REDJACK

First experiences with Cuckoo bags

John McHugh - RedJack, LLC and The University of North Carolina

Jeff Janies - Redjack LLC

Teryl Taylor - Dalhousie University

FloCon 2010

New Orleans

January 2010

REDJACK

What is a cuckoo bag?

- SiLK sets and bags have single index field
  - chosen from subset of SiLK record fields
  - bags have single volume data field: flows, pkts, bytes
  - pointer tree implementation limits key to 32 bits
- Cuckoo bags have multiple index fields
  - all meaningful SiLK record fields plus
    - derived fields such as country code, and
    - key fields can be masked or reduced in precision
  - multiple data fields, volume, plus “span”, plus TBD
  - efficient, hash based indexing

REDJACK

Why Cuckoo?

- Cuckoo bags use multiple hash functions, so there are several places to put an object.
- If these are all full, their occupants alternates are checked and if there is a space, the occupant is kicked out to the alternate space.
  - This is likened to the European Cuckoo bird which lays its eggs in the nests of other birds, dumping one or more existing eggs.
  - The search for an entry to move is done recursively until a space is found, or we give up.

REDJACK

Give Up?

- At every level, the search expands.
  - Takes longer to find a hole
  - above about 90% table occupancy it is better to reallocate and rehash.
  - Since the new table is less than 50% full, no searching is required on the rehash
  - If you know how big the table needs to be, you can avoid searching altogether.
    - First search typically occurs at 65%+ occupancy

REDJACK

Advantages and disadvantages

- Works with IPv6 keys and multiple keys
- A set is a bag with no data
  - Can treat a bag as a set for set operations
  - Disk representation is similar to rwbags
- Key is explicitly part of memory representation
  - can require more space; depends on locality
- Constant time lookup for filter applications
  - does not grow with size as with R/B trees
  - can use multiple cores to speed hashing

What do we have?

REDJACK

- cubag program
  - like rwbag / rwset but more general

    --bag-file=<path>:<key>..<key>:<data>..<data>

    --set-file=:<path>:<key>..<key>
  - Can be repeated for multiple bags / sets
  - key fields: {s,d,nh}IP, v{4,6}{s,d,nh}IP, protocol, {s,d}Port, {s,e}Time, duration, sensor, input, output, {s,d}cc, {,initial,session}flags, attributes, application, typeclass, ICMPtypecode, IPversion, bytes, pkts
  - data fields: flows, bytes, packets, duration, span, counts
  - Times to second only
  - Span is minimum sTime, maximum eTime for key
  - Count is derived data field during projection

What else?

- Command options for `rw{set, bag}` superset
- Key modifiers
  - masking IPs and flags (&, 255.255.0.0) or (&,SAFR)
  - reduction of times (*,3600) or (*,86400)
    - hourly, daily grouping by start or end time
    - will build plugin for rwcount style binning
  - example
    - hourly volumes between /16s and hosts in a /16
    - v4sip(&,255.255.0.0),v4dip(&,0.0.255.255),sTime(/*,3600)
    - TCP Initial state flags per IP
    - v4sip,initialflags(&,SAFR)

REDJACK

cubagcat

- Simple listing of cubag
  - Count entries, describe bag
  - With or without headers (cubags are self describing)
  - epoch and clock time formats (times, duration, span)
  - zero padding of IPs, integer IPs for IPv4
  - No network structure (have to limit to IPv4, single key)
  - No binning (moves to bag tool)
  - Per field statistics

REDJACK

Example: Mixed IPv4, IPv6 Bag

sourceIP	protocol	IPVer	Flows
::	58	6	194
64.86.88.116	41	4	20
128.237.230.30	17	4	1
128.237.238.167	1	4	10
128.237.238.167	41	4	20
128.237.243.180	17	4	8
128.237.247.204	17	4	11
128.237.248.255	17	4	2
128.237.254.83	17	4	10
2001:200::8002:203:47ff:fea5:3085	58	6	1
2001:5a0:300::5	58	6	1
2001:5a0:300:100::2	58	6	1
2001:5a0:300:200::2	58	6	1

REDJACK

cubagtool (under construction)

- Everything rw{set,bag}tool does, cubagtool does better (or right)
- Additional operations for projection, binning
  - user defined field names for “count” field(s)
- Mix of unary, binary, n-ary operations
  - some unary ops combine w. others in one pass
- Stream operations allow arbitrary size growth
  - If inputs and outputs maintain sort order, memory representation of output not needed
    - set union, intersection, bag addition, subtraction

REDJACK

cubagtool hacks

- Work with text from `cubagcat`
- We need set prefix projection now
  - script to drop trailing set key fields and merge/count
- We also need set intersection and difference
  - script runs through 2 set listings, similar keys
  - 3 outputs (common to both, in first and not second, in second and not first) Could add set union, as well
- Finally, need to join bags on common key
  - output has key, selected data fields

REDJACK

Coming soon!!

- plugin for `rwfilter` that will filter flow records in the manner of the current tuples using a cuckoo set (will automatically extract the cover set of a cuckoo bag)
- `cubagbuild` to construct cuckoo sets and bags from text records.
- plugin for `cubag` for time distributed binning volume fields in the manner of `rwcount`.
- plugin for `cubag` to do sums of squares of data

REDJACK

Case studies

- We present 3 examples
  - Web activity profiling
    - looking for repeated connection patterns: host pairs, temporal regularity, consistent volumes
  - Client Server activity
    - Feeds FloVis activity viewer
  - Dark Space analysis
    - Characterizing traffic in empty network segments or the space between hosts

REDJACK

Web Profiling

- Demonstrate a clear, consistent communication pattern for a given host over a time interval.
- Patterns provide evidence:
  - Of similar activity.
  - User/process preference for external hosts
- *Note, here we only discuss the detection of the initial pattern and avoid discussion of the verification process of a candidate web profile.*

Cubags: Represent Trends

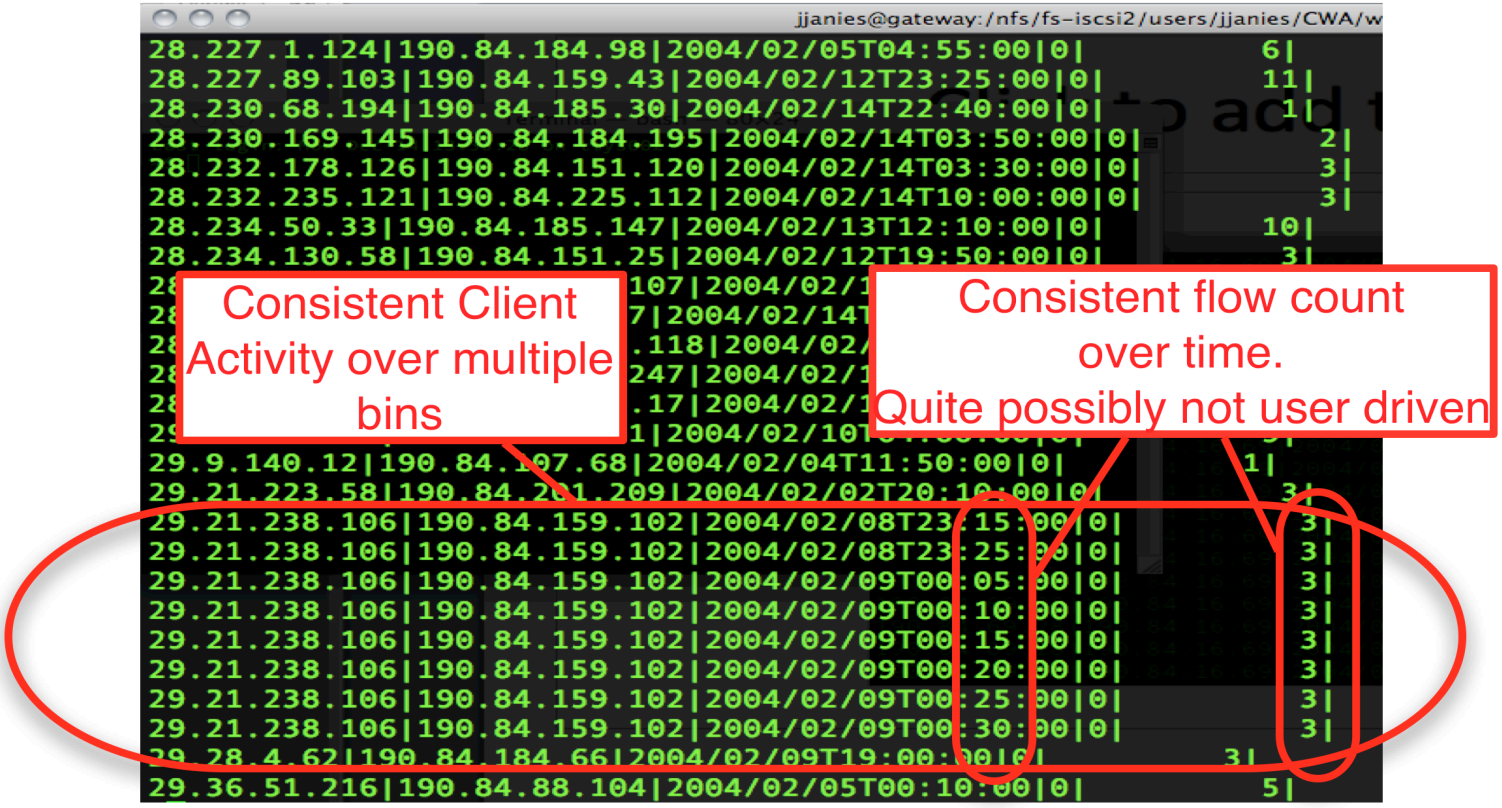
REDJACK

- Understanding common elements in client web activity.
  - Destination IP/Port
  - Intermittent/continuous
  - Size
- Trend of web client activity over time with 5 minute bins.

```
rwfilter --start=2004/02/01 --end= 2004/02/14 \
        --proto=6 --sport=1024- --dport=80,443 --pass=stdout | \
cubag --bag-file:clientActivity.cub:sip,dip,stime(/*,300):flows,bytes
```

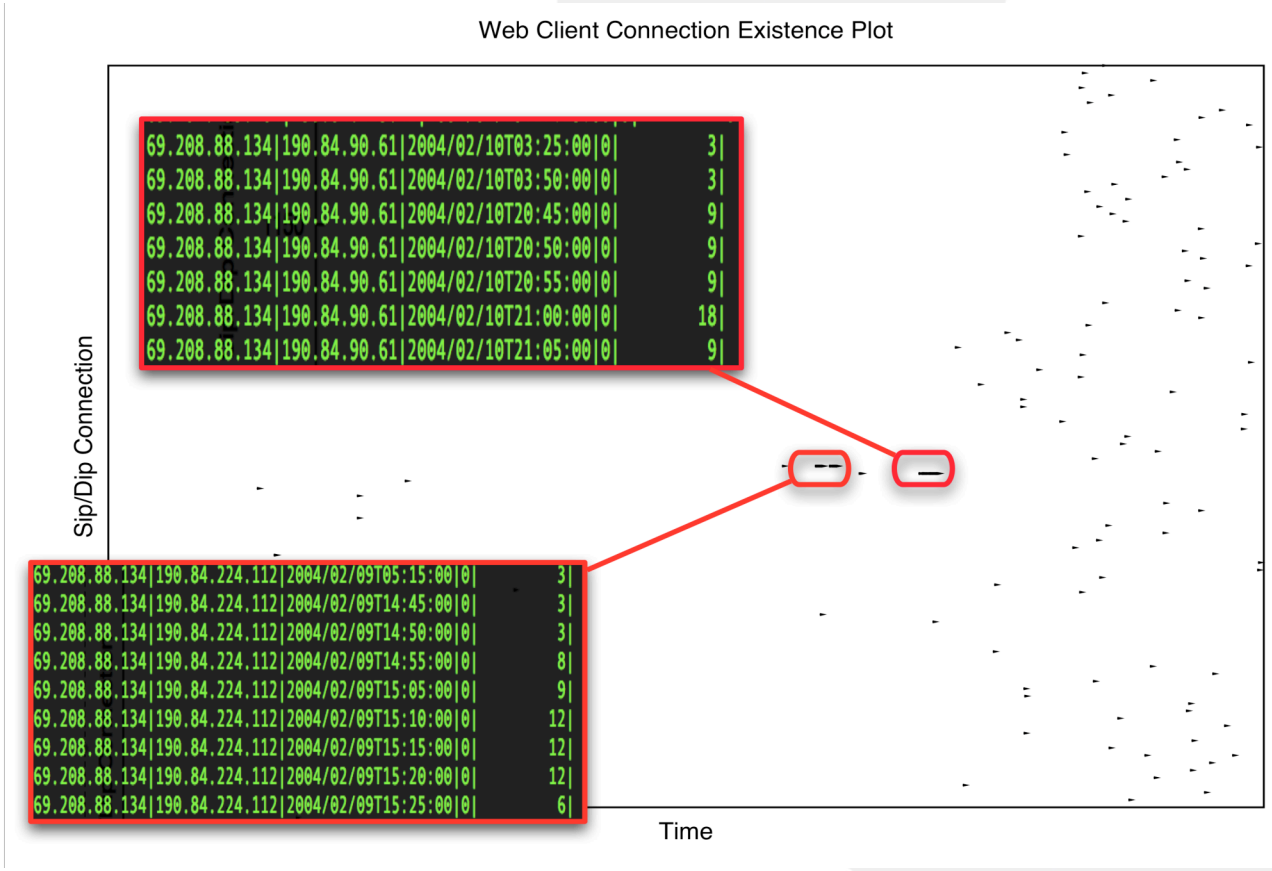
REDJACK

Cubag: Organized Raw Data with Meaning



REDJACK

Showing Consistent Patterns in Communication



REDJACK

Client / Server Characterization

- 5 categories: Idle, C, S, C/S-diff, C/S-same
  - Hosts that are client and server may be questionable
  - Look at changes over time - 1 hour bins
    - sudden changes suspicious
    - plot a week or more using FloVis Activity viewer
- Client starts conversations (TCP initial SYN)
- Server replies (TCP initial SYN/ACK)

REDJACK

Computing sets

- Client and server sets, with and without ports

```
rwfilter ... --flags-init=S/SAFR ... | \
cubag --set=cp.cus:v4sip,stime(/*,3600),dport \
        --set=c.cus:v4sip,stime(/*,3600)
```

- Server similar with SA/SAFR and sport
- Intersecting gets C/S, differencing gets C only and S only

```
cubagtool --intersect --output=cssp.cus cp.cus sp.cus
cubagtool --difference --output=cop.cus cp.cus cssp.cus
```

etc.

REDJACK

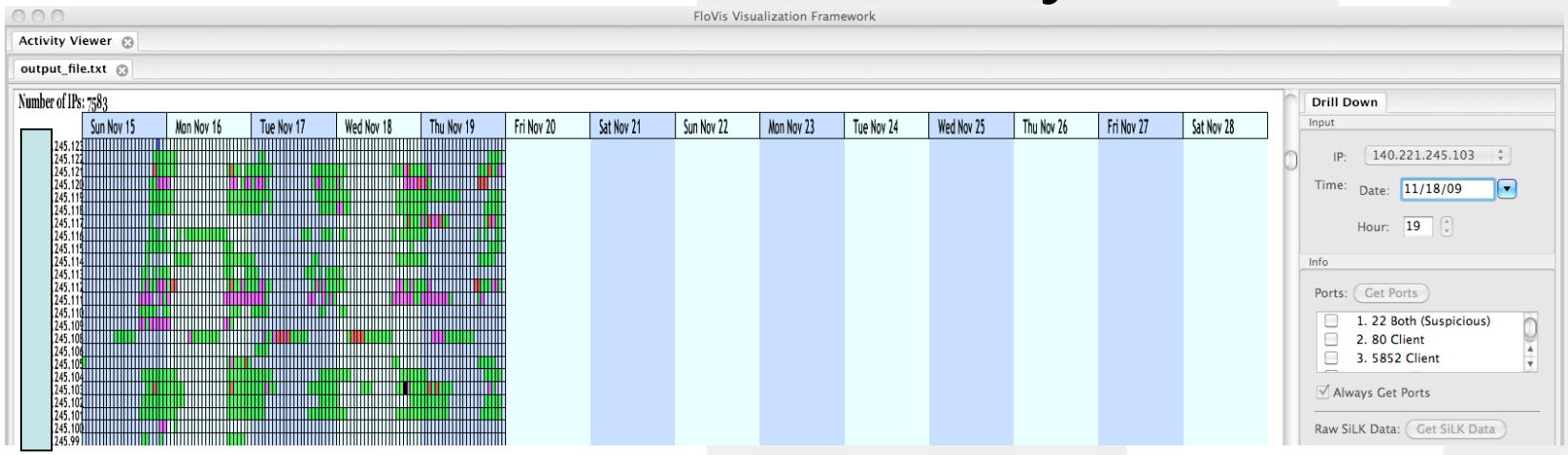
Two kinds of client / servers

- For a few services, it is normal for a host to be client and server (SMTP, DNS, etc.)
- For others, this may be suspicious
- We have sets of C, S, CS, with ports
  - the later are the CS on the same port
- We also have CS without port information
- Extract IPs from CS same port and difference with all CS to get CS on different ports

```
cubagtool --project:v4sip,stime --output=css.cus cssp.cus
cubagtool --difference --output=csd.cus cs.cus css.cus
```

REDJACK

Selected C / S activity results



What is it?

sIP	dIP	sPort	dPort	pro	pkts	bytes	initF	flags	sTime	dur
xxx.yyy.245.103	aaa.bbb.88.194	34359	22	6	725	55417	S	S PA	2009/11/18T19:28:09.845	163.961
aaa.bbb.88.194	xxx.yyy.245.103	22	34359	6	495	94839	S A	S PA	2009/11/18T19:28:09.894	163.912
ccc.ddd.118.175	xxx.yyy.245.103	15912	22	6	2	88	S	SR	2009/11/18T19:56:58.285	0.172
xxx.yyy.245.103	ccc.ddd.118.175	22	15912	6	1	48	S A	S A	2009/11/18T19:56:58.285	0.172
and later										
ccc.ddd.118.175	xxx.yyy 245.103	60076	22	6	3	132	S	S	2009/11/18T20:29:13.204	94.197
xxx.yyy.245.103	ccc.ddd.118.175	22	60076	6	8	352	S A	S A	2009/11/18T20:29:13.204	94.197

Harmless in this case, but worrisome nonetheless.

REDJACK

Dark Space

Dark space is unoccupied address space. Some organizations own large blocks of it. It is also the space between addresses in allocated space. The /22 that we observe has 117 active addresses, 899 that are dark (8 invisible). By filtering out the active addresses, we can look at the residue.

Note that the fact that there is legitimate activity in the space may provoke some of the dark space activity. Barford observed this a few years ago when he added activity to a previously dark /8. This data is from Feb. 2006 - Mar. 2007. Large scale collection failure in Aug. and Nov.

REDJACK

Who is there? What are they doing?

- TCP Scanners; Outside to dark for SYN only
  - Sets v4sIP,hour; v4sIP,dPort; v4sIP,v4dIP; dPort,v4sIP; dport,v4dIP; dPort,hour; hour,v4sIP; hour,v4dIP; hour,dPort; sCC,hour; sCC,dPort, sCC,v4dIP
  - Bags v4sIP:flows,pkts,bytes,span; dPort:f,p,b,span hour:f,p,b; sCC:f,p,b,span
- Project second field off sets; add count -> bag
- Join all bags with same key
  - gives bags by v4sIP, dPort, hour, sCC
  - counts from rollups and volumes, span
  - sort by field(s) of interest
- We present results by dPort only here

REDJACK

Dark space results - Sources/dPort

dPort	*SrcCnt*	DstCnt	Hours	Span	Flows	packets	bytes
445	*92314*	899	6609	397T15:07:29	454525	2510187	122636688
4662	*34602*	12	696	345T15:39:41	87837	261776	12980056
139	*16896*	899	4347	397T13:50:27	334937	1513998	73964832
80	*16422*	899	6065	347T05:43:03	266119	935422	45768460
1433	*13676*	899	1315	386T09:14:32	434305	1961423	94348040
9272	*12452*	1	151	7T00:31:15	22995	66551	3270628
47190	*5098*	1	18	0T16:24:17	15406	46316	2281044
34001	*4551*	1	46	2T15:18:13	18196	51396	2516076
135	*3593*	899	1550	389T20:54:06	111191	262077	12612116
14662	*3568*	1	53	3T02:39:44	7593	22776	1124904
2967	*3042*	899	506	82T02:48:06	38554	86082	4165340
23	*2962*	894	105	390T09:40:04	9077	28373	1426464
5900	*2579*	899	774	395T03:18:28	367558	1647276	79555252

REDJACK

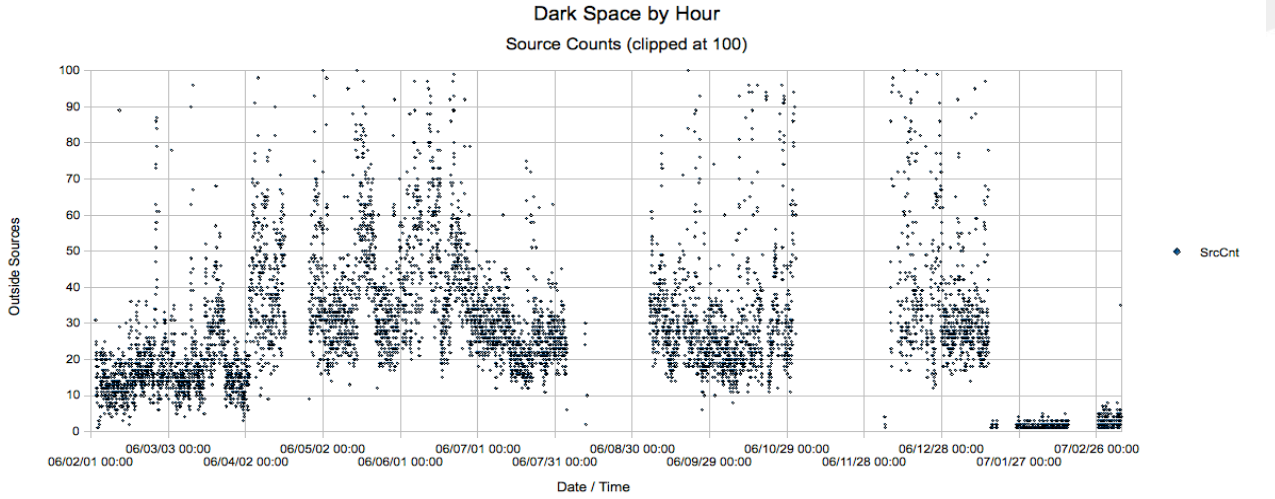
Dark space results - Hours/dPort

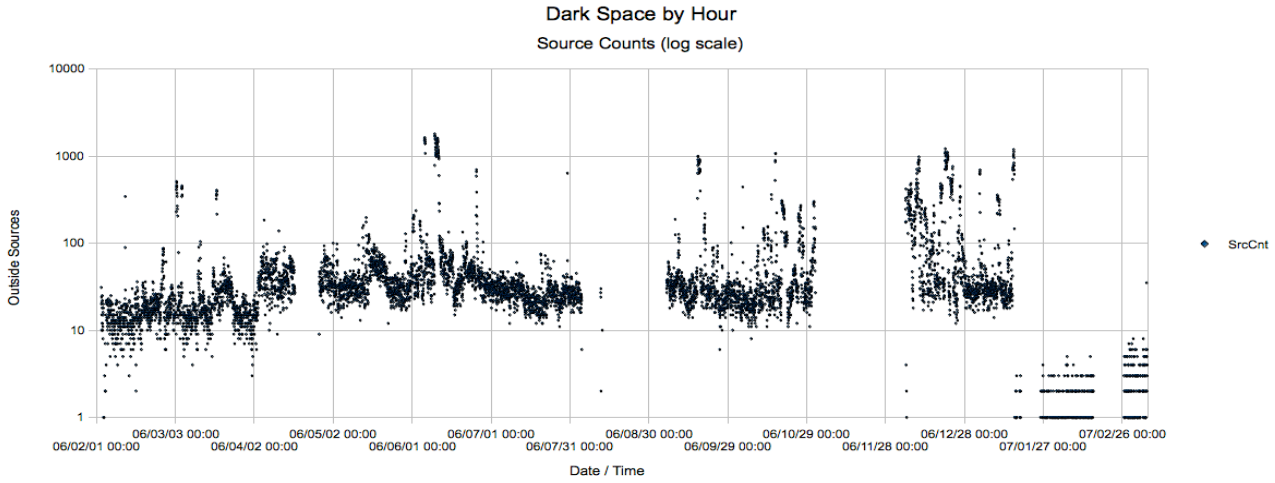
dPort	SrcCnt	DstCnt	*Hours*	Span	Flows	packets	bytes
445	92314	899	*6609*	397T15:07:29	454525	2510187	122636688
80	16422	899	*6065*	347T05:43:03	266119	935422	45768460
139	16896	899	*4347*	397T13:50:27	334937	1513998	73964832
3157	1612	1	*2683*	264T07:00:52	12206	35463	1707028
12879	1516	1	*2511*	264T06:54:34	11255	29923	1446792
1080	135	899	*2244*	376T16:22:58	19659	48549	2256972
43631	1381	1	*2200*	263T23:36:29	7396	19761	975604

Almost 1800 ports scanned. Hosts/port and hours/port vary widely
Cases with large number of sources, small number of targets unexplained. Port *du jour* effect also visible with short span.
SrcCnt = 100 at rank 54, 10 at rank 138, 1 at rank 540
DstCnt = 100 at rank 136, 10 at rank 171, 1 at rank 329
Hours = 100 at rank 62, 10 at rank 179, 1 at rank 661
Most activity is low frequency in some or all dimensions!

REDJACK

Dark Space Sources per hour





Sources per hour are in the 10-100 range most of the time

Bursts of up to about 2000 sources per hour occur irregularly and persist for several hours.

Conclusions

REDJACK

- Multikey sets and bags support complex analysis tasks
  - Time keys simplify multiperiod analysis
    - Eliminate false zeros of `rwcount`
  - Rich key set allows (almost) arbitrary viewpoints
  - Projection and join bring disparate sources together
  - Simple scripts can serve as report generation drivers
  - Stream operations remove memory size based limits
- Predictable space / time performance for real time
  - Can take advantage of multicore processors
  - Constant time lookup for filter applications
  - Arbitrary key fields
  - Adaptable to packet or other streams

REDJACK

Acknowledgements

- Funding, in part via a grant from Cisco Systems to the University of North Carolina
  - Thanks especially to Henry Stern of Ironport
- The FloVis project at Dalhousie for providing applications and data
- Ron McLeod of TARA
- RedJack for encouragement, support, and ongoing interest.
  - Greg Virgin, Michael Collins, Doug Creager

REDJACK

Questions?