

Flow Analysis in a Wireless Environment with short DHCP Leases

Sanket Parikh

John McHugh

Dept. of Computer Science

Dalhousie University

Project Objectives

- Analysis of Wireless Network Data from University of Dartmouth (Crawdad Archive)
- Adding MAC Layer information in Net Flow tools for identification of nodes and Activities performed by a node.
- Return converted flow data to the Crawdad archive.

Project Rationale

- The main issue in analyzing wireless network data from many environments is the assignment of temporary IP Addresses using DHCP with short leases.
- The total user population often exceeds the available address space, and a given user may connect to the network for short sessions from a number of different locations making complicating per platform analyses.
- Work to date has concentrated on mobility rather than platform behaviour.

The Data

- 160 GB of compressed tcpdump packet headers.
- Collected continuously from 2 Nov 04 - 28 Feb 04
- 18 collection points academic, library, residence
- Nothing beyond IP Headers except TCP ports and flags, UDP ports.
- Anonymized with prefix preserving technique
 - Usage agreement precludes attacking anonymization to determine user identity.
 - Low order 24 bits of MAC also anonymized
 - List of known wireless MAC addresses provided

Technical Approach - 1

- Tried to use vlan tag fields to avoid altering YAF record format.
- Use the Forward and Reverse vlan tag fields to get source and destination MAC addresses into the yafscii
- Since these are 16 bits use perfect hash of MAC
- Problems:
 - vlan tag is in unidirectional extension of flow. Need both, even for unidirectional flows.
 - would like to use with real time and when MAC set not completely known

Technical Approach

- We added MAC to the bidirectional flow root in yaf, with both source and destination MAC addresses.
- There are a number of subtleties here, including the use of memcpy that introduces field order dependencies (an IPv4 optimization) and the assumption that MAC flag implies vlanid not zero.
- Once the MAC addresses are into the yafscii output, we started converting it into SiLK for further data analysis
- Shortly after we finished, CERT added MAC address support to YAF and we will use it in the future.

Technical Approach

- We created a module *yafscii2tuc.c*
 - Inserts minimal perfect hash index of MAC in in / out
 - Adds sensor id from command line to identify the sniffers.
- We split the output of the *yafscii2tuc* into separate hourly streams and use *popen* to send each one to a separate invocation of *rwtuc* so that the resulting files are in a proper date hierarchy.
- We also use *rwsort* on the *rwtuc* output to ensure time order and because *rwtuc* does not compress.

Minimal perfect hashes

- A Minimal Perfect Hash maps a set of N unique strings into integers in $[0..N-1]$
 - Packages available on internet designed for null terminated strings
 - Modified for counted strings
 - Extracted all MACS from Dartmouth packet data
 - Grouped to bring common usages together, e.g. known wireless, gateways, etc. then created MPH
 - 17000+ MACs, 11,000+ with IP packets.
- Lookup is constant time, collision free

MAC types

- There are 5 categories of MACS actively involved
 - Known Wireless MACs with IP traffic
 - Other MACs with IP packets
 - Multi cast MACs
 - Gateway MACs
 - Broadcast MACs
- A large number of MACs have no IP traffic
 - Some appear only at link layer, others in MAC list but not seen
- We used rfilter to build sets for each type of MAC address based on the input and output field values

Project Outcomes

- We found some interesting information during analysis of the datasets. There are traces which shows some IP addresses appeared in two different sniffers located to different locations.
- The reason may be the physical location of sniffers for collecting data. Though sniffers were not located at proper distance from each other, there might be the chances for getting same IP traces in two different sniffers.
- This seems improbable and needs further study

Remaining problems

- yaf does not deal with decreasing time well
 - In live capture, packets are always in increasing time order no matter what the clock says
 - In playback the same holds unless the file has been reordered.
 - Several Dartmouth sensors exhibit decreasing time, probably due to ntp or other clock adjustments.
- Data from one of the sensors “breaks” the pipe
 - This may be related to the time problem above or may be due to another problem
 - Truncated packets may lead to other pathologies in yaf

Next steps

- We want to reassign the IPs currently used to a consistent IP that is related to the MAC index.
- First we need to determine if any wireless IPs are associated with gateway MACs.
 - This would occur if a wireless unit talked to another wireless unit via a routed connection, e.g. units connecting via separate sniffers.
 - Start by creating sets for each MAC type and looking for intersections
 - May have to explore DHCP strategy in more detail.
- This is currently underway.

Next Steps

- With the technique we used for this research should prove useful for similar data from wireless “hot spots”, airport, hotels and convention center networks and more.
- Same approach can be used to analyze data by using MAC layer information in Flow Analysis tools to identify the activities and movements of nodes in Wireless Networks.

???