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