7/16/2014

BTP Report

Big Data Analysis

Yogesh Dorbala – cs1100112
Kishore Rajendra – cs1100118
IIT INDORE

CONTENTS

Big Data - An Introduction

Big Data:

- The rate of data creation has increased so much that 90% of the data in the world today has been created in the last two years alone. This acceleration in the production of information has created a need for new technologies to analyze massive data sets.
- The term Big Data refers to large-scale information management and analysis technologies that exceed the capability of traditional data processing technologies.

Big Data Analytics:

- Big Data Analytics is a paradigm where large data sets are split into small ones and given to worker nodes to compute and all the results are appended to give the result.
- This has a huge advantage as data with variety, velocity and volume can be managed easily. The need to analyze and leverage trend data collected by businesses is one of the main drivers for Big Data analysis tools.
- The technological advances in storage, processing, and analysis of Big Data include (a) the rapidly decreasing cost of storage and CPU power in recent years; (b) the flexibility and cost-effectiveness of datacenters and cloud computing for elastic computation and storage; and (c) the development of new frameworks such as Hadoop, which allow users to take advantage of these distributed computing systems storing large quantities of data through flexible parallel processing

Technical factors driving Big Data adoption:

- Storage cost has dramatically decreased in the last few years. Therefore, while traditional data warehouse operations retained data for a specific time interval, Big Data applications retain data indefinitely to understand long historical trends.
- Big Data tools such as the Hadoop ecosystem and NoSQL databases provide the technology to increase the processing speed of complex queries and analytics.
- Extract, Transform, and Load (ETL) in traditional data warehouses is rigid because users have to define schemas ahead of time

Types:

- Big Data technologies can be divided into two groups: ***batch processing***, which are analytics on data at rest, and ***stream processing***, which are analytics on data in motion.

Hadoop:

- Hadoop is one of the most popular technologies for batch processing. It lets user to define the formats according to their needs and analyze the data.
- Several tools can help analysts create complex queries and run machine learning algorithms on top of Hadoop. These tools include Pig (a platform and a scripting language for complex queries), Hive (an SQL-friendly query language), and Mahout and RHadoop.
- Stream processing does not have a single dominant technology like Hadoop, but is a growing area of research and development (Cugola & Margara 2012). One of the models for stream processing is Complex Event Processing.

Overview:

- Big Data analytics can be leveraged to improve information security and situational awareness. For example, Big Data analytics can be employed to analyze financial transactions, log files, and network traffic to identify anomalies and suspicious activities, and to correlate multiple sources of information into a coherent view.
- The report explores how to use Big data Analytics to solve current problems with having to analyze Terabytes of data through the help of research paper publications.

“BotCloud: Detecting Botnets Using MapReduce”

Botnets:

- A botnet is a network of compromised hosts (bots) which are controlled by an attacker also called the botmaster. The botmaster sends commands via a C&C (Command and Control) channel.
- The current botnets are based on peer to-peer technologies where each bot acts as a client and a server.

Overview:

- Botnets are a major threat of the current Internet. Understanding the novel generation of botnets relying on peer to-peer networks is crucial for mitigating this threat.
- Nowadays, botnet traffic is mixed with a huge volume of benign traffic due to almost ubiquitous high speed networks. Such networks can be monitored using IP flow records but their forensic analysis form the major computational bottleneck.

Approach:

- Due to scalability issues in high speed networks, common solutions focus exclusively on Netflow data. This is an aggregated view of the network traffic excluding content and thus, avoid many privacy issues which have to be considered in forensic analysis.
- The paper proposes a method to detect new generation of botnets from large dataset of Netflow data, such as those gathered by each individual operator.The normal approach is extended by leveraging cloud computing paradigms especially MapReduce for detecting densely interconnected hosts which are potential botnet members.

Using Big Data:

- Botnet detection method is based on Hadoop, an open source implementation of MapReduce. A common deployment of an Hadoop cluster is represented in the lower part of figure with a master and slave nodes.
- The first key component is the Hadoop Distributed File System (HDFS) for storing data. The namenode daemon maintains the file namespace (directory structure, the location of file blocks). However, the blocks are directly stored on the slaves (datanodes) and guarantee a redundancy.

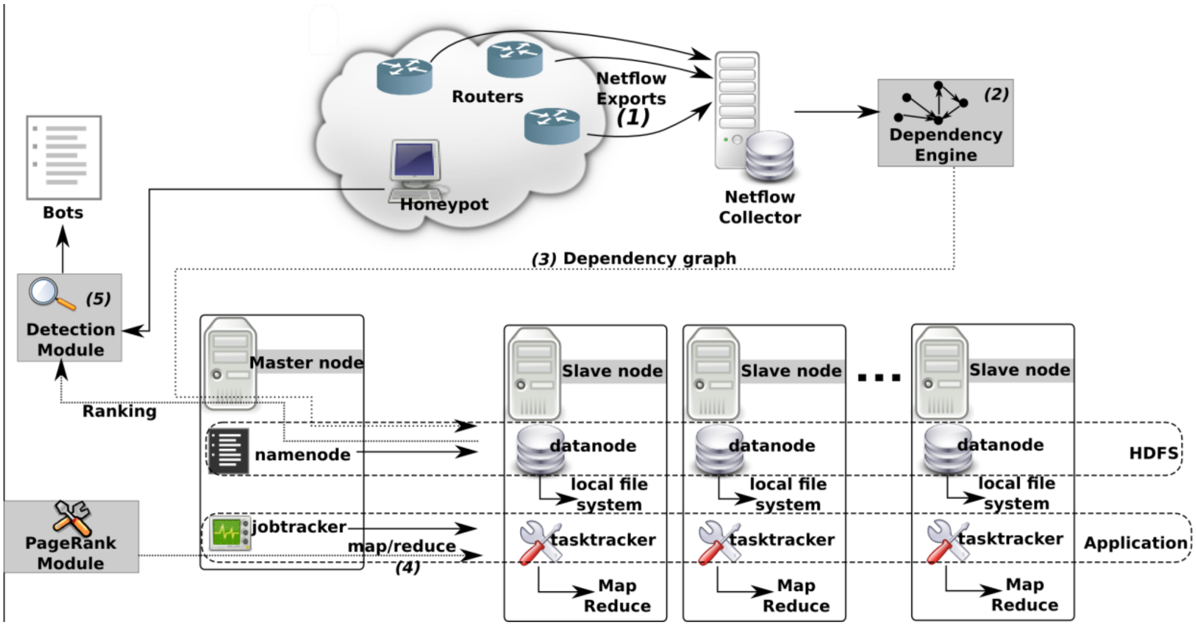

Fig. Bot Cloud Framework

- Considering the application side, the jobtracker takes as input a MapReduce job and is responsible to coordinate (task assignment) and monitor the map and reduce tasks. For improving the robustness, the jobtracker and namenode daemons may be executed on different master machines.

Final Say:

- This paper describes a scalable method for detecting P2P botnets regarding the relationships between hosts.
- The evaluation shows a good detection accuracy and a good efficiency based on a Hadoop cluster.

References:

- Research Paper on *"BotCloud: Detecting Botnets Using MapReduce",* by Jerome Francois, Shaonan Wang, Walter Bronzi, Radu State, Thomas Engel.

“Detecting DDoS Attacks with Hadoop”

Overview:

- Recent distributed denial-of-service (DDoS) attacks have demonstrated horrible destructive power by paralyzing web servers within short time. As the volume of Internet traffic rapidly grows up, the current DDoS detection technologies have met a new challenge that should efficiently deal with a huge amount of traffic within the affordable response time.
- This paper devised a DDoS anomaly detection method on Hadoop that implements a MapReduce-based detection algorithm against the HTTP GET flooding attack.

Approach:

- Counter-based detection is a simple method that counts the total traffic volume or the number of web page requests. Since the DDoS attack with the low volume of traffic such as the HTTP GET incomplete attack is prevalent these days, the frequency of page requests from clients will be a more effective factor.
- The detection method is based on Hadoop which is an open-source distributed cluster platform that includes a distributed fie system, HDFS and the programming model, MapReduce.

Hadoop – MapReduce Algorithm:

- In the MapReduce algorithm, the map function filters non-HTTP GET packets and generates key values of server IP address, masked timestamp, and client IP address. The reduce function summarizes the number of URL requests, page requests, and server response between a client and a server. Finally, the algorithm aggregates values per server. When total requests for a specific server exceeds the threshold, the records are marked as attackers.
- A small Hadoop testbed consisting of one master node and ten slave nodes was configured for experimental purpose. To manifest the scalability of the algorithm, the performance of the counter-based DDoS detection method was measured by varying cluster nodes.

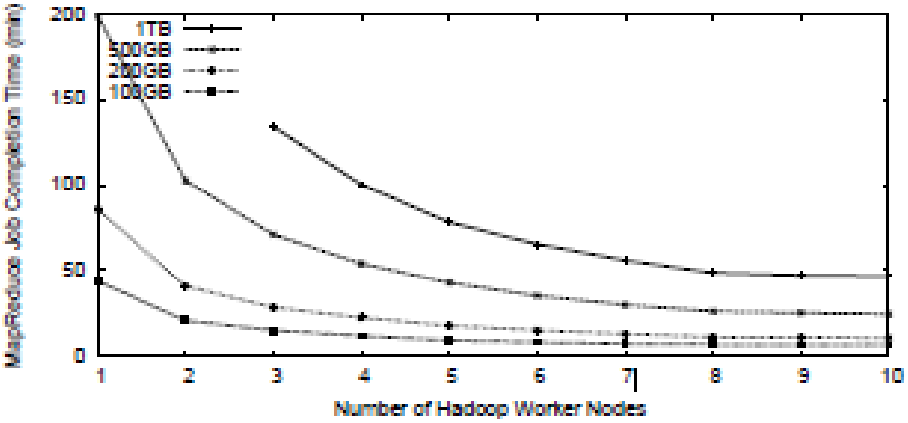

Fig. Completion time of a Counter based DDoS detection job

- Figure shows that the detection job with ten worker nodes is over 8 times faster than one node's and 2.9 times faster than 3 nodes' respectively. From evaluation results we could observe the increased performance enhancement as the volume of input traffic becomes large.

Conclusion:

- In this paper, the focus was on a scalability issue of the anomaly detection. The paper also introduced a Hadoop-based DDoS detection scheme to detect multiple attacks from a huge volume of traffic.
- This method leverages Hadoop to solve the scalability issue by parallel data processing.

References:

- Research Paper on *"Detecting DDoS Attacks with Hadoop",* by Yeonhee Lee, Youngseok Lee.

"Flow-based Brute-force Attack Detection"

Brute-force Attack:

- Brute-force attacks are a prevalent phenomenon that is getting harder to successfully detect on a network level due to increasing volume and encryption of network traffic and growing ubiquity of high-speed networks.

Overview:

- This paper presents several methods for the detection of brute-force attacks based on the analysis of network flows and discusses their strengths and shortcomings.
- It also demonstrates the fragility of some methods by introducing detection evasion techniques.

Types of Attacks:

- Attacks in general can be divided into two categories depending on their impact on traffic patterns. On one side there are noisy attacks that disrupt these patterns significantly, and on the other side there are stealthy attacks that are much harder to gather and examine as they, by virtue try to remain undetected. Different methods to defend from these attacks are described in this paper.
- Since a lot of network traffic is encrypted, the traditional approach to network-based intrusion detection is becoming ineffective. Packet payload which is searched for signatures of known attacks by deep packet inspection is opaque, only packet headers can be analyzed.

Flow-based Detection:

- Thus, flow-based detection is one of the possible ways to deal with encrypted traffic. The later sections in the paper explain how it is possible to use network flows to discover anomalies and intrusions.
- Brute-force attacks are most frequently detected at the host level by inspecting access logs. If the predefined number of unsuccessful login attempts is reached, an alert is fired, the attacker blocked or other attempts significantly delayed. This approach is effective, even for distributed attacks.
- The main drawback is that it does not scale well.

Brute force Detection Types:

- *Signature based* - The flow based signatures describe network traffic by specific values, or ranges of values, of flow features and computed statistics. The signatures are then searched in acquired flows. This approach is very straightforward and simple.
- *Similarity based* - searching for similar flows instead of matching specific signatures. It is believed that the similarity of traffic can point to machine-generated traffic, for instance brute-force attacks. This is a more generic approach and its detection capability essentially depends on a chosen algorithm

Downfalls of Network flows for Detection:

- When network traffic is converted to network flows, certain information remains (IP addresses and ports) and some information is derived (bytes per ow or packets per flow). Other information, like the packet payload, is inevitably lost. This loss of information implies that network flows are not suitable for the detection of all kinds of malicious network activity.
- The acquired flows are sent at least with a 5-minute delay which could be considered too late in case of attack which happen very fast.

Conclusion:

- This chapter gave an overview about current research in the field of flow-based attack and anomaly detection.
- We summarized state of the art concepts for attack detection, especially brute-force ones, and concluded the chapter by discussing evasion strategies as well as the limitations inherent to the process of detecting attacks in network ows.

References:

- Research Paper on *"Flow-based Brute-force Attack Detection",* by Martin Drasar and Jan Vykopal.

"TCP Flow Analysis for Defense against Shrew DDoS Attacks"

Shrew or RoS Attack:

- The Shrew or RoS(reduction-of-service) attacks are low-rate DdoS attacks that degrade the QoS to end systems slowly but not to deny the services completely.
- These attacks are more difficult to detect than the flooding type of DDoS attacks. In this paper, the energy distributions of Internet traffic flows in frequency domain are explored.

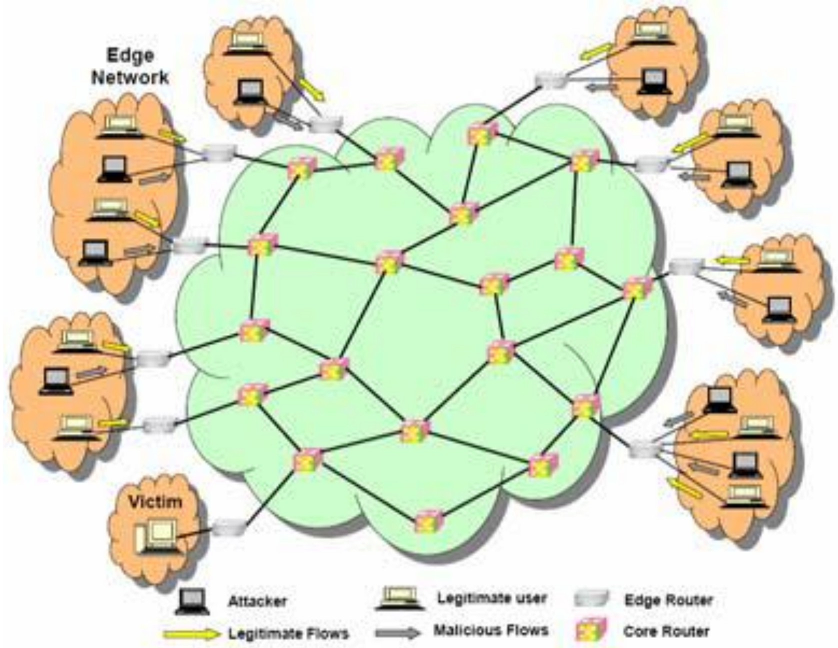

Fig. RoS attack scenario on end systems at Internet edge networks

Approach:

- Normal TCP traffic flows present some form of periodicity because of TCP protocol behavior. The results reveal that normal TCP flows can be segregated from malicious flows using some energy distribution properties.
- Combining flow-level spectral analysis with sequential hypothesis testing, a novel defense scheme against shrew DDoS or RoS attacks is proposed. This detection and filtering scheme can effectively rescue 99% legitimate TCP flows under the RoS attacks.

Traffic Spectrum:

- The main idea is to use spectral analysis to establish a *traffic spectrum* that describes the behavior of Internet traffic flows using frequency-domain characteristics.
- Using traffic spectrum information, the paper proposes a novel approach that can distinguish attack flows from legitimate flow.

Advantages:

- The scheme distinguishes normal TCP flows from others by observing the energy distribution and its standard deviation.
- The scheme detects malicious RoS attack flows accurately and swiftly. Legitimate TCP applications are saved from the attack flows.
- The scheme segregates legitimate TCP flows fromflooding DDoS attack flows. This property is very helpful to minimize the collateral damage to legitimate flows while packet-dropping mechanism is adopted

.

Conclusions:

- Analyzing Internet traffic spectrum in frequency domain enabled in solving some network anomaly problems that could not be solved effectively by volume-based traffic monitoring in real time.
- The scheme effectively rescues legitimate TCP flows from RoS attacks, which are very hard to detect in time domain for their stealthy properties.

References:

- Research Paper on *“TCP Flow Analysis for Defense against Shrew DDoS Attacks”,* by Yu Chen and Kai Hwang*.*

Objectives for the Project

- In our project, we plan to use Hadoop to analyze very large data-sets of Network-Flow data for different process.
- There are many new languages and databases to support Big data framework Hadoop. We would exlpore to find the best suitable tools and languages for our project.
- Several databases are designed specifically for efficient storage and query of Big Data, including Cassandra, CouchDB, Greenplum Database, HBase, MongoDB, and Vertica.
- Some tools that might be needed in our project are: Pig (platform and scripting language) , Hive (SQL-friendly language), Mahout and Rhadoop (data-mining and machine learning algorithms for Hadoop), and Spark framework.
- We will use different methods described in other papers to accurately analyze the network flow. Using Big data analysis, we have the power and capability to analyze very large sets of data in very less time.

End Deliverables for the Project

- Report on the Network flow analysis using BigData taking into consideration all the factors governing the flow.
- After collecting real-data for a network flow, we will analyse and detect any anomolies or attacks. If required, we may have to generate attacks by ourselves to verify our system.
- Deliverables are methods to prevent or detect attacks using Big-Data framework Hadoop.