Security problems and challenges in a machine learning-based hybrid big data processing network systems

By: Jeff Whitworth and Shan Suthaharan

J. Whitworth, and S. Suthaharan. 2014. Security problems and challenges in a machine learning-based hybrid big data processing network systems. ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 82-85.

Abstract:

The data source that produces data continuously in high volume and high velocity with large varieties of data types creates Big Data, and causes problems and challenges to Machine Learning (ML) techniques that help extract, analyze and visualize important information. To overcome these problems and challenges, we propose to make use of the hybrid networking model that consists of multiple components such as Hadoop distributed file system (HDFS), cloud storage system, security module and ML unit. Processing of Big Data in this networking environment with ML technique requires user interaction and additional storage hence some artificial delay between the arrivals of data domains through external storage can help HDFSto process the Big Data efficiently. To address this problem we suggest using public cloud for data storage which will induce meaningful time delay to the data while making use of its storage capability. However, the use of public cloud will lead to security vulnerability to the data transmission and storage. Therefore, we need some form of security algorithm that provides a flexible key-based encryption technique that can provide tradeoffs between time-delay, security strength and storage risks. In this paper we propose a model for using public cloud provider trust levels to select encryption types for data storage for use within a Big Data analytics network topology.

Keywords: Big Data | hybrid Cloud | encryption | retrievability | machine learning

Article:

1. INTRODUCTION

The development of cloud computing services introduces several factors impacting the Big Data problem. We focus in this paper on the specific challenges posed to the security of data stored with service providers in a hybrid cloud configuration for solving Big Data classification problem using Machine Learning (ML) techniques in highly secure manner. As Infrastructure as a Service (IaaS) offerings mature they increase in complexity which also increases the variables impacting data risk levels. Not only are IaaS offerings maturing, but new offerings are being introduced into the market at a rapid pace. However, there is no standard model for determining

the risk level of a cloud service. This concern grows as corporations and governments continue to move services from internal hosting to external cloud service providers to meet the growing demands of Big Data computation [5, 3, 11]. Storage IaaS imposes some inherent security challenges by the very nature of cloud computing.

As other researchers have outlined [11, 9, 4], one strategy to address cloud computing security is to only send nonsensitive data to cloud services. However, while this strategy may work for tagged data, many organizations simply do not know which data is sensitive and which data is not. Furthermore, there is often not a binary choice in classifying data. While cloud infrastructure services may have security vulnerabilities, their services for storage and processing are useful in many applications including Big Data storage and processing. In another paper [10], a network topology is proposed to handle Big Data classification problems in network intrusion detection using Machine Learning techniques. This network topology makes use of the Big Data technologies like Hadoop Distributed File System and public/private cloud infrastructure.

In this paper, we examine this network topology used to perform Big Data analytics on network intrusion data [10] that leverages the IaaS model for data storage. Figure 1 illustrates the same network topology presented in [10]. By leveraging cloud storage providers, the model (Figure 1) can function in as hybrid cloud system in order to store large amounts of data associated with Big Data in a more cost effective and efficient manner while maintaining a private system for data analysis. Due to the sheer volume of Big Data, we take the approach of treating all data as a whole and not attempting to rely on ever changing data classifications. We introduce a Data Key Store (DKS) into a hybrid computing ecosystem that uses data encryption and integrity techniques to address data confidentiality as well as verifying data retrievability from a public storage IaaS system.

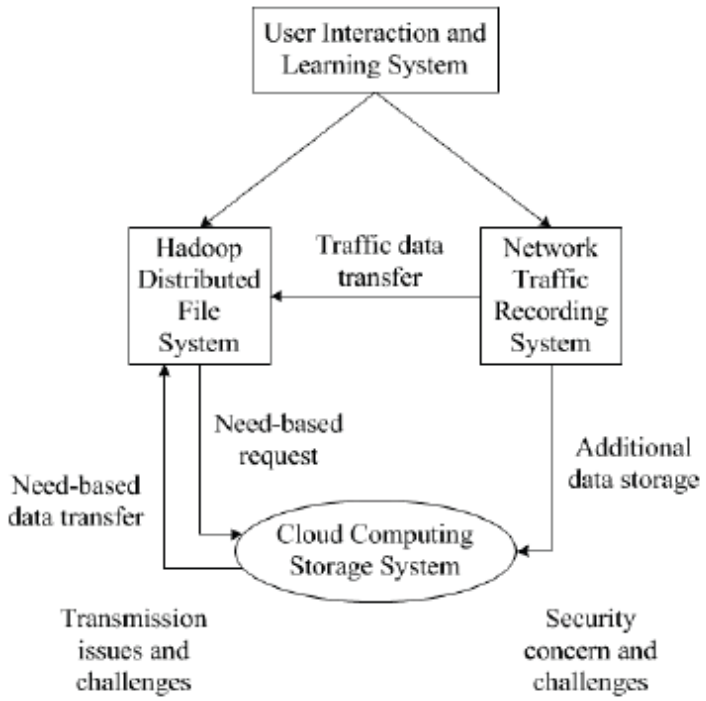

Figure 1. Big Data Network Topology

2. A HYBRID CLOUD FOR BIG DATA

The accepted definition of cloud computing identifies three deployment models: Private, Public and Hybrid [9]. While all three models pose their own challenges, the model put forth in this paper is focused primarily on the hybrid model in which computation occurs privately and data is stored both privately and publicly. We examine the topology outlined in Figure 1 that shows a hybrid cloud ecosystem used for Big Data analytics in which data that requires immediate computation is stored locally within the Network Traffic Recording System (NTRS). Data that can be computed at a later time is stored within a Cloud Computing Storage System (CCSS).

The public portion of the hybrid cloud in this model the CCSS - is used to provide additional storage to take advantage of the elasticity and savings afforded by a public cloud provider. This provides value to the Big Data problem due to the mere volume of data that can be produced while analyzing large amounts of network traffic. However, storing data that may be considered private causes security concerns in this model. While it is assumed that securing data in transit is already addressed in this model levering techniques common to the cloud computing environment such as SCP and SFTP, the issue of securing data at rest still poses a problem. While public cloud vendors may offer mechanisms to secure data at rest [2], it is assumed that more proactive steps must be taken to mitigate risks. In addition, there are certain performance advantages gained by customizing Hadoop for specific data processing, shown to be up to a 70efficiency [8]. These gains offer compelling reasons to keep data processing within the private cloud.

2.1 Storage IaaS Security Challenges

A significant challenge with regards to storage Infrastructure as a Service is determining the trust relationship between a provider and a consumer. In our case, the user is performing Big Data analytics on network traffic which may or may not contain sensitive information. If data is going to be stored with a cloud storage provider, a trust level must be established that can then be used to determine if the provider is capable of storing secure data and what mitigation must occur, if any.

Two valuable features of cloud computing services are mutlitenancy and elasticity. These two characteristics are what allow for the attractive prices of cloud computing while providing the efficiencies required to offer the service for a profit [4]. These are examples of commonplace techniques for cloud service providers that introduce risk as multiple tenants use the same resources.

3. DETERMINING TRUST

While providers may attempt to mitigate these risks, some level of risk remains due to the loss of total control by the user. Therefore, some level of trust between a consumer and provider must be established. This level of trust is used as a foundation for our model. In general, greater trust means the user must mitigate less risk. We assume that several characteristics, like those mentioned in section 2.1, can be ascribed a weight associated with the type of Big Data being

analyzed. Together these characteristics and weights can be combined to determine a trust level with a specific provider. This serves as a model that can be used to determine the level of encryption needed to safely store data. When working with large amounts of data associated with Big Data analytics minimizing computational overhead is valuable. Unnecessary data encryption on high volume, high velocity data would lead to inefficiencies.

3.1 Characteristics and Weights

In order to facilitate a trust determination we use a system of identifying characteristics of both the data and storage providers. We then ascribe weights to each of these characteristics that reflect a mitigation level. While this system is rudimentary, it is meant to serve as groundwork for future research into automatically determining these weights with various providers.

Two characteristics have already been discussed in the previous sections: muti-tenancy and elasticity. Other examples of characteristics include secure data clearing once data is deleted, 3rd party data storage [1], safe harbor, storage duration, etc. Depending on the type of Big Data being stored with an IaaS provider these characteristics may become valuable when determining the feasibility of levering a particular IaaS provider for data storage. Due to the ever-increasing number of possible characteristics one must consider when developing a trust level with a provider, our model remains generic enough to accommodate any number of characteristics. Unfortunately several characteristics currently require human intervention to make a determination. This is a challenge posed by the cloud services industry due to lack of standards and transparency.

Each characteristic is then ascribed a weight proportional to the importance of the characteristic in relation to the Big Data being analyzed. These weights can be controlled within the User Interaction and Learning System so they can be adjusted depending on the data being analyzed. For example, a public data set common to Big Data may have very low weights while analysis of secure network communications may have much higher weights. Health related data or Personally Identifiable Information (PII) may have an even higher weight. Our model is flexible enough that the weights of each characteristic may be changed depending on the type of data being analyzed.

3.2 Determining Trust Levels

Calculating risk in such a dynamic ecosystem poses significant challenges. In this paper we do not attempt to identify all possible risks but instead put forth a more abstract model that can account for an unknown number. As a base, the Equation 1 below is used to identify the risk associated with a single characteristic and the weight ascribed by producing a value α. Three parameters influence the efficiency of this model: α, c and *weight*. The relationship between these three parameters can be explained by a simple nonlinear model presented in Equation 1. The equation explains the relationship between the risk level (α) and importance level (weight) for a fixed mitigation level. Similarly, it also explains the relationship between the risk level (α) and mitigation level (c) for a fixed importance level (*weight*).

$$\alpha = c * weight \tag{1}$$

Here, the value *c* refers to an identified mitigation level between 0 (full mitigation) and 10 (no mitigation) performed by the provider. At this time the level is determined by manual analysis of the IaaS provider such as in place agreements, service terms and peers understanding and may be different between multiple providers. The *weight* refers to an importance of the characteristic as it relates to the Big Data, with 0 being no importance to 1 being the highest importance. Therefore an α of value 0 would have no risk while a value of 10 would be the highest risk.

The value β is an ensemble of all risk levels α, as shown in Equation 2.

$$\beta = \alpha_1 + \alpha_2 + ... + \alpha_n \quad (2)$$

A higher β value implies a lower overall trust associated with a provider in the CCSS. Since the value of β is the sum of an unknown number of α values, there is no upper bound defined for the possible risk. It can also be concluded that a completely trusted environment, such as a private cloud storage system, may have a β value of 0.

3.3 Defining EType Levels

The value β can then be used in association with identified encryption types (ETypes) to determine an appropriate level. Each encryption type has a max acceptable β value, as seen in Table 1 below.

Table 1. EType Example

Encryption Type (EType)	Max β value
1-XOR	0
2-DES3-CBC-MD5	3
3-RC4-HMAC	5
4-AES256-CTS	14

The example in Table 1 shows an EType of 1 that is only appropriate for use when the β value is 0, while the EType of 2 is appropriate for β values 0-3. The EType 4 is appropriate for all β values up to 14. A β value greater than 14 would mean that the provider cannot reasonably be used. It is important to note that higher ETypes are appropriate for use with all previous ETypes. This allows for both the appropriate level of encryption to be used as well as flexibility when selecting efficient algorithms. It is assumed that lower ETypes will allow Big Data analysis to be performed quicker due to decreases in computation for encryption.

4. NTRS TO CCSS

In this section we expand on the Network Traffic Recording System (NTRS) in Figure 1 to provide mitigations of security concerns and challenges when storing data in a CCSS. We do this by including a Data Key Store as part of the NTRS that manages data sent to the CCSS. An additional benefit of the Data Key Store model described in this section can also be used in order to provide proof of retrievability. Being able to periodically prove the ability to retrieve data becomes increasingly important as datasets grow and cannot efficiently be examined without significant burden [7], as in the case of Big Data. However, in an ecosystem using IaaS for

irreplaceable data storage, it is important to be able to prove a provider actually possesses the data and not just a placeholder. While the Data Key Store model can be used solely to handle encrypting data at appropriate levels, it can be extended to provide a form of retrievability proof.

4.1 Data Key Store

The Data Key Store (DKS) is used to manage the data stored in the IaaS provider as well as associated keys needed for encryption and decryption of data. We propose the DKS as part of the NTRS in Figure 1 as a mechanism to move data to the IaaS provider while overcoming security concerns and challenges. This actually introduces additional capabilities of a hybrid cloud system that may prove valuable when working with Big Data, such as allowing the DKS to use multiple public cloud storage services. Data can even be stripped across multiple storage providers allowing added resiliency for the data stored with an IaaS provider. However, these benefits are left for further research.

Each block stored has a 5-tuple value that includes blockID, PubID, Key, Hash and EType. These values are used to track a block of data (blockID), the data storage location (PubID), the encryption key (Key), a hash of the data (Hash) and the encryption type used (EType). This allows for flexibility to store data with multiple vendors, each with their own associated risk level and their appropriate encryption type. The HDFS can retrieve data from the CCSS and request decryption keys from the DKS.

4.2 Proving Big Data Retrievability

The DKS contains information that can be used in a basic proof of retrievability, while laying the foundation for more complex proofs [7]. The ability to prove an IaaS provider actually contains the data stored becomes more difficult as data size increases. Due to the loss of data control and varying trust levels it becomes more important to verify data is actually stored and not just a stub file or pointer to a 3rd party storage vendor that may no longer contain the data. However, with such large files, simply retrieving the data from the IaaS provider is not efficient.

Since the DKS has the ability to break data into blocks, it is also trivial to store a hash of the block. It is much more efficient for the DKS to periodically retrieve a single block and verify the hash. While this is much more trivial than the model shown by Juels and Kaliski [7], it does provide a simple proof of retrievability that can be extended with further research.

4.3 Intentional Injected Delay

Finally, we introduce the idea of intentionally injecting delay into the examined network topology for Big Data analytics [10]. Due to the nature of Big Data it is possible that the NTRS may see an extremely high volume of data at times. The data delay imposed by using a public cloud storage provider may actually become beneficial by introducing delay into the system and acting as a buffer for the HDFS. As discussed in section 2, the ability to process data in a customized Hadoop environment may provide distinct advantages over one size fits all implementations depending on the type of data being processed. In order for Hadoop to provide a simple mechanism for data processing it must be generic enough to process a wide range of data

types. This leads to default implementations proving inefficient for specific known data types [6]. However, this means that significant performance improvements may be gained by customizing Hadoop implementations [8]. These efficiencies may be realized when data is processed within the private ML environment. However, building a ML system that can withstand the high volume of data produced with Big Data can easily prove cost prohibitive. Intentional injected delay helps to solve that problem.

While it may seem like introducing this delay will create an ever increasing data backlog, this data can be processed when the data flow ebbs or the system expanded to handle the increased load. By directing data away from the HDFS at an appropriate threshold, the HDFS can focus on computing data of a higher priority without negative impacts of additional data strain while still allowing data to eventually be processed locally.

Injecting delay into the system has added benefits to smaller scale research systems. A smaller scale ML research environment may see significant oscillation in data flow. However, it is very conceivable that storing the amount of data produced would incur significant cost to store locally. Leveraging an IaaS provider for data storage results in a buffer for processing as well as data storage in an on-demand and cost effective manner. This would still allow smaller scale research systems to use commodity hardware and not have to invest in expensive equipment that would sit idle when not processing data. Injecting intentional delay has benefit in scenarios where data volume varies.

5. CONCLUSION

Using the model for determining trust with an IaaS provider allows data to be stored in both the appropriate location with the least amount of overhead. This adds value to the Big Data problem when introducing a hybrid compute model by helping to minimize the amount of overhead required encrypting and decrypting data stored with an IaaS provider. The inclusion of the DKS to the examined topology for Big Data analytics [10] also provides a foundation that can later be expanded to address more complex proof of retrievability and data resiliency, both of which become more important as the size of data grows. Extension of the described model to utilize multiple IaaS providers for data storage within a single topology may have added benefits to the Big Data model and further research is worthwhile.

6. REFERENCES

[1] Amazon web services llc. amazon s3 service level agreement. http://aws.amazon.com/s3-sla/, 2013.

[2] Amazon web services llc. amazon simple storage service (amazon s3). http://aws.amazon.com/s3/, 2013.

[3] U.s. general services administration. cloud first. http://www.gsa.gov/portal/content/190333, 2013.

[4] M. Almorsy, J. Grundy, and I. M¨uller. An analysis of the cloud computing security problem. In *the proc. Of the 2010 Asia Pacific Cloud Workshop, Colocated with APSEC2010, Australia*, 2010.

[5] W. B. Dai Yuefa, G. Yaqiang, Z. Quan, and T. Chaojing. Data security model for cloud computing. In *Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009) Qingdao, China*, 2009.

[6] H. Herodotou and S. Babu. Profiling, what-if analysis, and cost-based optimization of mapreduce programs. *Proc. of the VLDB Endowment*, 4(11):1111–1122, 2011.

[7] A. Juels and B. S. Kaliski Jr. Pors: Proofs of retrievability for large files. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 584–597. ACM, 2007.

[8] J. Lin and M. Schatz. Design patterns for efficient graph algorithms in mapreduce. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 78–85. ACM, 2010.

[9] P. Mell and T. Grance. The nist definition of cloud computing. national institute of standards and technology. *Information Technology Laboratory, Version*, 15(10.07):2009, 2009.

[10] S. Suthaharan. Big data classification: Problems and challenges in network intrusion prediction with machine learning. In *Big Data Analytics workshop, in conjunction with ACM Sigmetrics*, 2013.

[11] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan. Sedic: privacy-aware data intensive computing on hybrid clouds. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 515–526. ACM, 2011.