Detecting Long Flows

John M^{c}Hugh

RedJack, LLC

John dot McHugh at RedJack dot com

FloCon 2011, Salt Lake City

January 2011

The problem

- A small number of observed flows persist for days, weeks or months. These are interesting because they represent persistent communications that may account for substantial volumes of traffic. From an analysis standpoint, such connections can be analyzed once to determine whether or not the activity involved is malicious or benign. The malicious activity should be easily actionable, and the benign activity can be whitelisted, eliminating the need for subsequent analysis while it persists.

Origins

- We started with the problem of small flows (a few short packets per flow) that were not classifiable as scans.
- This led to *keep-alives* which led to long flows.
- The motivation for extending the keep-alive work to the current long flow detection scheme came, in part, from conversations with John Heidemann at the DHS Predict PI meeting in July 2010.

  See *On the Characteristics and Reasons of Long-lived Internet Flows*, by Lin Quan and John Heidemann in the proceedings of the 2010 Internet Measurements Conference

Some definitions

- A unidirectional connection is defined by either
  - a triple of source, destination address, and protocol,
  - for ICMP a 5-tuple with message and code added to the triple or,
  - for TCP and UDP connections, a 5-tuple with source, and destination port added to the triple.
- A long connection is defined as a unidirectional connection that
  - persists for a minimum time that exceeds an arbitrary threshold – say a day or a week
  - with no lapses in activity that exceed an arbitrary gap period – say an hour or two.

The approach

- Analyze segments of data with start times covering intervals equal to or less than the maximum gap
  - Any flow beginning in one interval can continue in the next interval.
- Start with an interval
  - Build table indexed by connection with earliest start, latest end times from flow records
- For additional intervals
  - Add new connections
  - Extend existing connections
  - Discard connections with excessive gaps
    - Archive long discards

The final result

- At the end, you get a table of long connections.
  - Long discards from the entire analysis period
  - Long flows that are still active at the end of the analysis period.
- If we were feeding a “long flow” database in real time, we would perform the following for each analysis interval
  - Enter new long flows in the database as they are recognized
  - Update entries for continuing long flows
  - Mark expired long flows as no longer active.

It’s mostly done with cubags

- The cubag is an extension of the usual SiLK bags and sets to tables with multiple key and data fields
  - Most SiLK data fields can be used as a key fields
  - Volume parameters include flows, packets, bytes, and “span”
    - span is a pair of Epoch times for earliest start and latest end times associated with a given key.

Preparing the data

- We start with hourly cubags – key; data

  sIP, dIP, proto, sPort, dPort; flows, pkts, bytes, span

```
rwfilter    --start-time=${Y}/${M}/${D}T${h} \
            --proto=0-255 --type=all --pass=stdout | \
cubag       --bag-file=${Y}_${M}_${D}_${h}.cub:\
                      v4sIP,v4dIP,protocol,sport,dport: \
                      span,flows,pkts,bytes: \
                      16 \
            --warnings=noprint,zero stdin
```

Processing the bags

- We work with 4 `cubag` files, each having the same format.
  - – Cumulative flows - flows carried forward from the the previous interval
  - – Current flows – Flows originating in the current interval.
  - – Archived long flows – Long flows that expire in the current interval are added to this file
  - – New cumulative flows – Flows that start in this interval or that started in a previous interval and could be continued in the next interval

The algorithm

1. Add the current and cumulative bags
   - keys are a union of source bag keys
   - volumes add as expected
   - adding spans is a min start, max end operation
2. Remove entries whose span end is less than the start of this interval
   - Add any removed entries whose duration satisfies “long” to the archive.
     - Disambiguate archive by adding span start as key field
3. Carry the retained entries forward as the cumulative input for the next interval

The implementation

- At the time the results were obtained the cubagtool program was under construction. Hourly flow bags were produced using the rwfilter and cubag commands described earlier. The bags were processed using a program written in snobol 4 that implements the algorithm
- Step 1) could be done with the current cubagtool
- Steps 2) and 3) require an enhancement to allow operations on the start / end times of span fields

Results

- We processed data from June and July of 2006 for data from a /22 network.
- For this run, we defined
  - “long” as a day (1440 minutes)
  - “gap” as an hour (60 minutes)
- Time to process ranges from a few seconds per hour to a few 10s of seconds per hour depending on the number of connections originating and being carried forward.
- The next few slides show the hourly behaviors

```
========================================
Current time is Fri 23 Jul 2010 13:39:45 EDT
Processing data for 2006/06/19T02:00:00
Normal end of processing.
   482158 new records processed.
   404805 cumulative records processed.
   482388 records in the new cumulative file
        439871 copied directly from new file
        230 cumulative records retained on span end time.
   42287 records merged from cumulative and new file
        1444 cumulative long connection merged records.
        59 reached long threshold in this run.
   362288 cumulative records expired due to excessive gap
        81 long connection records expired.
========================================
Current time is Fri 23 Jul 2010 13:42:22 EDT
```

Discussion

- The new cumulative file mostly from current interval,
  - Most likely to expire during the next interval.
- 19 days processed, about 1400 active long connections
- About the same number of long records expire during a given hour as reach the long status.
- A look at the file of expired long connections at this point showed about 10,000 connections, most a little over a day long.
- Only seven of the expired connections were over 10 days in duration at that point.

```
Current time is Fri 23 Jul 2010 14:37:39 EDT
Processing data for 2006/07/01T14:00:00
Normal end of processing.
8646 new records processed
7498 cumulative records processed.
8651 records in the new cumulative file
    6093 copied directly from new file
    5 cumulative records retained on span end time.
    2553 records merged from cumulative and new file
           142 cumulative long connection merged records.
           0 reached long threshold in this run.
    4940 cumulative records expired due to excessive gap
           4 long connection records expired.
========================================
Current time is Fri 23 Jul 2010 14:37:42 EDT
```

Discussion

- July 1 is a national holiday at the collection location
- Vast majority of the long connections expired in the period leading up to this snapshot.
- At this point, there were about 18,000 discarded long connections
  - longest being over 20 days.

More results

- A second run was made over the same data.
  - “long” was defined as a week (10080 minutes)
  - “gap” was defined as 2 hours (120 minutes)
- Using hourly bags, approximately twice as many flows were carried from hour to hour
  - Processing time per hour increased
- The final discards file contained 632 long flows
  - Mix of TCP (8), UDP (450), ICMP (157), ESP (17)
  - Selected results on the following slides

TCP results

Src	Dst	sP	dP	Span	Flows	Pkts	Bytes
E1	O1	445	3763	2006/06/19T19:47:51-20T10:33:46	13440	13630	559719
O1	E1	3763	445	2006/06/19T19:47:51-20T10:31:26	13440	13449	541097
E2	O2	13868	3101	2006/06/07T13:50:57-11T08:29:09	16890	39315	2925559
O2	E2	3101	13868	2006/06/07T13:50:57-11T08:23:24	16895	25716	1662597
E2	O2	13872	3101	2006/07/21T22:16:54-10T01:43:05	15163	36565	2970362
O2	E2	3101	13872	2006/07/21T22:16:54-10T01:43:05	15179	25377	1692514
E2	O3	13884	3101	2006/06/24T09:51:01-23T06:00:22	34843	85913	7886791
O3	E2	3101	13884	2006/06/24T09:51:01-23T05:59:03	34869	59935	4115263

Inside and outside addresses replaced with En and On

Span is of the form Start - Duration "T" separates date and time

Flow rates in the 1-3 flows per minute range

Selected UDP Results

Src	Dst	sP	dP	Span	Flows	Pkts	Bytes
E4	04	53	53	2006/06/01T00:05:54-35T05:21:45	2128	2143	139393
04	E4	53	53	2006/06/01T00:05:54-20T00:30:32	1203	1213	235605
E5	05	123	123	2006/06/01T00:01:35-60T23:57:25	19546	19546	1485496
05	E5	123	123	2006/06/01T00:01:35-60T16:10:17	18688	18688	1420288
E6	06	4672	4012	2006/07/24T07:04:54-07T16:49:15	177	177	9735
06	E6	4012	4672	2006/07/24T07:04:55-07T16:49:16	179	179	9845
E7	07	2051	5060	2006/06/01T00:00:25-18T19:29:26	52737	138882	85330427
07	E7	5060	2051	2006/06/01T00:00:25-18T19:29:26	52740	140628	72716078
E7	07	2051	5060	2006/06/21T18:16:22-40T05:42:02	13295	38628	25513661
07	E7	5060	2051	2006/06/21T18:16:22-40T05:42:02	13295	30867	16887838

Note that several connections operate in the 1-3 flows per hour range

Selected ICMP Results

Src	Dst	Msg	Code	Span	Flows	Pkts	Bytes
E8	O8	8	0	2006/06/01T00:00:11-48T23:22:57	70283	352860	29611975
O8	E8	0	0	2006/06/01T00:00:11-48T23:22:57	70275	352567	29588613
E8	O9	8	0	2006/06/01T00:00:11-48T23:22:57	70300	351649	29538420
O9	E8	0	0	2006/06/01T00:00:11-48T23:22:57	70301	351493	29525316
E8	Oa	8	0	2006/06/01T00:00:13-48T23:22:55	70080	365793	30396441
Oa	E8	0	0	2006/06/01T00:00:13-48T23:22:55	70073	365190	30346113
Ob	Ea	8	0	2006/06/01T00:00:40-60T23:58:52	40349	40350	2098200
Ea	Ob	0	0	2006/06/01T00:00:40-60T23:58:52	40348	40349	2098148
Ob	Eb	8	0	2006/06/01T00:00:40-60T23:57:31	40381	40382	2099864
Eb	Ob	0	0	2006/06/01T00:00:40-60T23:57:31	40331	40332	2097264

Note that E8 and Ob are the pingers, O8, O9, Oa, Ea, and Eb respond.

Selected ESP Results

Src	Dst	Span	Flows	Pkts	Bytes
E8	Oc	2006/06/01T00:00:43-61T00:09:24	3,079	8,303,449	1,293,880,931
Oc	E8	2006/06/01T00:00:45-61T00:09:17	3,257	7,332,752	1,349,614,428
E8	Od	2006/06/01T00:08:54-61T00:12:03	3,043	2,009,201	294,115,345
Od	E8	2006/06/01T00:08:51-61T00:12:05	3,052	2,003,439	293,250,968
Ec	Oe	2006/06/26T22:56:21-35T01:03:15	51,728	1,627,288	1,114,832,430
Oe	Ec	2006/06/26T22:56:21-22T12:13:02	37,216	1,150,178	267,872,172
Oe	Ec	2006/07/21T23:09:02-10T00:50:34	12,045	353,625	78,122,493

E8, one of the pingers, is also a heavy user of ESP (protocol 50)

The Oe-Ec tunnel direction has a gap in service. It appears that a gap of > 2 hours appeared on July 18. A long connection was reestablished on July 21 and lasted through the end of the analysis period. There may have been shorter connection(s) during the gap. The Ec-Oe portion of the tunnel was continuous from June 26 through the end of the analysis.

REDJACK

Future work

- Plugin for cubagtool to do the calculations and discards (probably faster than snobol program)
- Prefilter TCP data to remove complete connections reducing the carry forward load
- Treat ICMP separately to capture ping / ping response (done after the fact this time)
- Adapt for continuous data streams
  - Long connection database
- Consider filtering to remove flows targeting unoccupied addresses.
  - Downside: Misses persistent connection attempts

Conclusions

- We have developed a simple and efficient mechanism for identifying persistent connections in internet data.
- The technique can be tailored for arbitrary definitions of persistence and acceptable lapses in communication
- Although persistent connections are few in number, they often account for significant data transfers and should be considered as part of a broader traffic classification process

REDJACK

A flake by any other name ...