nMASE – A Search Engine for Network Trace

Abhishek Chandratre[1], Prerit Auti[2], Ritesh Porey[3], Vipul Chaskar[4]

Information Technology Department, Pune Institute of Computer Technology, Pune
Computer Science and Information Technology [CS]

Abstract: ***Computer networks have been largely adopted in small and big settings in corporate and residential environments. With explosion of mobile traffic and ever increasing use of software defined networking, network traffic monitoring holds a key position in detecting network errors and security failures. Monitoring big networks has become a herculean task with having to deal with huge magnitude of data which is prohibitive for proper leverage considering the time and space required to exploit it. Much of the current work in this domain is focused on techniques to efficiently analyze and classify network traffic. Rule based classification or filtering of traffic is the first major step in any analysis procedure so that analysis is performed on focused set of packets. However defining, constructing and application of such rules involve considerable level of difficulty. In this paper we address this issue. We believe that searching for relevant content in network trace should be as easy as searching for relevant web pages through search engine. In this paper we touch upon this much neglected area by introducing concept of network search engine and ranking of network traffic. "nMASE- A Search Engine for Network Trace" is built upon Boolean model of information retrieval and is capable of finding relevant content from a huge network trace based on simple natural language queries. This leads to significant reduction in the time that network administrators spend in scanning through colossal network activity.***

Keywords: Network Monitoring, PF_RING, Bitmap Indexes, Search Engine, Network Flows, Custom Packet Capture

1. Introduction

Network Monitoring Systems consist of capturing traffic from a central location and analyzing it for interesting events. These systems face challenges of dealing with large amount of captured raw data. Also searching and filtering this data to find relevant information is time consuming. Typical purpose of network monitoring is to identify the issues arising in a network and inform the network administrator about the same. Key benefits of network monitoring include reliability, troubleshooting and helps the admin to be aware and updated about the happenings in the network. Another obstacle these systems face is of correctly determining which protocol being used. The current system use rules for filtering which are cumbersome to write and understand when dealing with enormous amount of data.

To overcome these drawbacks we introduce nMASE- A Search Engine for Network Trace. It analyzes and records network activity and provides the user with relevant results through a Web Interface. It incorporates flow-based ranking and searching. Users desire easy search and retrieval of required network activity. nMASE is an intelligent end to end system which performs everything from capturing raw packets to displaying search results through web interface. The prevalent approach is to use network monitoring tools which just capture the packets and show it on screen.

2. Motivation

Current Open Source Network Monitoring tools such as tcpdump and Wireshark allow the captured data to be browsed via GUI. However, they consume a significant amount of RAM and it is tedious to search for required data using rule base filtering. nMASE provides an elegant solution for these problems, with benefits like faster and easier search facilitated by indexes, reduced packet drop rate and low RAM consumption. This will provide much needed intuitive system for administration and monitoring purposes.

3. Background

A. Flow

Packet flow or network flow is a stream of packets from a source machine to destination in a specific time frame. A flow consists of an ordered set of packets from one particular source IP and Port to a specific destination IP and Port. Basically, it is a sequence of packets transferred between two unique Sockets on the network. The flows used in nMASE are bidirectional i.e. packets going to and fro belong to the same flow. Thus, the terms „source‟ and „destination‟ are relative.

B. PF_RING

PF_RING is a kernel module used for kernel level packet capturing. PF_RING [10] is a protocol handler used to efficiently transfer the packets from Network Interface Card (NIC) buffer memory to a ring buffer present in the kernel space. This module resides at second layer of OSI model and works by capturing raw packets with hooks placed in kernel [11]. We are using this module as part of our packet capturing module to enhance the speed of packet capture.

C. Bitmap Indexes

A bitmap index is a type of database index that uses bit vectors. These indexes work efficiently for columns with low cardinalities i.e. those having small number of distinct values. Bitmap indexes have considerable improvement over other indexes in terms of performance and space. These indexes use bitmap and perform bitwise logical operations to answer the queries. Bitmap indexes are used with significant success in areas of databases and information retrieval.

D. Meta Data

Metadata summarizes basic information about data, which can make finding and working with particular instances of data easier. It is data which gives relevant information about payload present in network trace. It also gives information about containers of data and about individual instances of application data.

E. Keyword based Searching

Keyword based search involves extracting key terms from query and searching for the same in the present data set. It is different from natural language search where there is overhead of creating syntax tree with respect to queries.

4. Architecture

nMASE can be deployed at network taps from where all the flowing network traffic as input. This system is divided into three major components.

1) Kernel level packet capturing module.
2) Metadata Extractor.
3) Information Retrieval module and Interface.

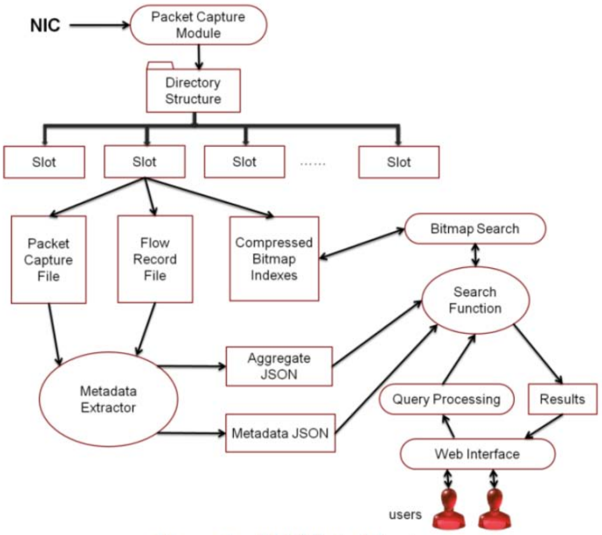

Figure 1. nMASE Architecture

A. Kernel level packet capturing module

All network activity for nodes inside a network passes through a central router also called as network tap. Out first module is deployed at this point where we capture all the packets. These packets are copied from ring buffer of PF_RING kernel module which was previously taken directly from NIC temporary buffer. These packets are then sent to the first module where they are processed up to transport layer.This module generates flow records by analyzing incoming packets.

1) Tree Structure for Packet chaining: We are using a new concept of segregating the packets based on a tree structure corresponding to their IP addresses and ports. This tree structure helps in separating the packets belonging to different flows and chaining them together. Each flow record points to the first packet of its chain and then all the subsequent packets point to next one in the flow. We use separate trees for each transport level protocol. Whenever the first packet of a new flow comes, a new path is generated in the tree which is followed by all the packets in the same flow. These flows are terminated either by timeout or FIN/RST flags.
2) Compressed Bitmap Index Creation: This module also generates compressed bitmap indexes based on Word-Aligned Hybrid (WAH) scheme. These indexes are generated based on following fields- source and destination IP, source and destination port and transport protocol. The compressed indexes provide us with better space efficiency. WAH scheme leverages the fact that most of bitmap indexes are sparse.
3) Storage in Directory Structure: These packets, flow records and indexes are stored in different directories segregated by time slot. For each hour new directory is created in which its corresponding files are stored. This segregation helps in narrowing down search to specific time range which user may specify in query.

B. Meta Data Extractor

Second module mainly deals with application layer data. It reads the flow records and their corresponding packets which first module had stored in directory structure. Then it performs Deep Analysis on the packets to determine application layer protocol and extract and store interesting application-level metadata.

1) Determining Application layer protocol: Port number is not an accurate indicator of type of payload present inside the network. For example, our studies have found that around 5-10% of traffic to port 80 is not HTTP. HTTP is the only protocol which is assigned to port 80. Therefore, nMASE manually checks for application layer protocol by matching it against a set of signatures of protocols that are supported. This is done by analyzing a fixed number of packets belonging to each flow. On matching of a signature of a particular protocol, its corresponding probability (likelihood of presence of that protocol in flow under consideration) is increased. If probability of a protocol exceeds a defined threshold, that flow is marked as having matched protocol at application layer.
2) Extracting and storing metadata: After determining the application layer protocol for all the flows, the second module extracts metadata from payload. Performing search and analysis on the entire set of packets is cumbersome and time consuming. This is why all the important informationcommonly required is extracted to generate metadata. This metadata is stored in JSON- a light weight file format. The data in these files are organized and stored based on flows. Also, this metadata is aggregated to help in querying through it easy and fast.
3) Detecting protocol based attacks and errors: While analyzing and generating the metadata, this module also detects the presence of errors and attacks in the captured trace by matching it with predefined set of attack patterns. For example, presence of HTML <script> tags inside HTTP request URLs indicate a possible Cross Site Scripting (XSS) attack.

C. Information Retrieval module and Interface

This module provides a web interface for the user along with information retrieval and query processing functionalities.

1)Query processing: In this sub-module query accepted from the user is parsed to extract the important key words and values from it. This query is converted in to internal expanded representation. This sub-module accepts query which may contain key words related to Network, Transport and Application layer. For example, consider the query “IP starting from 192.168 who visited the website www.wikipedia.org”. This will show the matching flows originating from IPs 192.168.*.* who accessed the given website.

2) Information Retrieval: This module is a key part of the overall system. It comprises of search functionality and ranking mechanism.

Search Function: It accepts the query in the expanded form. It then performs two major types of searches based on key words present in the query. For network and transport layer keywords like IP and Port address, it searches through compressed Bitmap Indexes created in the first module. For application layer keywords this function searches through the aggregated Meta data which is generated by the second module i.e. Meta data extractor. It combines the result fromboth and uses the ranking mechanism to display them.

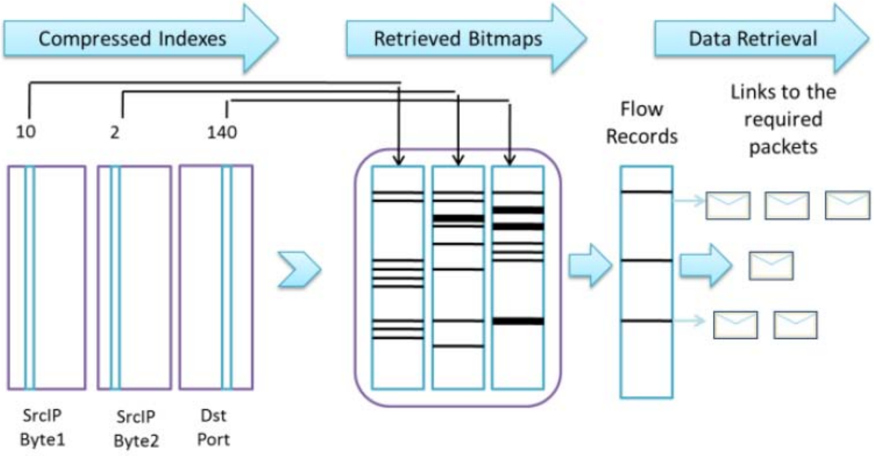

Figure 2. Data Retrieval based on Bitmap Indexes.

Ranking Mechanism: nMASE assigns scores to every flow. There are two components of scores i.e. static and dynamic. Static component include presence of errors, data usage and presence of payload. The Dynamic component includes number of hits, age and percentage match with the query. The combined score is calculated and then the sorted results are returned to the interface.

[3] Web Interface: nMASE provides a familiar search engine interface as simplistic as Google. The user is shown the summary of all matching flows along with Meta data. The user even has the capability to view the structure and content of individual packets belonging to the flow.

5. Practical Application

nMASE can be deployed on Server, from where all the network transactions takes place. Network administrators thus can monitor the complete history of network in lucid manner. They can fix problems, without even applying rules. By providing Ranking and Visual cues users can easily spot the errors. Users can also generate reports and view statistics with help of nMASE. Naïve users can also use nMASE for educational purpose. They can monitor their network, keep a track of past events, learn how network actually work in real life. nMASE will overwrite the traditional method of monitoring and fixing problems.

6. Results

The main advantage of this idea is to eliminate the user‟s need to study every packet in order to determine any particular information of a single packet or an entire flow. This system makes use of a ranking mechanism which helps to attain relevant information depending upon various factors. Also, the use of bitmap search indexes for storing the metadata can speed up the information retrieval process. Along with this, storage space can be significantly reduced with the use of compression techniques and making use of aggregated metadata.

We have implemented a prototype of this model and the following results have been collected.

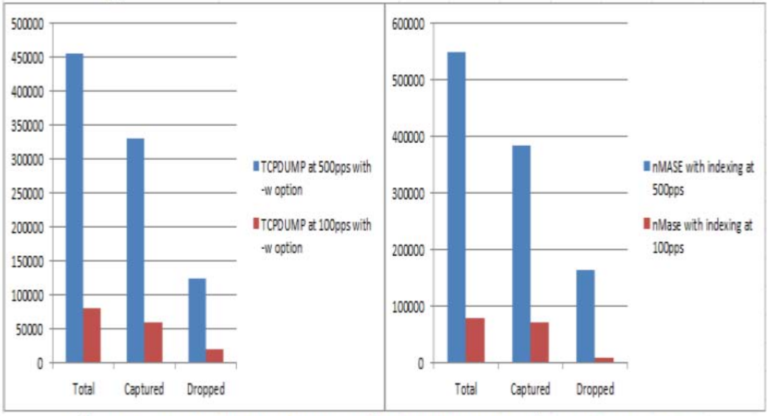

Figure 3. TCPDUMP and nMASE Packet Drop Rate

7. Conclusion and Future Work

This paper presented nMASE- A Search Engine for Networks to analyze network activity and display relevant data to the users. It can monitor and record network activity by being deployed on a central location. It aims to reduce the burden of network administrators by significantly reducing the time spent in scanning through the network logs. nMASE efficiently minimizes the space requirements by using compressed Bitmap search indexes. It stores metadata in light weight format, and the user is given the facility clear old capture files. In future, this system can be extended to support larger networks.

References

[1] Randy Heins, "Indexing full packet capture data with flow", FloCon, January 2011
[2] AntonisPapadogiannakis, Michalis Polychronakis, Evangelos P. Markatos, "Tolerating Overload Attacks against Packet Capture Systems", USENIX Annual Technical Conference (ATC) 2012, June 2012
[3] Dreger, Feldmann, Mai, Paxson, Sommer, "Dynamic Application layer Protocol Analysis for Network Intrusion Detection", USENIX Security Symposium Proceedings, 2006
[4] Antunnes, Neves, Verissimo, "Reverse Engineering of Protocols from Network Traces", Proceedings of the 18th Working Conference on Reverse Engineering (WCRE), October 2011
[5] Francesco Fusco, XenofontasDimitropoulos, Michail Vlachos, Luca Deri, "pcapIndex : an Index for Network Packet Traces with Legacy Compatibility", ACM SIGCOMM Computer Communication Review (CCR), Jan. 2012
[6] Francesco Fusco, Marc Stoecklin, Michail Vlachos, "NETFLi: Onthefly Compression, Archiving and Indexing of Streaming Network Traffic", International Conference on Very Large Databases, 2010
[7] Derek Banks, "Custom Full Packet Capture System", SANS Institute InfoSec, 2013
[8] Young-Hwan Kima, Roberto Konowb, Diego Dujovneb, Thierry Turlettia, WalidDabbousa, Gonzalo Navarroc, "PcapWT: An Efficient Packet Extraction Tool for Large Volume Network Traces", HAL-INRIA archives, Jan 2014.
[9] Rick Hofstede, PavelCeleda, Brian Trammell, IdilioDrago, RaminSadre, Anna Sperotto and AikoPras, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX", Communication Surveys and Tutorials, IEEE, May 2014
[10] Improving Passive Packet Capture: Beyond Device Polling, L. Deri, Proc. of SANE 2004.
[11] www.ecsl.cs.sunysb.edu/elibrary/linux/network/LinuxKernel.pdf