the presence and absence of hypothetical workstation software that would remove the virus if found.
- Presence of the AV software led typically to $\approx$ 35% reduction in 5% VaR.
- Computation time less than a minute.

Attack Flow for Single Threat

Trojan.taidoor exploits WinOS vulnerabilities in PDFs, such as BID47314. Drops a backdoor that attaches to services.exe, svchost.exe, communicates with a C&C server. Used to target think tanks involved in U.S./Tawain relations & policy formulation. -Symantec, 2012.

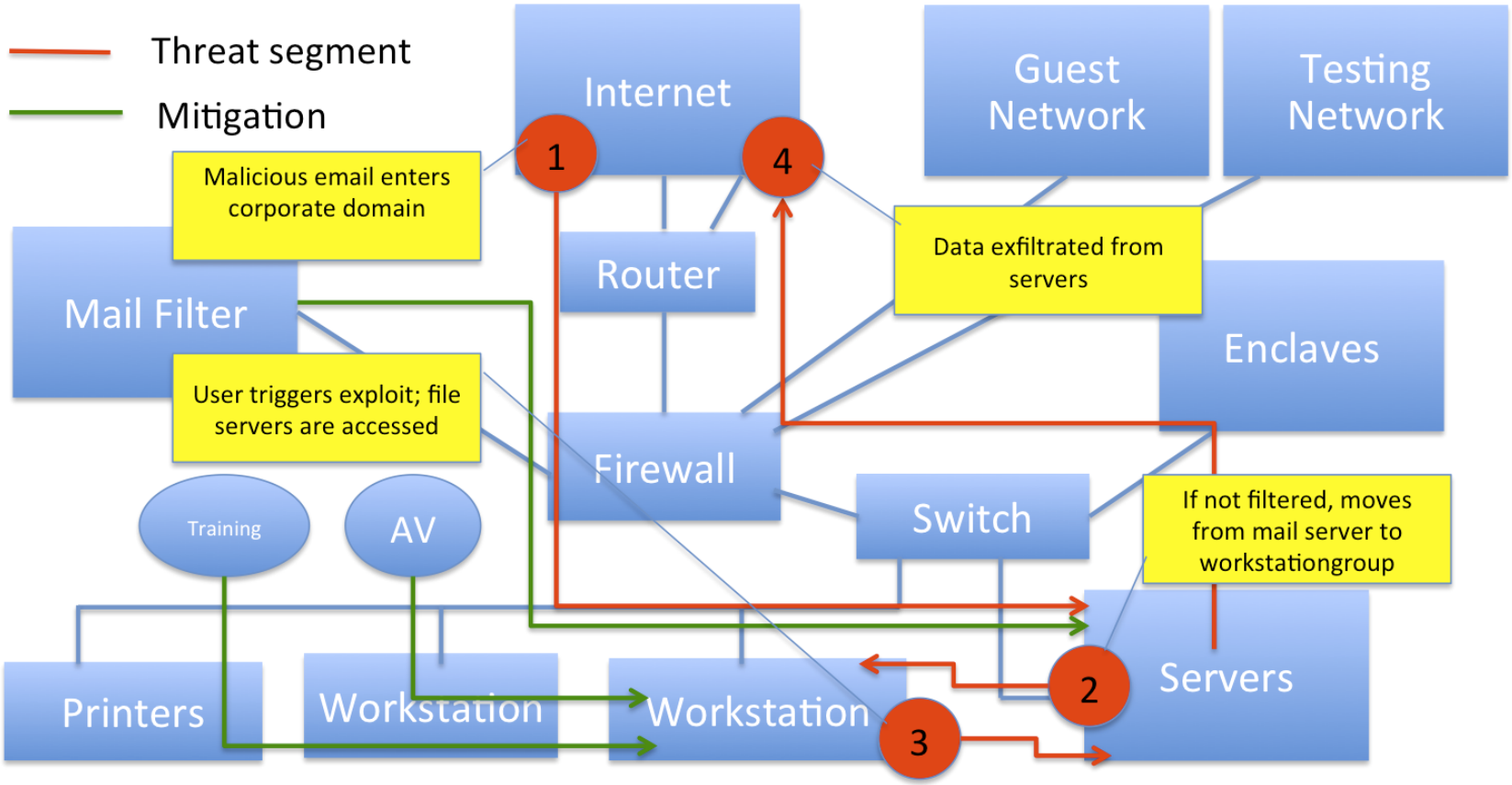


Figure 4: Attack flow of Trojan.Taidoor

Corresponding Attack Tree

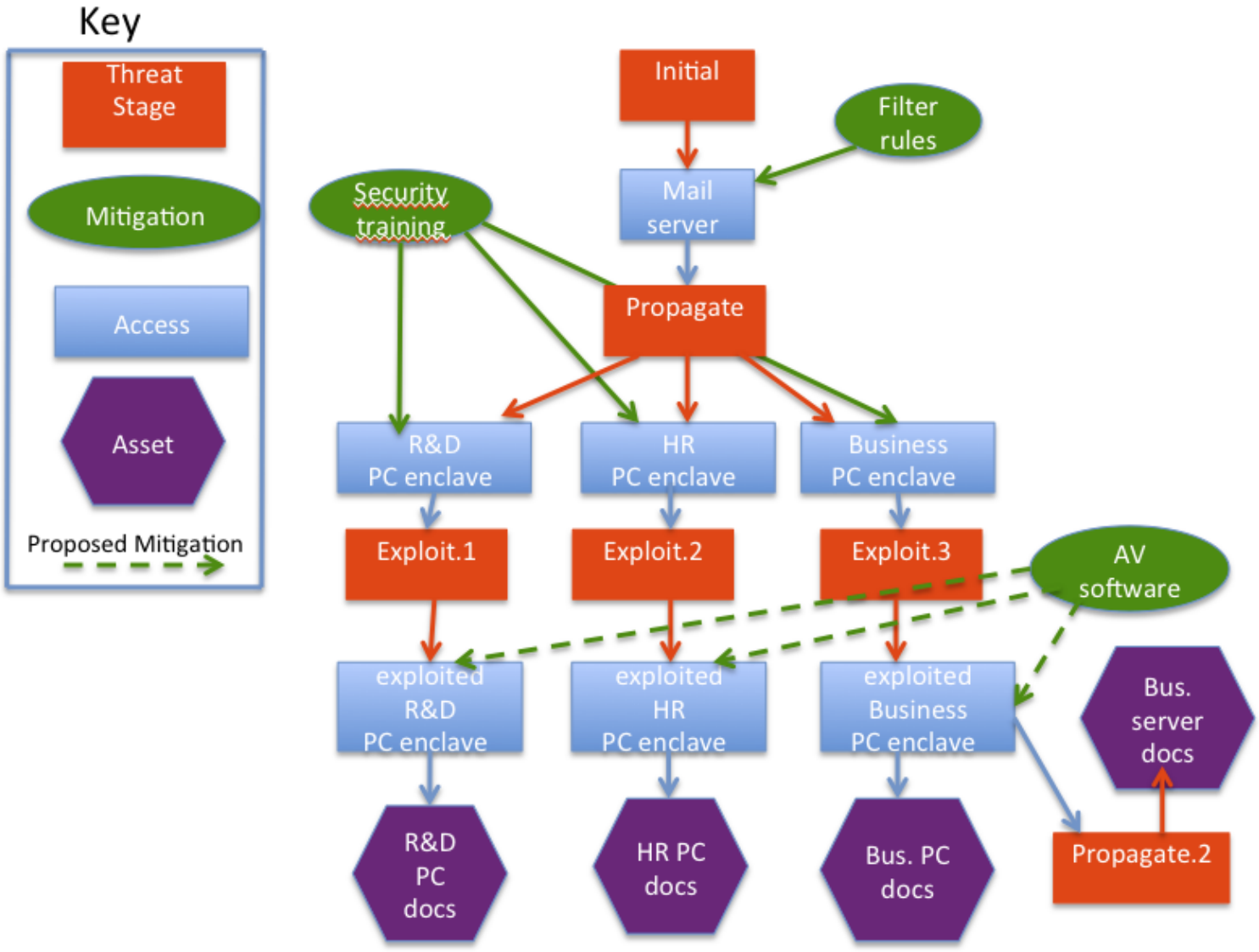


Figure 5: Partial attack tree for one time-step of evolution

Reduction in CyberV@R

We see from the graphs that the $ amount of the 5% VaR, expressed as a percentage of total projected value of intellectual property, is reduced by ≈ 37 percentage points, when virus-removing software is introduced on each workstation node (giving the virus less opportunity to spread).

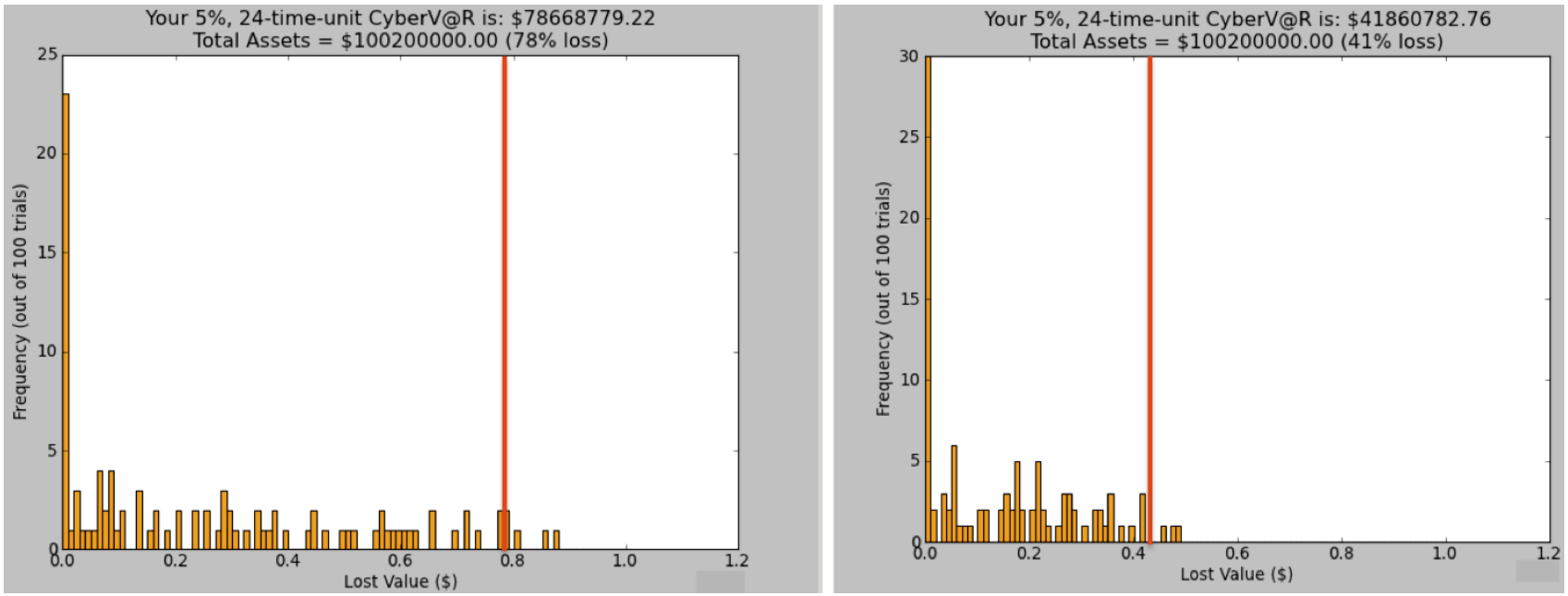


Figure 6: Computed reduction in VaR when AV added to workstations

Scaling CyberV@R

cyberpoint

- We're exploring use of Netflow and related tools to automate construction of the IT infrastructure input to the dynamic Bayesian networks.
- Historical Netflow data might be sampled and categorized with aid of visualization tools, to uncover empirical incident rates for threat types. See for example [Yin2005]. This could be automated as well.
- For organizations with 100,000s of nodes, CyberV@R computation can be deconstructed as a series of iterated MapReduce jobs. Each iteration covers one time step. The map jobs each work independently on one subnet's worth of information. A single reduce instance combines the jobs into a new Bayesian network.
- Reducer can replace sufficiently infected subnets from the computation chain with a single threat node added to each remaining peer subnet. A large network reduces to a few "last standing" subnets after several iterations.

Thanks and Questions

cyberpoint

- *I thank you for your time and attention.*
- *I also thank the FloCon 2013 organizers for the opportunity to present.*
- *Your questions and comments will be appreciated!*
- *Follow the links at www.cyberpointllc.com for the full CyberV@R technical report.*

More Details

cyberpoint

ADDITIONAL DETAIL SLIDES FOLLOW.

Proof of concept: Risk models in finance

cyberpoint

- The canonical value at risk model (c.f. [Hull2000]) involves a portfolio of stocks; say for exampe U.S. $10,000 in shares of company A and U.S. $20,000 in shares of company B.
- Say, based on historical data, the daily volatility σ_A of A's stock price is 5%, and the daily volatility σ_B of B's price is 10%. Assume also that fluctuations in stock price over a time horizon of T days are modeled as $\mathcal{N}(0, \sigma^2 T)$[5]. So the T-day standard deviation for the A holding is given by:

$$\sigma_A = 10,000 \times 0.05 \times \sqrt{T}$$

and similarly the standard deviation for B is given by:

$$\sigma_B = 20,000 \times 0.10 \times \sqrt{T}.$$

[5] a normal distribution with mean 0 and variance $\sigma^2 T$

Risk models in finance (continued)

cyberpoint

- Say ρ gives the correlation of stock price movements in A and B. Then the T-day distribution for the change in value Δp of our portfolio is given by $\mathcal{N}(0, \sigma_{AB} = \sigma_A^2 + \sigma_B^2 + 2\rho\sigma_A\sigma_B)$.
- Using this information, one can find X s.t. $P(\Delta p < X) = 0.02$, that is:

$$X \text{ s.t. } 1 - \frac{1}{\sigma_{AB}\sqrt{2\pi}} \int_{x=X}^{x=\infty} e^{-x^2/2\sigma_{AB}} dx = 0.02.$$

- We say that X is our 2% VaR (that is, any losses greater in magnitude than $|X|$ fall in the 2% tail of likelihood) . For $T = 10$ and $\rho = 0.75$, $X \approx -\$6382.00$.
- In our CyberV@R model, we will want to perform similar calculations over distributions of possible losses of intellectual property (or incurring of liabilities) over time, due to various forms of cyberattack on our organization's computing infrastructure.

CyberV@R: specification of the model

cyberpoint

A CyberV@R model is:

- A particular JSON encoding of a two time-slice dynamic Bayesian network in which each node is one of four types (threat stage, mitigation, access, and asset).
- The Bayesian network describes a union of time-evolving attack trees, one per threat type of interest.
- The edges of the network observe a set of constraints designed to model the flows of multi-stage attacks throughout the IT infrastructure.
- Each node is labelled with a conditional probability distribution; VaR is computed by Monte Carlo sampling over the joint distribution.
- All conditional probability distributions are parameterized, with parameters derived from empirical estimates passed as input to the model. Within the model itself, there is no learning cost associated to discovering / fitting the prior distributions.

CyberV@R: threat stage nodes

cyberpoint

- A *threat stage node* represents a particular stage of a particular threat, and is identified by a node id and a time index.
- The associated conditional probability distribution is Poisson: P(n attempts at executing stage at t) $= \frac{\lambda_t^n}{n!} e^{-\lambda_t}$ (this represents the odds of there being n attempts to execute the stage, between time t and $t+1$).
- A threat stage node optionally connects (upstream) to an access node (defined later), and connects downstream to an access node, having the same time index.
- In practice, mitigation nodes might be active threat types as listed by an AV provider, known to exploit certain CVEs (as listed in the National Vulnerabilities Database).
- If an organization has access to historical Netflow data, these might be mined and categorized with aid of visualization tools, to uncover empirical incident rates for threat types. See for example [Yin2005].

CyberV@R: mitigation nodes

cyberpoint

- A *mitigation node* represents a security mitigation (IPS, AV software, patch set, etc.). It is identified by a node id and a time index.
- The corresponding probability distribution will be a Bernoulli variable (independent of time) giving the odds of the mitigation thwarting any given attempt by a threat stage of type τ; e.g. P(attempt blocked) $= M$ where $0 \leq M \leq 1$.
- Mitigation nodes have outgoing edges to access nodes only (see below).
- As above, statistical analysis of Netflow data might be used to gauge effectiveness empirically by examining historical data in the presence and absence of comparable mitigations.

CyberV@R: access nodes

cyberpoint

- An *access node* represents an element of the IT infrastructure (a router, hub, server, or workstation, or cluster thereof). It is identified by a node id and a time index.
- At time t, an access node is reached by a threat stage with odds given by:

$$\mathrm{P}(\text{access}) = \sum_{n=1}^{n=\max} \frac{\lambda_t^n}{n!} e^{-\lambda_t} [1 - (1 - (1 - M_{j_1}) \cdots (1 - M_{j_N}))^n] \ ,$$

  i.e. at time t there are N mitigations in place, up to "max" threat stage execution attempts occur, and at least one gets by all the mitigations.
- An access node has as parents a single threat stage node, and zero or more mitigation nodes. It connects to a follow-on threat stage node, or an asset node (the object of the attack). Netflow data can be mined to discover these nodes.

CyberV@R: asset nodes

cyberpoint

- An *asset node* represents an aspect of the organization (intellectual property, operational continuity, absence of legal liability) that is at risk due to cyberattack.
- At time t it carries a dollar-denominated value $V_I(t)$, where I is the node id. It has access nodes for parents, and no children.
- The conditional distribution is simple: if a parent access node is reached at time t, then a fixed amount δV_I is taken from the asset node value. Otherwise the asset node value remains as it was.
- The arrangement of threat stage,mitigation, access, and asset nodes over all threat types, at an initial time point, constitutes the starting state of the Bayesian network. One evolves the network through time by sampling each node according to its distribution (always sampling parents before children).

Computing value at risk via Monte Carlo

cyberpoint

In outline form, the VaR computation then reduces to Monte Carlo sampling over the network:

Procedure:estimate P-% CyberV@R

Input: JSON-encoded Bayesian Network, # of trials N, # of time steps T, percentage P

Method:

LossArray $=$ []

Sort Bayesian Network in topological order

FOR $n = 0 \cdots, N-1$

    trialLosses $= 0$

    FOR $t = 0, \cdots, T-1$

        FOR each threat type:

            Sample each node in order, according to node's CPD

                IF asset node l is reached, trialLosses $+= \delta V_l$.

    LossArray.insert[trialLosses]

sort LossArray(ascending)

return LossArray[floor(P*N)]

Bibiography


J. Brenner, *America the Vulnerable*, Penguin Press, New York: 2011.


R. Clarke, *Cyber War: The Next Threat to National Security and What to Do About It*, HarperCollins e-books: 2010.


Executive Office of the President of the United States, *The Comprehensive National Cybersecurity Initiative*, available at http://www.whitehouse.gov/sites/default/files/cybersecurity.pdf, accessed February 9, 2012.


J. Hull, *Options, Futures, and Other Derivatives*, 4th ed, Prentice Hall, Upper Saddle River, NJ: 2000.


D. Koller and F. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge MA: 2009.


National Vulnerabilities Database, version 2.2, available at: http://nvd.nist.gov/home.cfm/


N. Poolsappasit, et. al., Dynamic Security Risk Management Using Bayesian Attack Graphs, *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 1, January/Februrary 2012.


S. Doherty, P. Krysiuk, *Trojan.Taidoor: Targeting Think Tanks*, Symantec Security Response, 2011, available at: http://www.symantec.com/security_response/whitepapers.jsp, accessed June 25, 2012.


X. Yin, et. al., "VisFlowConnect-IP: An Animated Link Analysis Tool For Visualizing Netflows," FloCon 2005.

FloCon®

Network Flow 2012: Year in Review

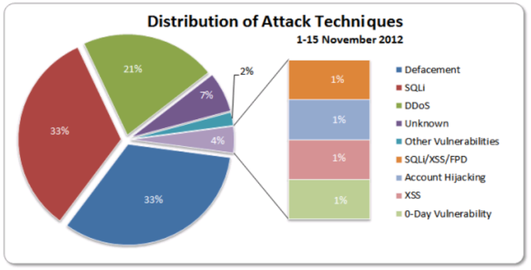
@GeoWarnagiris

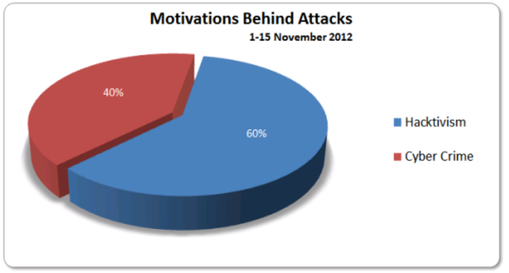
gwarnagi@cert.org

CERT

Software Engineering Institute | Carnegie Mellon

- Attacks
- Government
- Industry
- Miscellaneous





1/3 PGH makes the WashPo "In" list

1/10 USCERT warns of phishing campaign using USCERT email addresses

1/12 AlienVault releases details of Sykipot trojan variant targeting ActivIdentity smart cards

1/15 zappos.com customer details accessed

1/17 Justin Beiber's web site hacked 200 accts dumped

1/18 SOPA blackout protests

1/19 MegaUpload Seized

1/21 Several websites defaced protesting MegaUpload takedown or SOPA/PIPA

1/29 Pastebin DOS 4th time in the month

1/31 6.7GB NASA DB uploaded to pastebin

1/2 South African Postbank robbed for $6.7m

1/6 Symantec 2006 source leaked

2/2 Susan G. Komen for the Cure hacked

2/3 Conf call re: Anon between FBI and Scotland Yard leaked

2/4 403 NASA usernames and passwds posted

2/5 Syrian Ministry of Pres Affairs email server leaked

2/6 SuperBowl site creds leaked

2/8 Foxconn hacked

2/8 Westboro Baptist taken down

2/10 cia.gov taken down

2/12 Several Greek Govt sites downed

2/13 trade.gov.cn user name/pass leaked

2/14 Cryptome infected

2/14 WSJ reports Nortel has been owned for 10 years by Chinese

2/26 UN (UNEP) DB leaked

2/26 Wikileaks publishes Stratfor emails

2/27 HP Security Intelligence and Risk Management platform

3/2 Cray launches new big data appliance

3/2 $71K in bitcoins stolen from linode.com

3/7 5 members of Anon/Lulzsec arrested

3/7 Vatican web site DDoS

3/12 BBC Persian attack disclosed

3/22 MITRE publishes "Social Radar" research

3/24 Canadian New Democrat Party DDoS

3/27 Beiber's Twitter hacked

3/27 Pastebin DDoS

3/30 GlobalPayments announce breach 50k - 10m affected

3/31 PBS hacked

4/1 Utah Dept. of Heath 500K personal records

4/2 Al Qaeda DDoS

4/2 China's CEIEC sensitive US documents leaked

4/3 Sophos hacked

4/12 Another FBI/MI6 call leaked - 2 arrests

4/14 w0rmer of CabinCr3w arrested

4/16 Austria's Federal Criminal Police Office arrest 15 year old for hacking into 259 companies across the span of three months

4/16 US Gov & LE DDoS

4/23 Iran oil industry attacked

4/26 Taliban web site defaced

4/27 13M Hotmai accts breached

4/29 Yahoo and AOL email breached

5/5 US natural gas pipeline industry attacked

5/8 R.Weaver successfully defends PhD thesis

5/12 Bitconica $87k stolen

5/16 Wikileaks and Pirate Bay DDoS

5/22 1.7GB from US DOJ released

5/24 US State Dept. defaces Al Qaeda web site

5/24 Mayor of W. New York arrested for attacking critical site

5/26 U. of Nebraska 654K user details est. $127M

5/28 Flame announced

6/1 Toronto Police Dept. attacked

6/4 ca.gov admin acct creds dumped

6/5 6.5M LinkedIn passwd hashes dumped

6/6 1.5M eHarmony passwd hashes dumped

6/7 Last.fm warns 40M users to change pass

6/11 IPv6 support in Qosmos Network Intel Suite

6/13 Google and MS announce 0-day being used to exploit Gmail

6/21 Iran claims "massive" cyber attack

6/26 Cosmo and Mir of UGNazi arrested

7/3 Several Chinese web sites hacked and acct info dumped

7/5 Alaska road side construction signs hacked

7/7 Several Chinese web sites hacked and acct info dumped

7/8 1300 Israeli email addrs and passwds posted to Anon Arab website

7/17 Madi Persian info stealer revealed

7/19 Mom hacks high school to change sons grades

7/19 $80K in Nike merchandise stolen via web site bug

7/23 "AC/DC Worm" reported in Iran

7/26 Aussie ISP breached 40GB exfiled

7/31 Dropbox breached

8/2 EPA hacked

8/2 MLB Facebook hijacked by rogue employee

8/3 Mat Honan's digital life destroyed

8/7 Wikileaks DDoS

8/8-8/20 Anon #OpAustralia

8/9 Battle.net hacked

8/9 Gauss banking malware details released

8/9 Utah computerized signs hacked

8/13 Argus 3.0.7.1 released

8/15 Saudi Aramco cyber attack (shamoon)

8/16-8/31 OpFree Assange

8/21 Moscow Court web site hacked

8/27 Sourcefire releases FireAMP

8/29 LulzSec member arrested and accused of an extensive breach of the Sony Pictures Europe.

8/29 Bro 2.1 released

9/1 Pastebin DDoS

9/4 Bitfloor Bitcoin exchange hacked ~$250K

8/30 Qatari Nat. gas company infected (shamoon)

9/9 Israel arrest 3 suspected hackers

9/11 Pirate Bay founder arrested for hacking

9/12 Gartner projects 8.4% increase in IT security spending $55B 2011 $60B 2012 $86B 2016.

9/18 Mirage RAT details released

9/18 IEEE data breach

9/18 Online poker player hacked and $100K stolen

9/18 US Bank DDoSes begin

9/21 USDA defaced

9/26 Sourceforge compromised

9/26 Spain's National Police DDoS

9/26 Telvent Canada accuses Chinese hacking group of intrusion

9/27 DARPA holds "Plan X" workshop

9/28 Adobe certificate compromise

9/30 White House discloses July intrusion by actors related to the Chinese Govt

10/8 House Intel Committee warns on risk of using Huawei gear

10/7 WoW mass avatar murder

10/10 Bank DDoS continue

10/16 S. Env. Law Center hacked

10/19 GitHub DDoS

10/24 South Carolina Dept. of Revenue breach 3.6M tax returns

10/24 63 B&N keypads hacked

10/25 Israeli National network disconnects from civilian

10/30 Lancope launches the StealthWatch Labs Intelligence Center

11/4 NBC.com defaced

11/5 Guy Fawkes day many exploits

11/8 2 EU officials claim laptop hacked in hotel in Azerbaijan

11/11 FreeBSD.org unauth access

11/21 France claims US attacked Sarkozy w/a Flame variant

11/22 Symantec releases Narilam details

11/23 Some GoDaddy DNS hijacked

11/26 Nationwide owned

12/1 Mexican president cyber protests

12/3 8K Tumblr blogs spammed

12/3 Dalai Lama site distributing Mac malware

12/4 Swiss National security agency warns of large secret data leak

11/28 First Security Insurance bank account robbed

12/7 Napatec achieves 100Gbps on Dell commodity hw

12/12 CERT announces Common Sense Guide to Mitigating Insider Threats

12/12 Bank DDoSes cont

12/13 ITU treaty rejected in Dubai conference

12/17 Westboro Baptist Church targeted OpWBC

12/18 NASA announces launch of newest Tracking and Data Relay Satellite (TDRS) for Internet connectivity to Antarctica

12/19 PGH named one of the most entertaining cities in the US by CNN.com

12/19 Al Qaida websites DDoSed

12/21 Council on Foreign Relations poisoned

12/31 SecurityOnion 12.04 released

FloCon 2012

Dates: January 9 - 12, 2012
Location: Sheraton at the Capitol, Austin, Texas
Theme: The progression of analytics from ideas, to prototypes, to tools
Keynote: Martin Roesch
Training: Argus, iSiLK, Network Profiling
Topics:

- Visualization
- Data sources
- Analysis
- Augmentation
- Monitoring above the enterprise

Chairpersons: Ed Stoner, Rhiannon Weaver

FloCon 2013

Dates: January 7-10, 2012
Location: Hyatt Regency, Albuquerque, NM
Theme: Flow in the era of big data: logistics, characteristics, statistics
Keynote: Steven Bellovin
Training: Argus, SiLK, Security Onion
Topics:

- Visualization
- Data sources
- Analysis
- Augmentation
- Signature vs. anomaly detection
- Real time vs. near real time

Chairpersons: Rhiannon Weaver, George Jones, Donovan Truitt

FloCon®

Network Flow 2012: Year in Review

@GeoWarnagiris

gwarnagi@cert.org

CERT

Software Engineering Institute | Carnegie Mellon

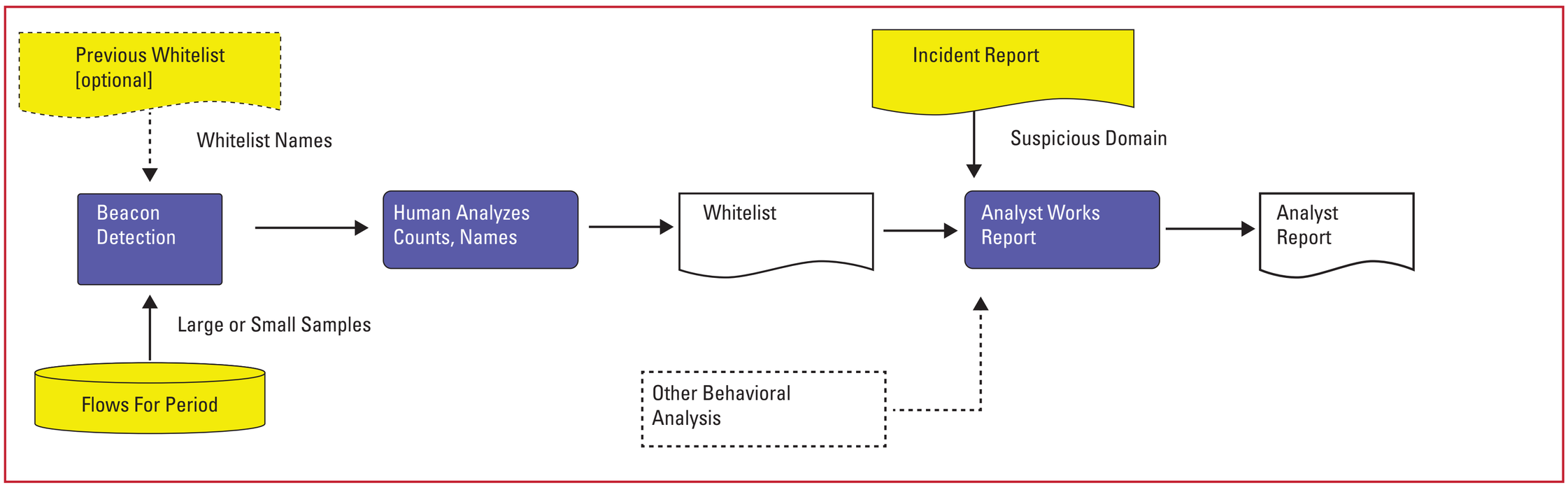
Credit to:
http://hackmageddon.com
@paulsparrows

US-CERT

UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Behavioral Whitelists of Beaconing Activity

US-CERT: Brian Allen, Robert Annand



What

- Create whitelists of beacons for use in incident analysis.

Why

Threat Discovery

- "Is something malicious on my network phoning home?"
- Which hosts on my network are 0wn3d?

Situational Awareness

- "What are the normal things that beacon on my network?" Why?
- Need to understand normal to spot abnormal.

How

Two approaches

Start small, work up

- One hour, well-known network, specific services
- Pull outbound traffic sample
- Run beacon detection programs on sample
- Create, maintain whitelists
  - Very specific, will miss things

Start big, work down

- Pull large sample
- Run beacon detection programs on sample
- Identify all beacons
- Create and maintain whitelist
  - lots of noise, false-positives

Issues

- Beacons within a single flow not visible
- Lots of beaconing over web ports
- Complete TCP connections
- Low and Slow
- Talk to asset owners: policy? What's normal?

So What?

Finding malware beacons directly

- But may still need to validate with AV, C2 server lists, etc.

Finding the normal (precursor for anomaly detection)

- NTP, AV updates, software updates, SNMP, regular data transfers. . .

This is one example of behavioral sets. Others might include

- Blacklists, High-Volume Webservers, destinations never seen before, proxies, clients, etc.

Enables analysts to ask questions like

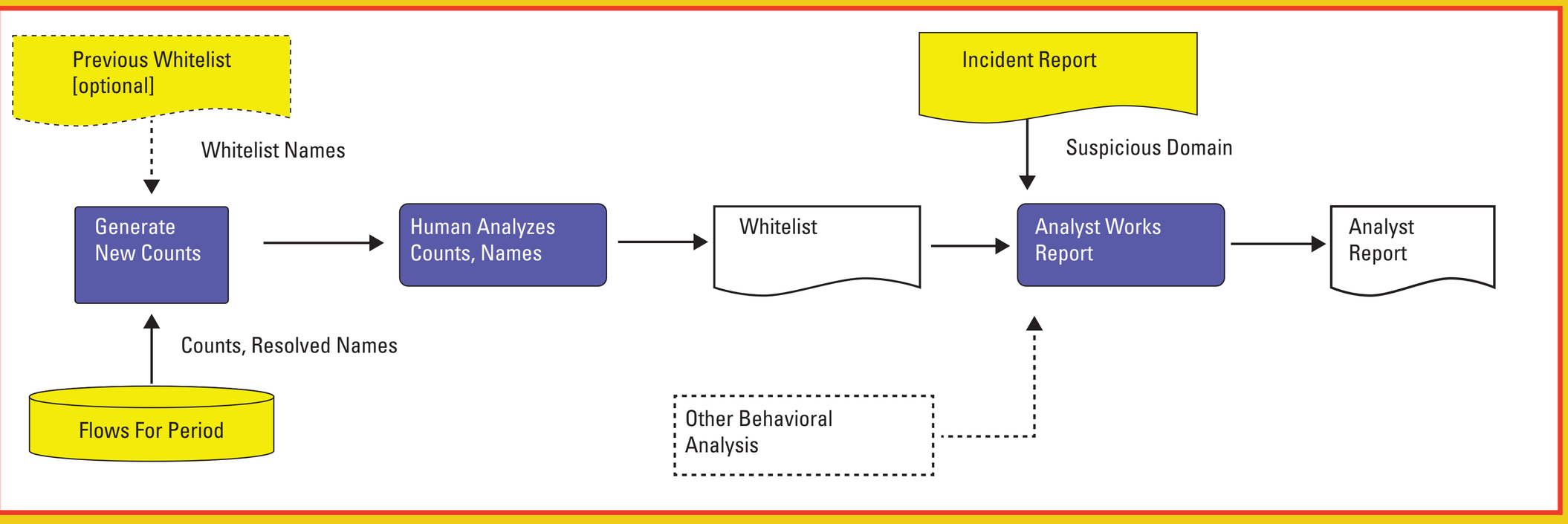
- Tell me everything I know about this destination in terms of behavior over time.

Volumes, times, services and behaviors-of-interest will vary.

FloCon®2013

Behavioral Whitelists of High Volume Web Traffic to Specific Domains

CERT: George Jones, Tim Shimeall



What

Create whitelists of external domains receiving large volumes of web traffic for use in incident analysis.

Why

Situational Awareness

- "Is that bad traffic coming from an obscure isolated IP or from GigantoProxyFarmInc?"

Sanity Checks for Security Analysts

- "Am I about to tell my constituents to block all traffic to or from MegaCloudCo?"

How

Pull outbound web traffic some period.

Count flows and bytes per destination address.

For the "large" destination

- Deduplicate and resolve domain names
- Drop unresolved addresses
- Drop large, popular sites that may actually be used for malicious purposes (e.g., don't whitelist something large that may be suspect)
- Save the results as sets (ip, name,flows|bytes)

Have a human decide which results to include in whitelists

- Rerun and maintain

So What?

- Avoid issuing embarrassing false reports
- Maintain credibility
- This is one example of behavioral sets; Others might include:
  - Blacklists, beacon destinations, destinations, never seen before, proxies, clients, etc.
- Enables analysts to ask questions like:
  - Tell me everything I know about this destination in terms of behavior over time.
- Volumes, times, services and behaviors-of-interests will vary.

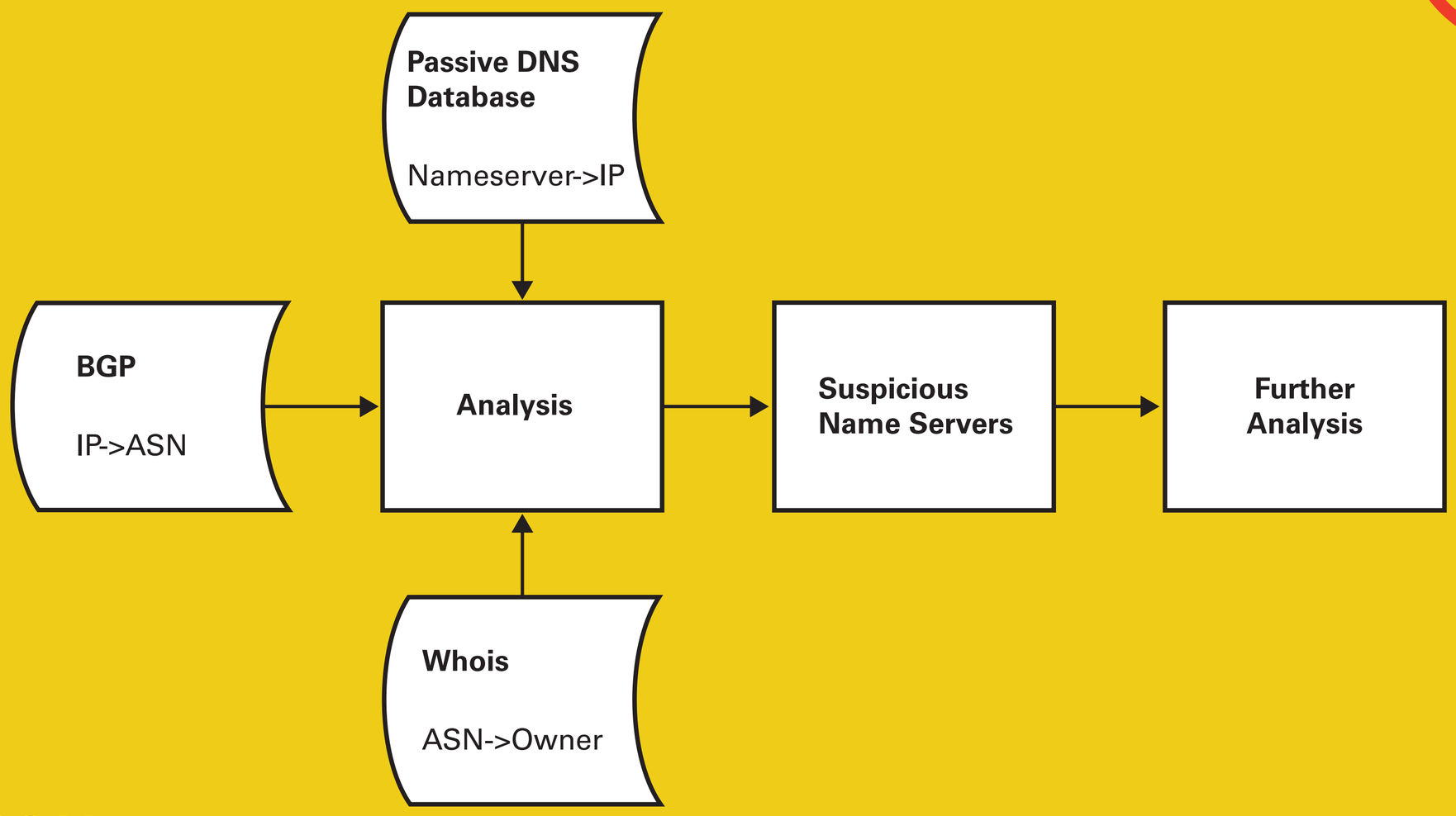
http://www.cert.org/flocon

CERT | Software Engineering Institute | Carnegie Mellon

FloCon®2013

Name Servers Should Not Move

Leigh B. Metcalf, Jonathan Spring



GOAL

Find name servers that move from IP address to IP address too much.

METHOD

Using a passive DNS Database, detect changes in IP addresses from A records three or more times in a month.

CAVEAT

Organizations may have valid use cases for moving name servers (e.g., content distribution networks [CDN]).

FILTER

Select name servers whose IPs are in different ASNs (according to BGP, using available routing data); treat ASNs as different only when they have different owners (according to WHOIS).

RESULT

Name servers that move without a valid reason are probably malicious.

http://www.cert.org/flocon

CERT | Software Engineering Institute | Carnegie Mellon