IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

Security in the Cloud

Akarsha B.M[1], Yogesh M J[2]

The National Institute of Engineering, Autonomous under Visvesvaraya Technological University, Mysore[1]

Assistant Professor, National Institute of Engineering, Mysore[2]

Abstract: Data security remains a top concern for the adoption of cloud-based delivery models, especially in the case of the Software as a Service (SaaS). This concern is primarily caused due to the lack of transparency on how customer data is managed. Clients depend on the security measures implemented by the service providers to keep their information protected. However, not many practical solutions exist to protect data from malicious insiders working for the cloud providers, a factor that represents a high potential for data breaches.
This paper presents the High-Performance Anonymization Engine (HPAE), an approach to allow companies to protect their sensitive information from SaaS providers in a public cloud. This approach uses data anonymization to prevent the exposure of sensitive data in its original form, thus reducing the risk for misuses of customer information. This work involved the implementation of a prototype and an experimental validation phase, which assessed the performance of the HPAE in the context of a cloud-based log management service. The results showed that the architecture of the HPAE is a practical solution and can efficiently handle large volumes of data.

Keywords: Cloud Computing, SaaS, Data Confidentiality, Data Anonymization, Performance.

1. INTRODUCTION

Enterprises are highly recognizing cloud computing's compelling economic and benefits. Pooling IT and virtualizing resources in cloud enables organizations to realize cost savings and boost up's deployment of new applications. Anyway the valuable business benefits cannot be enabled without defined data security challenges posed by cloud computing. Deploying confidential information and critical IT resources in the cloud raises concerns about vulnerability to attack, especially because of the anonymous, multi-tenant nature of cloud computing. Applications and storage volumes often reside next to potentially hostile virtual environments, leaving information at risk to theft, unauthorized exposure or malicious manipulation.Regulation made by government for data privacy and location tells about the additional concern of legal and financial consequences.

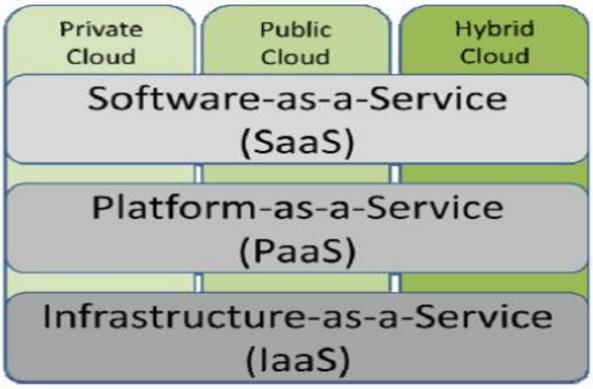
1.1 CLOUD COMPUTING DEFINED

The latest evolution in distributed computing is cloud computing which takes advantage on advance technology. Cloud raised due to the advent of nominal and faster microprocessors, RAM and storage which came with client-server model. As bandwidth became more lenient, low cost and speeder the networks interconnected to form internet. IT departments came up with the datacenters for the firewalls to protect data. Servers were accessed via browsers and a vendor was happy on the datacenters which could deliver the content remotely and almost immediately at the minimal cost. Scalable virtualization technology, cloud computing gives users access to a set of pooled computing resources that share the following attributes:

- Multi-tenancy
- Highly scalable and elastic
- Self-provisioned
- Pay-per-use price model

With respect to expenditure, cloud computing allows administrator to provide storage and operational site for launch within minutes or hours for historical costs.

1.2 TYPES OF CLOUDS


Private Cloud
Public Cloud
Hybrid Cloud
Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)

SaaS: is the software distribution model where applications are hosted by the vendor or service provider and made available to the users over the network. It is associated with pay-as-you-go subscription.
PaaS: is a set of software and development tools hosted on the provider's server. This offers an integrated development environment where even a developer is allowed to develop their application without any clue.
IaaS: is a single tenant cloud layer where the cloud computing vendors dedicated resources are only shared with contracted clients at pay-per-use fee.

2. CLOUD COMPUTING SECURITY CHALLENGES

In datacenters, Managers in IT come up with protocols and controls to build a strong perimeter round the infrastructure and data which they want to secure. This

DOI 10.17148/IJARCCE.2015.41124

configuration is relatively easy to manage, since organizations have control of their servers' location and utilize the physical hardware entirely for themselves.

2.1 MULTI-TENANCY

Multi-Tenancy is the architecture in which the software application serves single instance to multiple customers, each customer is called tenant. Tenants may be enabled to customize some part of the applications like User interface (UI), business rules but not the application source code.Risks in these environments is shared uniquely which are into the user's resource stack. For example, consumer will not know the neighbor identity, profile and intentions. The virtual machine running next to the consumer's environment could be malicious, looking to attack the other hypervisor tenants or sniff communications moving throughout the system. Because the cloud consumer's data sits on common storage hardware, it could become compromised through lax access management or malicious attack.

2.2 DATA MOBILITY AND CONTROL

Data stored in cloud can stay anywhere in the world, Data moving from physical servers to virtual memory makes it mobile. Storage administrators can easily reassign user information across data centers to facilitate server maintenance; Careful controls must be applied to data in cloud computing environments to ensure cloud providers do not break these rules by migrating geographically sensitive information across political boundaries. Further, legislation such as the US Patriot Act allows federal agencies to present vendors with subpoenas and seize data which can include trade secrets and sensitive electronic conversations without informing or gaining data owners' consent. [3]

2.3 DATA REMANENCE

Although the recycling of storage resources is common practice in the cloud, no clear standard exists on how cloud service providers should recycle memory or disk space. In many cases, vacated hardware is simply re-purposed with little regard to secure hardware repurposing. The risk of a cloud tenant being able to gather pieces of the previous tenants' data is high when resources are not securely recycled. Resolving the issue of data remanence can frequently consume considerable negotiating time while establishing service agreements between an enterprise and a cloud service provider. [3]

2.4 DATA PRIVACY

The public nature of cloud computing poses significant implications to data privacy and confidentiality.The data in cloud is usually stored in plain text, and few companies have an understanding of the sensitivity levels their data stores hold. In fact, a recent report by the Cloud Security Alliance lists data loss and leakage as one of top security concerns and data breaches are expensive to have. Recent laws, regulations and compliance frameworks compound the risks; offending companies can be held responsible for the loss of sensitive data and may face heavy fines over data breaches. [3]

3. RELATED WORK

Data confidentiality has been an active research area in recent times as it remains a top concern for adoption of cloud computing model. As a result, many different approaches have been proposed to ensure data security in the cloud. One proposed solution is to simply avoid external clouds and build in-house private clouds. In this scheme, companies try to retain the advantages of the cloud model by employing private/hybrid cloud initiatives, hence avoiding the issues of public clouds [8]. However, this approach is expensive and not affordable for most companies.

Another alternative for data protection is to use traditional cryptography techniques to encrypt all cloud data. While this technique might be a good solution to protect data when it is transmitted or stored at the vendor side, it is not appropriate for data which is used for computation. The problem is that this technique highly restricts further data use, such as searching and indexing. Some state-of-the-art cryptography works have offered more versatile encryption schemes that allow operations upon and computation on the ciphertext [13, 14, 15]. However they are still too slow to be viable for real-world applications. Another encryption approach is Silverline [16], which identifies and encrypts all functionally encrypt table data (any sensitive data that can be encrypted without limiting the functionality of the application in the cloud). However, the applicability of this approach is also limited because it assumes that web applications do not require access to raw data, which is rarely the case.

Our approach is closely related to the work described in [17], in the context of using data obfuscation to protect sensitive attributes. However, their solution requires cooperation from the service providers to implement logic on their side, situation which is not always feasible. Another approach related to our work is presented in [18], which also aimed to protect data from cloud service providers. Here, the authors describe three conditions to prevent that users' confidential information be collected by service providers. Firstly, separate software and infrastructure service providers. Secondly, hiding information about the owners of the data, and finally, the use of data obfuscation. Nevertheless, this flexibility is not always possible as it is often the case that the provider offering the software manages the infrastructure or platform service as well, so the user has no control over this.

4. PROPOSED METHOD

The context of our approach is shown in Figure 1. In most companies, different data sources exist within thesecure boundaries of the enterprise intranet. However, whenever an interaction occurs with a SaaS application,the company's data might leave the security of the intranet and be transferred to the SaaS provider. When thisscenario arises, a preferable situation is that the information could be protected before being sent. For this purpose,companies can implement their own methods to protect their critical

DOI 10.17148/IJARCCE.2015.41124

information in-house, before uploading theirdata to the cloud. This will keep the data safe from potential misuse by the service provider, while still retainingutility to be processed and analyzed. One technique that could be used for this purpose is data anonymization,which is the process of altering the original data in such way that it is difficult to infer anything private about theentities represented. This simultaneously limits the loss of information such that data is still meaningful permittingits analysis. Similarly, in some cases, users may need to access their original data. Once data is processed inthe cloud, the output can then be returned to the enterprise secure intranet and a reversibility mechanism canbe used to reveal the original values from the anonymized data.

Our approach, the HPAE, aims to fulfil thoseresponsibilities in Figure 1. The following sections describe in detail the components of our proposed architecture and implementation.

Finally, this work has the following goals:

• The development of an approach employing anonymization to protect confidential data on a public cloud.The objective is to protect customers' sensitive data, using the HPAE, from SaaS providers when data isprocessed in the cloud. Moreover, as the concept of confidentiality varies among users, our solution aims tobe flexible enough to allow users to configure their privacy policies.

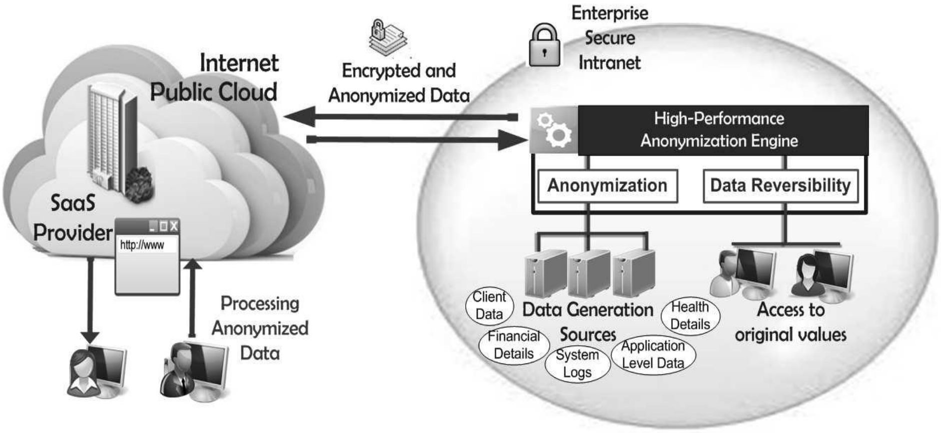


Fig 1: Contextual View of HAPE

• The design of an architecture that efficiently handles large volumes of data offering high-throughput and fast processing. Performance is a determining factor in the adoption of a new approach, thus we aim to provide a solution that offers good performance (in terms of throughput) such that it is practical and allows users to anonymize data on-the-fly. Furthermore, the design should be modular and extensible to facilitate the accommodation of new components, such as new anonymization techniques, new input/output types, more efficient libraries/data structures etc.

• The development of a prototype tool in Java to demonstrate our approach and measure the performance of our architecture. To facilitate ease of integration of the HPAE with existing systems inside organizations, our Java implementation will provide support for various types of input data sources as well as various types of output destinations.

4.1 HPAE ARCHITECTURE

This section describes the different components in the architecture of our prototype. The HPAE is the core component and it is composed by one-to-many Data Processor (DP) threads; although only a single DP is shown here, there may be as many DPs as input types. There are four stages in our approach: Configuration, Reading, Anonymization and Writing.

The prototype tool for the HPAE is still currently being developed; moredetails about the implementation details are provided in Section 4.1.5.

4.2 CONFIGURATION

In this stage two files are configured: the Engine Descriptor and the Rules Descriptor. The Engine Descriptor is an XML file containing the configuration parameters to initialize the HPAE and setvalues for the DPs. Among others, some of the parameters found in this file are: input sources/output destinations for DPs, the Rules Descriptor file for each DP and other technical properties like buffer size and the number of Event Handler threads to process the data. The Rules Descriptor is an XML file where the rules to identify sensitive information are defined (i.e. IP addresses, emails, etc.). The rules are defined in the form of patterns that can be position based, expressions or occurrences of specific strings. One file can be configured for each input source. Once the descriptor files have been configured, they are validated. The Configuration Parser is in charge of parsing the XML files and validating that parameter values are correct (i.e. positive numeric values, duplicate values, etc.). The output of this process is either; presenting a list of configuration errors that need to be fixed to the user, or passing the list of DPs to be run to the HPAE. The configuration of the descriptor files is currently performed manually as a GUI has yet not been implemented.

4.3 READING

The components used in the reading process will correspond to the selected type of input. In order to minimize the effort required by organizations in making changes to their current systems (which can have different technologies implemented), the HPAE supports the most common input sources and output destinations. The HPAE reads and writes data from/to databases, files, directories, TCP connections and messaging services like queues or topics. Support for the database type has yet not been implemented and remains as future work. To start the data processing, the corresponding Listener waits for data to become available. For example, in the case of TCP connections, the Listener will wait until a connection is made to the specified port number. Once a client connection is accepted, the corresponding Reader will start receiving data from the socket. The receivedpayload is encapsulated in an Event object, which can be a line of text in a file, a message from a queue/topic, atuple from a database, etc. These Events are sent to the Publisher, which claims the next available slot in the InputBuffer and adds the entry. Buffers are circular data structures used to exchange the data between the differentprocessing stages. These stages are asynchronously processed, meaning, one stage (i.e. Anonymization) does notneed to wait until the previous one is fully finished (i.e. a file fully read) to start its processing.

4.4 ANONYMIZATION

The Event Handlers are the threads that fetch the Events from the Input Buffer and process them. The logic topublish entries to the buffer and retrieve them is based on the Disruptor pattern [19], explained in more detail inSection 3.1.5. Each Event Handler processes one Event from the buffer. To apply the anonymization, the EventHandlers use the Rules Container and the Anonymizer. The Rules Container has the patterns that are used toidentify sensitive information. The Anonymizer contains the set of anonymization techniques to be applied. TheEvent Handlers perform the pattern matching; if a match occurs, the anonymization technique for that attribute is applied. Our prototype uses data substitution, extracting the sensitive attribute and replacing it with a new tokenin the Event.As some anonymization techniques are not reversible, it is desired to have mechanisms that allow re-identificationof the data regardless of the selected technique. In some scenarios, customers might require the original information, for example to perform root cause analysis of an incident investigation.

The HPAE aims to provide variousforms of Reversibility Mechanism to retrieve the original value such as keeping a translation table that tracks allthe transformations done to the data or building dictionaries on-the-fly. This latter could also provide a consistentanonymization (using the same value mapping) across multiple data streams, which can be useful to correlateinformation from several sources. Currently, the Reversibility Mechanism remains part of our future work.Since the HPAE can have multiple Event Handlers working in parallel, there is no guarantee on the orderthey will finish processing an Event. Thus, to ensure that the Events exit in the same order as they were received, the Sequencer is used. This component iterates the Input Buffer and retrieves the Events (once they have beenanonymized) to publish them in the correct order to the Output Buffer.

4. 5 WRITING

Based on the output type defined for the DP, the corresponding Writer is created. This component retrieves theanonymized Events from the Output Buffer and sends them to the specified destination. As an optional step, if theuser indicated in the Engine Descriptor file to gather performance statistics for the DP such as throughput, theseare calculated by the Statistics Collector. In the current implementation, statistics are collected and written to a CSV file for analysis.

5. CONCLUSION

There are many new technologies emerging with technical advancements and with the potential of enabling human lives easier. The enterprises are planning to deploy there applications on private and public cloud environment, though the security challenges to ne newly defined.By giving enterprises control over how and where data is accessed, it allows them the flexibility to move between cloud vendors without being tied to any one provider's encryption system. Information is defended by Secure Cloud versus manipulation. In this paper, we presented the HPAE, a practical approach that will enable organizations to implement their own controls to protect sensitive information from service providers in a public cloud environment. We demonstrated that the approach is practical by implementing a prototype and then validated its feasibility to efficiently handle large volumes of data. In our experiments, we achieved a throughput of more than 560,000 eps, using less than 35% of CPU and only 102 MB of memory. Delivered as an on-premise console or a Software-as-a-Service, SecureCloud represents a complete solution for safeguarding information in private clouds and public Infrastructure-as-a-Service environments.

REFERENCES

1. All material from "Security Guidance for Critical Areas of Focus in Cloud Computing v2.1", http://www.cloudsecurityalliance.org
2. NIST Cloud Model: www.csrc.nist.gov/groups/SNS/cloud-computing/index.html
3. The Need for Cloud Computing Security - Trend Micro USA
4. K. Kessinger and M. Gellman, "2010 ISACA IT Risk/Reward BarometerUS Edition," 2010.http://www.isaca.org/About-ISACA/Press-room/News-Releases/2010/Documents/2010 ISACA Risk Reward Barometer Results US.pdf
5. F. Gens, "New IDC IT Cloud Services Survey: Top Benefits and Challenges," 2009. [Online].Available:http://blogs.idc.com/ie/?p=730
6. S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing,"
7. Journal of Network and Computer Applications, vol. 34, pp. 1–11, Jan. 2011.
8. M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and Privacy in Cloud Computing: A Survey," 6th International Conf. on Semantics, Knowledge and Grids, pp. 105–112, Nov. 2010.
9. R. Chow, P. Golle, and M. Jakobsson, "Controlling data in the cloud: outsourcing computation without outsourcing control," in ACM Workshop on Cloud Computing Security, Chicago, IL, 2009.

DOI 10.17148/IJARCCE.2015.41124

10. iSMG, “Overcoming the Apprehension of Cloud Computing,” 2012. [Online]. Available: http://docs.ismgcorp.com/files/handbooks/Cloud-Survey-2012/Cloud Survey Report 2012.pdf
11. Cloud Security Alliance, “Top Threats to Cloud Computing,” 2010. [Online]. Available: https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf
12. Ponemon Institute, “2009 Annual Study: Global Cost of a Data Breach,” 2009. [Online]. Available:http://www.securityprivacyandthelaw.com/uploads/file/Ponemon COB 2009 GL.pdf
13. S. et al., “Data Breach Trends & Stats,” 2012. [Online]. Available: http://www.indefenseofdata.com/data-breach-trends-stats/
14. D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposium on Security and Privacy. IEEE, 2000, pp. 44–55.
15. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,”Advances in Cryptology Eurocrypt, no. 3027, pp. 506–522, 2004.
16. C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. thesis, Stanford University, 2009.
17. K. Puttaswamy, C. Kruegel, and B. Zhao, “Silverline: toward data confidentiality in storage-intensive cloud applications,” 2011.
18. M. Mowbray, S. Pearson, and Y. Shen, “Enhancing privacy in cloud computing via policy-based obfuscation,” The Journal of Supercomputing, pp. 267–291, Mar. 2010.
19. S. S. Yau and H. G.An, “Protection of users’ data confidentiality in cloud computing,” Proceedings of theSecond Asia-Pacific Symposium on Internetware, pp. 1–6, 2010.
20. LMAX-Exchange, “LMAX Disruptor High Performance Inter-Thread Messaging Library.”[Online].Available: http://lmax-exchange.github.com/disruptor/
21. L. Dignan, “Software development budgets on the rise, study finds.” [Online]. Available: http://www.zdnet.com/software-development-budgets-on-the-rise-study-finds-3040092512/

DOI 10.17148/IJARCCE.2015.41124