

Capstone Project Proposal

Introduction to flow Data Science by Springboard

Network Data Flow analysis

By Gabriel Fontenot

Introduction

The problem statement

Network engineers have been collecting flow data for decades regarding performance and behavior. The flow data sets collected include but not limited to device configurations, syslog (device messaging), and device snmptraps (fault management). Engineers and operations have for the most part used the collected flow data to assist with various troubleshooting and design exercises.

Using the statistical, probability and machine learning modules, the idea is to enhance the overall flow data analysis experience and provide a concise and targeted approach to understanding network and application behavior.

The host OS (distribution) is a constant and necessary part of the overall application structure. The intent of the analysis below is to understand how the OS is utilized by various components of the network to influence application behavior. This will include understanding the volume of traffic per OS and how each OS uses the network resources.

The results inferred can further network engineers and architects understanding of applications and be considered when looking at network designs.

About the flow/host data set

The flow data sets subject to flow data analysis include a packet capture describing the overall network behavior (who is talking to who) and a description of the host communicating on the network during the specified time interval. The network flow data includes source and destination ip addresses along with host descriptions to assist in identifying traffic patterns and anomalies.

The intended flow data set has the following structures and descriptions.

Flows

- 1) host_name : Factor : the host name of the guest OS
- 2) last_software_update_at : int : the last time the host agent was updated
- 3) data_plane_disabled : logic : the status of the data plane forwarding
- 4) platform : Factor : the guest OS host distribution

- 5) agent_type : Factor : the current agent deployed on the guest os
- 6) current_sw_version : Factor : the current agent version as installed on the host
- 7) enable_pid_lookup : logic : the current status of the PID lookup functionality
- 8) last_config_fetch_at : int : the timeframe the guest OS last checked in

Sensors

- 1) start_timestamp : num : the timeframe the flow started
- 2) src_port : int : the source port for the selected flow
- 3) rev_pkts : int : the packet count for the reverse flows
- 4) rev_bytes : int : the byte count for the reverse flows
- 5) proto : Factor : the IP protocol used for the current flow
- 6) src_address : Factor : source IP ADDRESS of the host that initiated the flow
- 7) timestamp : Factor : the timestamp the flow was collected
- 8) fwd_bytes : int : the byte count for the forward direction of the flow
- 9) src_hostname : Factor : the host name of the source host in the flow
- 10) dst_address : Factor : destination IP ADDRESS of the host of which the source is communicating with
- 11) src_is_internal : Factor : the current state of the host relative to its location
- 12) dst_port : int : the destination port for the selected flow
- 13) rtt_usec : int : the rtt latency associated with the current flow
- 14) vrf_id : int : the vrf id for the current flow
- 15) vrf_name : Factor : the vrf name for the current flow
- 16) fwd_pkts : int : the packet count for the forward direction of the flow
- 17) server_app_latency_usec : int : the application latency as derived from the application agent
- 18) total_network_latency_usec : int : the calculated network latency
- 19) total_pkts : int : the total fwd and rev packet count
- 20) platform_windows : logic : the state of the platform, windows
- 21) total_bytes : int : the total fwd and rev byte count

Preparing the data

RStudio Libraries

The libraries listed below are utilized to provide functionality required to parse, analyze and display the appropriate flow data sets.

Loading the data sets

The 'csv' files representing the flow analysis are loaded into the variables flowData and SensorData. The data loaded is comprised of a network capture of multiple hosts communicating on the network during the define collection period and a host description file containing host attributes describing the given hosts.

Cleaning the flow/host data sets

To prepare the data for analysis, the data must first be normalized.

The collected data sets contain IPv4 and IPv6 addresses. For the purpose of this exercise, all IPv6 addresses will be removed as they are not part of the considered analysis. The data sets contain entries without hostnames defined, the associated flows will be identified and removed from the analysis as they are flows that do not contain agents.

To get a holistic view of the flows, the flow data and the host data will be combined. This effort will allow for additional opportunities to correlate flow and host level data. Once the data sources have been combined, additional fields will be added to the data set to assist in providing analysis. To provide summarization for packet attributes, a summary field of the fwd and rev packets and bytes will be added. The additional fields will allow for total packet/byte count per flow. For evaluating time intervals, a new column will be added to translate the discovered time interval into an hour and minute format.

In an effort to focus the data analysis, a filter for the specific hosts to be evaluated will be applied. This filter will remove any entries for flows outside of the desired hosts.

When considering the overall problem set and the data to be analyzed, a dependent variable related to the host type will be required. This field will be a summarization of the type of platform and translated into a '1' or '0' representing the host type (1= Windows, 2 = Linux).

Exploring the data

Part of understanding the results of a network study is to first understand the data. The next few sections will provide a description of the data elements and foundation for the analysis. The data exploration will describe various data points related to the overall traffic across the network and the hosts communicating.

Counting records and variables

Using R methods and functionality, determine various flow data set counts. Counting various aspects of your flow data set can prepare you for dealing with the content during analysis.

- Count the number of rows and columns in the flow data set, data/SensorData.csv.

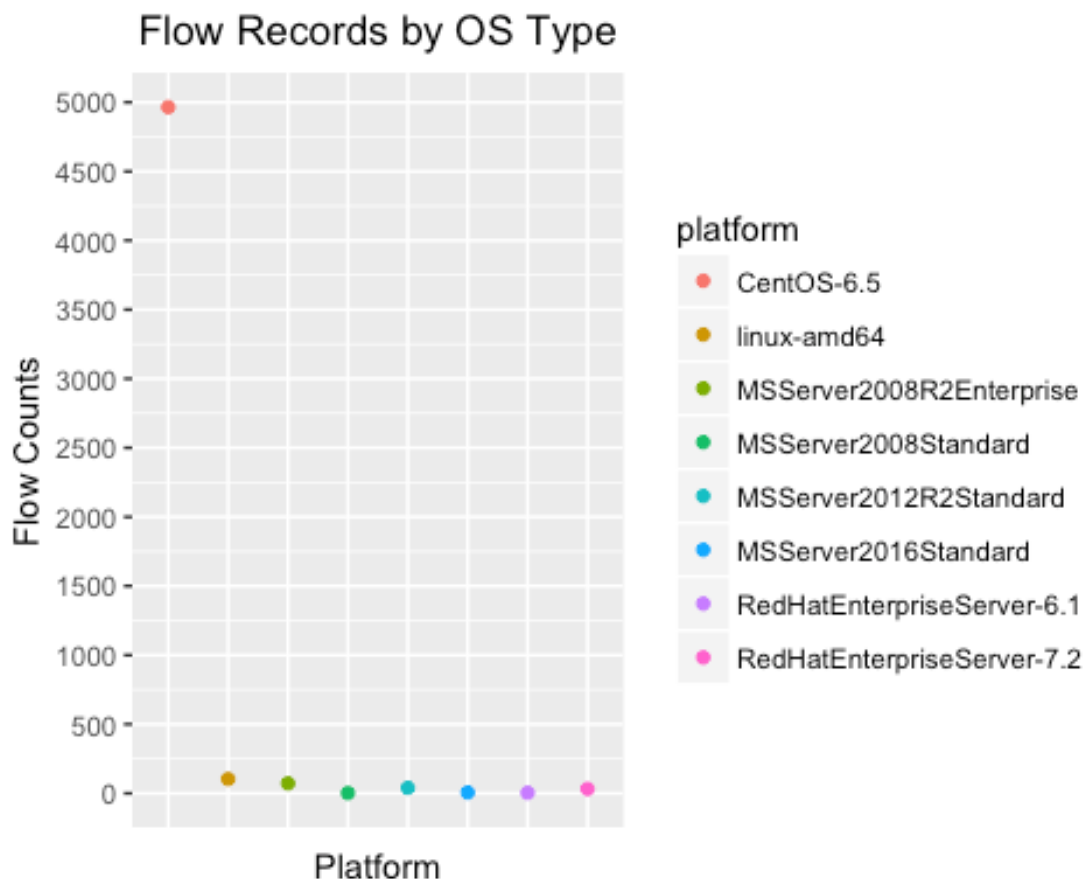
There are a total of 5225 ROWs and 35 COLUMNs in the combined flow data set. The 5225 entries in the data set contain the flow attributes and the host descriptions accordingly. The overall number of flows/records can further be filtered to allow for a more granular view.

- Count the number of rows in the flow data set per platform type (the data set dependent variable).

Within the data sets analyzed, the focus of the analysis will be how host types, Windows and/or Linux, affect traffic patterns in the network. From the selected data sets there are a total of 120 flows attributed to Windows based hosts and 5105 based on Linux hosts.

Understanding the data set as related to host OS.

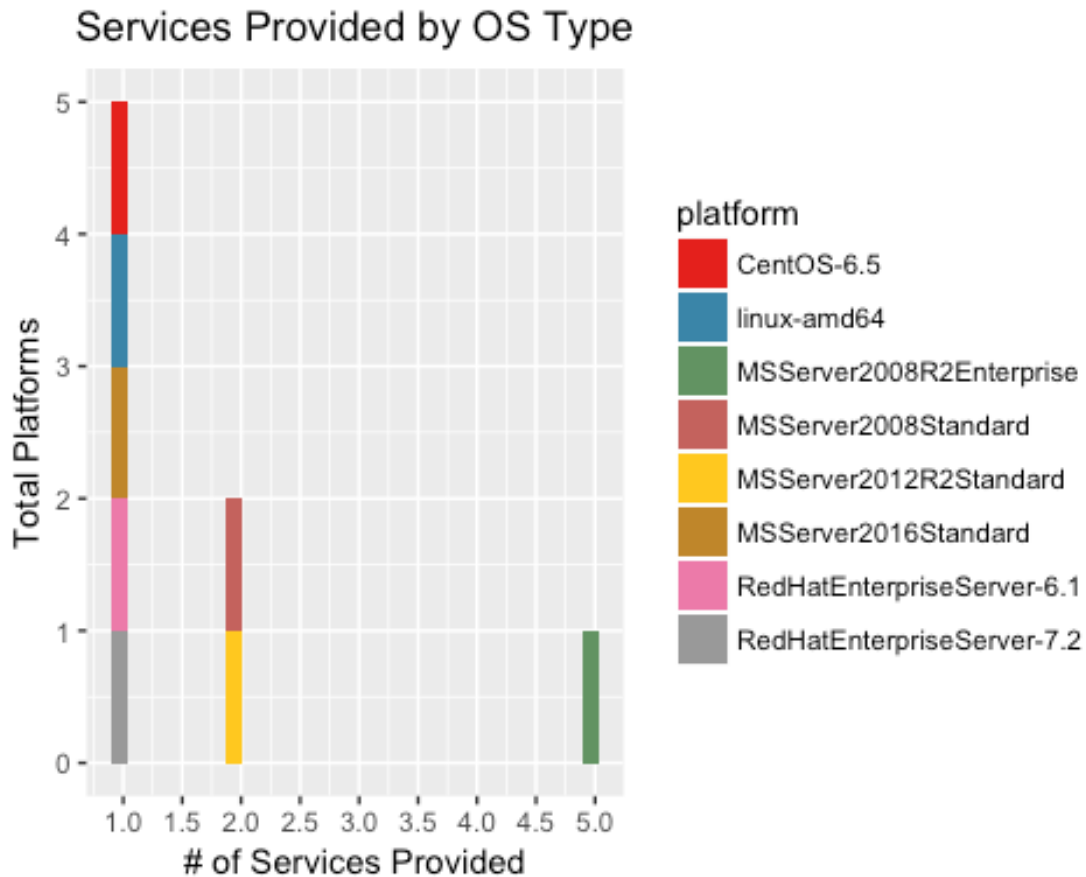
Using the combined data sets, we can get a better understanding of the guest OS distribution accross the collected flows. As a first step in the overall data analysis we need to gain insight into the overall distribution flowsby host guest OS type, Windows and Linux distributions.



As indicated in the table above, CentOS-6.5 is the most frequently utilized OS during the provided flow capture. As noted above in the table, the source of the traffic, host OS, is not evenly distributed accross OS types.

Services provided per OS

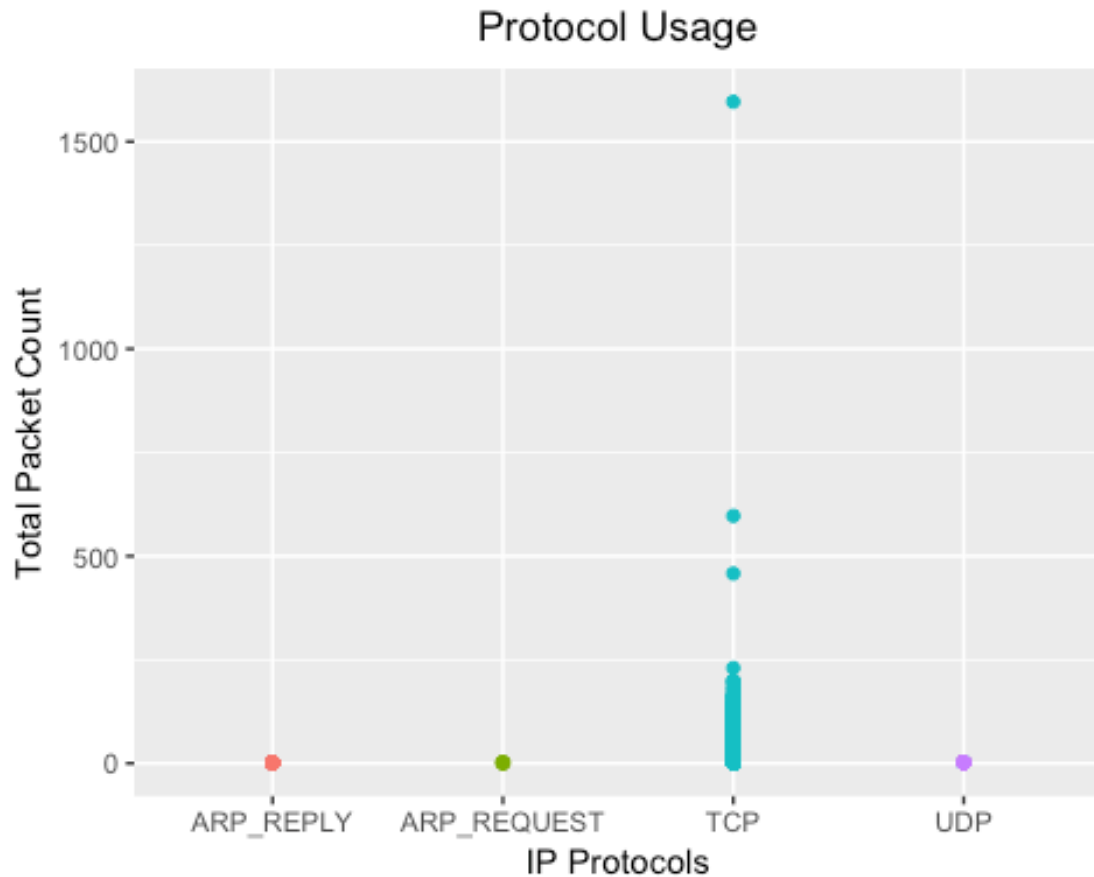
As we explore the flow data set further, we want to understand the number services provided by the guest OS type. This may be an indication of the guest OS (application) that is the most utilized on the network during the flow capture timeframe. The port number (destination port) ultimately gives an indication of the specific host behavior during the flow capture.



As indicated in the histogram above, we can determine that majority of the guest OS's are providing a single service. This can be interpreted as the majority of the OS's analyzed within the flow analysis are providing service to a single application.

Overall Protocol Usage

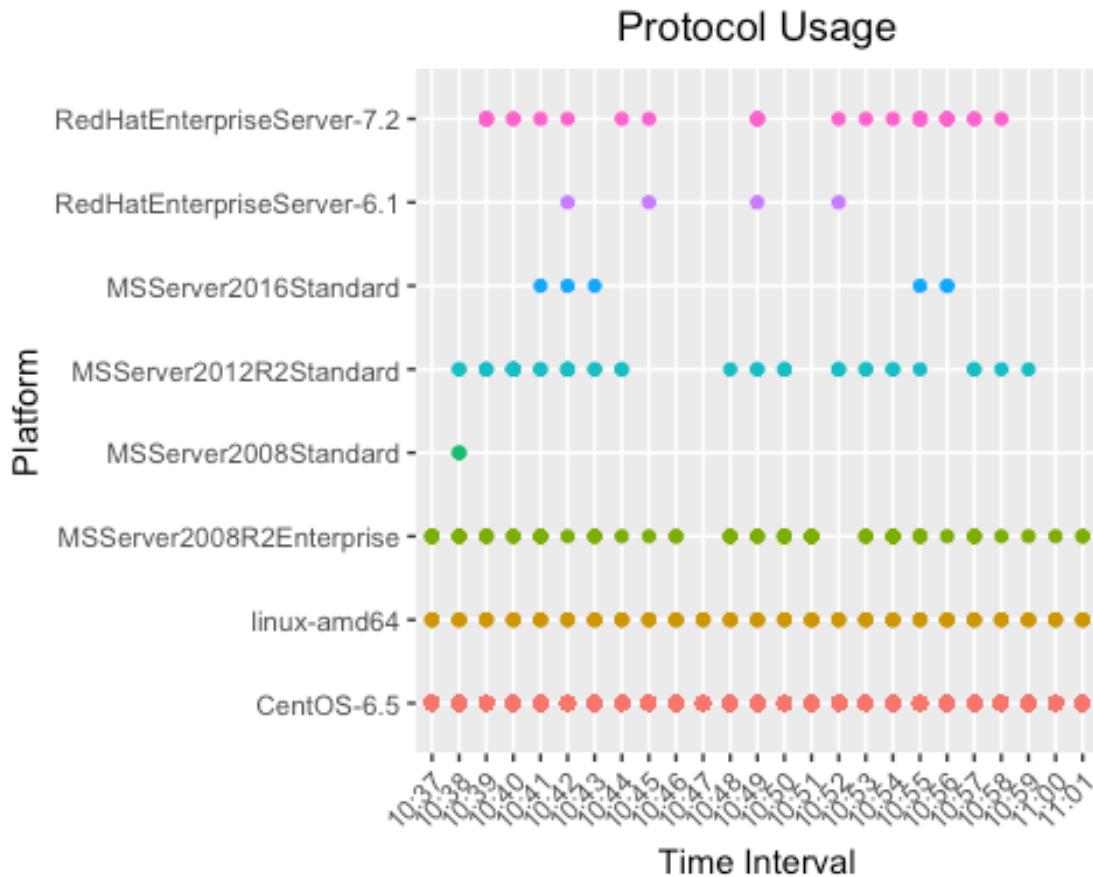
As we continue to understand how applications consume overall flows within the network, it's useful to understand overall protocol distribution. The section will focus on understanding how IP protocols are implemented for the flow capture period.



From the table above we can infer that TCP is predominately utilized via the host applications analysed during the capture period. The range of packets part of the TCP flows extend from 1 to 1500+ packets per flow.

Platform usage by Time

By further evaluating the OS traffic over time, we can get an idea as to when the applications communicated accross the network. The network consumption (flows consuming the netwrok) can be used to understand traffic patterns and network usage.

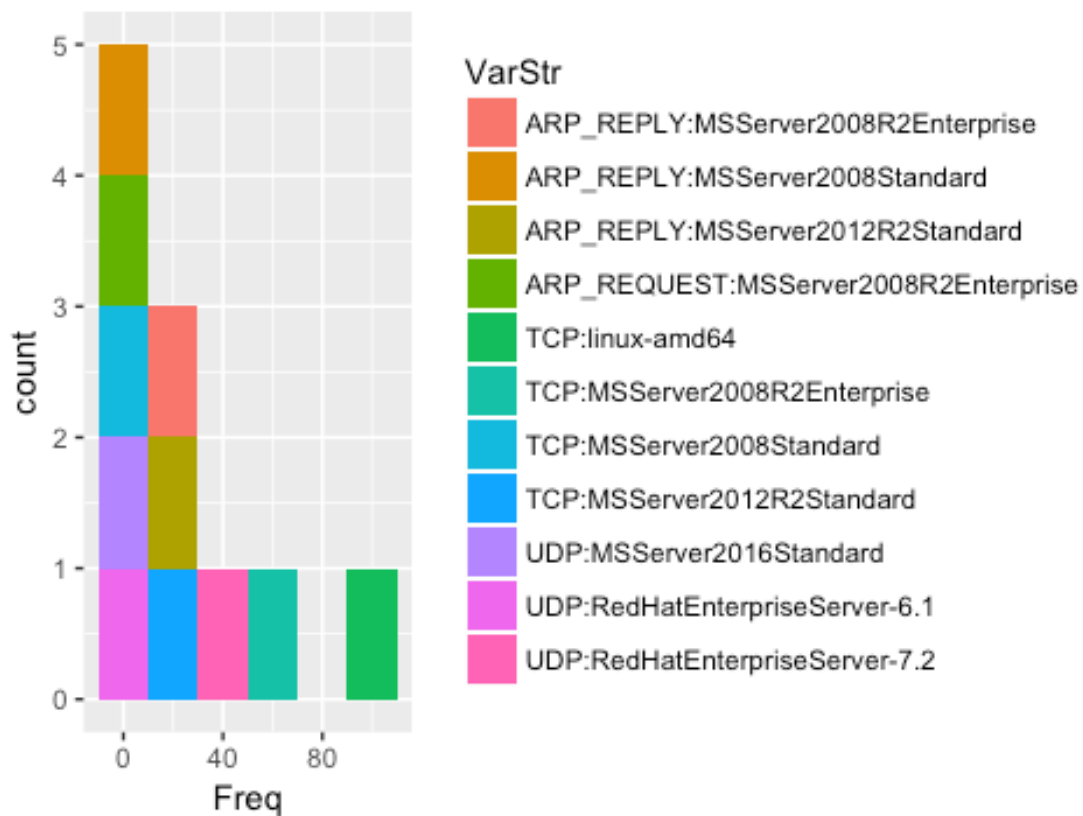


By graphing the usage of the infrastructure per host OS across the collected time intervals, we can determine that majority of the OS distributions communicate throughout the entirety of the interval with a couple of distributions only utilizing network resources on occasion.

Protocol usage by Platform

We have noted above how specific guest OS's utilize the network when considering the overall data collection. Now with R table functionality we can quickly evaluate the overall frequency of communications protocols per guest OS.

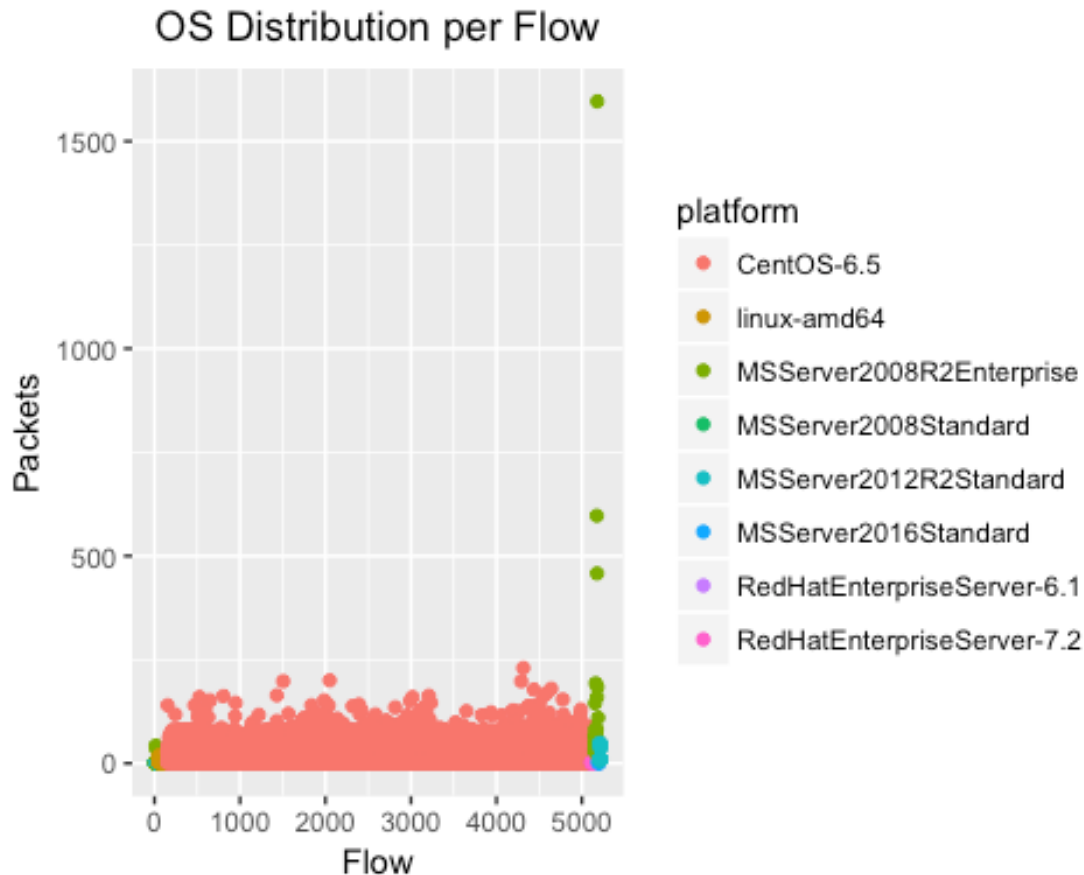
Distribution per OS Type and Protocol



Packect distribution over flow period

As noted in the previous section, we can see that CentOS makes up a majority of the traffic. Now that we have identified the top users of the network by OS distribution, let's now understand during the capture interval where the traffic was consumed.

To accomplish this task, we will first evaluate the traffic distribution across the entire interval, all data flows.



When exploring the total count of packets per OS type across the capture, we are able to identify that CentOS-6.5 is responsible for majority of the flows captured. The volume of CentOS-6.5 flows far out weights the all other OS distributions combined.

Analysing the data

Up until this point, we have used the data sets to glean data and begin to understand usage across the network. Now we will use data analysis techniques to help understand and predict flow behavior based on guest OS type, platform.

Data Analysis with Regression

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. Logistic regression is a special case of linear regression when the outcome variable is categorical.

This method is used to predict the probability of occurrence of an event by fitting data to a logit function. in this case the event is based on platform type.

The result of the regression model identifies that the insignificant values in the model are limited to REHL-6.1 and MS2008Standard.

The distribution of the deviance residuals for individual cases used in the model range from -3.8272 to .3789. The model output shows how `dst_pot` and `total_pkts` are statistically significant while `total_bytes` is not and can be removed from the model. As a result, for every one unit change in `total_pkts`, the log odds of the platform type windows (versus linux) increases by 0.006.

Now we can say that for a one unit increase in `total_pkts`, the odds of being a windows source flow (versus linux sourced flow) increase by a factor of 1.07

Data Analysis with a ROC Curve

ROC curves are commonly used to characterize the sensitivity/specificity tradeoffs for a binary classifier. Most machine learning classifiers produce real-valued scores that correspond with the strength of the prediction that a given case is positive. Turning these real-valued scores into yes or no predictions requires setting a threshold; cases with scores above the threshold are classified as positive, and cases with scores below the threshold are predicted to be negative.

Summary:

In conclusion from the data exploration, we are able to identify the overall behavior and impact of Windows for Linux distributions on the network. During the period of data capture the Linux distribution, CentOS-6.5, consumes majority of the flows in the network. This is a clear indication of the application host OS of choice for application distribution.

To better understand the services provided by the application we analyse the behavior of the flows as related to the services each distribution provides. The services provided indicate the type of application flow captured and present on the network. The results of the data exploration indicate that majority of the hosted application are serving up a single application for the consumers, clients.

To further the discussion of application discovery, the analysis shows that the application protocol of choice is TCP. The TCP based applications host applications that communicate evenly throughout the life of the flows. The indication for CentOS as the top talker during the flow capture is indicative of the Linux based applications dominant use of the network.

When looking further into the analysis of the flow set, the result of the regression model identifies that the insignificant values (OS distributions) are limited to REHL-6.1 and MS2008Standard.

The distribution of the deviance residuals for individual cases used in the model range from -3.8272 to .3789. The model output shows how `dst_pot` and `total_pkts` are statistically significant while `total_bytes` is not and can be removed from the model. As a result, for every one unit change in `total_pkts`, the log odds of the platform type windows (versus linux) increases by 0.006.

Now we can say that for a one unit increase in `total_pkts`, the odds of being a windows source flow (versus linux sourced flow) increase by a factor of 1.07.