Software Defined Privacy

Florian Kemmer, Christoph Reich, Martin Knahl
Institute for Cloud Computing and IT Security,
Furtwangen University
Furtwangen, Germany
{Florian.Kemmer,Christoph.Reich,
Martin.Knahl}@hs-furtwangen.de

Nathan Clarke
Centre for Security, Communications and Network Research
University of Plymouth
Plymouth, United Kingdom
N.Clarke@plymouth.ac.uk

***Abstract*—Cloud Computing has been one of the most fascinating and influential technologies in recent years and has literally revolutionised the way we access computing resources by providing a virtually unlimited amount of resources instantaneously. Despite this seemingly amazing breakthrough, there are still some clouds on the horizon, preventing the paradigm from an even greater impact: While most technological issues are solved or nearly solved, two major areas still not only lack appropriate mitigations, but even a thorough understanding: Fear of privacy and security issues as well as a lack of control over own data remain the biggest obstacles towards a wider adoption of Cloud Computing. In this paper, we argue that existing solutions to protect users' privacy are not enough as they only focus on a single aspect at a time while being too complex. Instead, we present a novel concept called "Software Defined Privacy" which allows easy orchestration of existing tools to describe and enforce privacy requirements of an IaaS Cloud Customer.**

I. INTRODUCTION

The arrival of the Cloud Computing paradigm has nothing less than revolutionised the way data is stored, accessed and used for computation by providing instant access to virtually unlimited resources. Instead of using in-house hosted infrastructure, the cloud customer rents the required computation power or storage capacities provided be the Cloud Service Provider (CSP). This allows for more flexibility in both up- and downscaling and is economically desirable as only the actually used resources need to be payed; unused hardware in local data centres becomes a relic of the past.

The most commonly agreed on definition of Cloud Computing defines five core characteristics [1]: On-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. Moreover, it defines three service models, namely Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) – with a focus put on the last-mentioned for this paper.

With the advent of this new paradigm, there also came doubts: Outsourcing potentially sensitive information to third parties not only means loosing control over what's happening to this data, but also requires a lot of trust to these third parties as a privacy breach – intentional or unintentional – could have disastrous consequences. It comes as no surprise, therefore, that privacy and security issues are still perceived as the biggest challenges needed to overcome for the final success of Cloud Computing [2].

The remainder of this paper is structured as follows: Section II will give a brief overview of various existing software defined systems and outline their benefits. Section III will introduce a number of existing privacy threatening issues in Cloud Computing and potential mitigation approaches. Our novel architecture called "Software Defined Privacy" will be introduced in section IV. Section V will then refer back to the introduced privacy threats and demonstrate how the framework will be able to counter these issues either on its own or by incorporating existing solutions. Finally, a conclusion will be given in section VI highlighting the importance of a unified privacy protection mechanism.

II. SOFTWARE DEFINED SYSTEMS

Software Defined Systems (SDSys) exist in numerous forms, like Software Defined Networks, Storage, Compute resources or combinations thereof, such as Software Defined Data Centres or Software Defined Clouds. While the technical implementation of each of those technologies might differ, they all share the same goal: To abstract from physical infrastructure and potentially different hardware vendors and present a single point of access and configuration to the user [3].

Usually, this is done by decoupling components required for configuration (*control plane*) from those that are implementing the defined rules (*data plane*) providing a centralised configuration, but distributed enforcement. Besides the easier configuration, this approach also provides easier scalability by allowing to *scale-out* instead of *scale-up*.

While programmable networks aren't completely new, but have been considered and investigated before [4], only the advent of highly dynamic infrastructures like *Cloud Computing* has made equally flexible networks necessary and consequently led to a wider adoption of such technologies coined *Software Defined Networking (SDN)*

By itself, SDN is both beneficial and harmful for network security leading to the conclusion that it is "not really" secure at this stage [4], [5]. On the one hand, it allows for easier acquisition of network traffic samples to analyse and also enables users to implement required changes faster. On the other hand, however, it also makes networks more vulnerable to denial of service (DoS) attacks by providing a single target.

978-1-5090-3684-4/16 $31.00 © 2016 IEEE
DOI 10.1109/IC2EW.2016.34

Recently, more security-focused approaches like *Software Defined Perimeter* or *Software Defined Security* have been presented [6]. *Software Defined Security* utilises the previously mentioned decoupling of various layers to define security policies which are then enforced by *SDSec switches*, which either forward or drop matching packets.

III. PRIVACY ISSUES IN CLOUD COMPUTING

Security and Privacy issues have been around since the emergence of Cloud Computing and undergone intensive research since then. Despite all efforts, the state of security is "still puzzling" [7] and further research needs to be conducted to make the cloud a safe place for data.

This section features a selection of privacy threats conceivable in Cloud Computing infrastructure which will later serve as examples to demonstrate the usefulness of *Software Defined Privacy*.

A. Intra Host Attacks

One of the biggest concerns in Cloud Computing arises from the multi-tenancy nature, where multiple, potentially competing companies share the same physical infrastructure, sometimes with an imbalance of awareness of this fact. A variety of attacks exploiting these characteristics have been demonstrated over the years, ranging from the exploitation of bugs in the hypervisor to executing influence over other VMs in form of DoS attacks. Reviewing past research also shows, that simply hoping for the technology to evolve and "become secure" is utopian as there are always new attacks and bugs unveiled, which endanger the infrastructure [8], [9], [10].

In an ideal or naïve world, this problem could be solved by exclusively assigning hardware to customers, which, however, would lead to a number of negative side effects and loss of the defining characteristics of Cloud Computing: Massively reduced flexibility as well as decreased utilisation of the physical infrastructure and thus eventually higher costs for both provider and customer.

To mitigate some of these negative impacts a compromise would be conceivable: Various authors have described models based on the *Chinese Wall Security Policy* to mitigate the aforementioned issues while keeping the typical cloud characteristics in place [11], [12]. Using such an approach, the cloud customer would be able to select his area of business to ensure that no other, directly competing company, would be placed on the same hardware and thus possibly be able to run attacks. While the customer still would be potential target for other attackers, it could be argued that they wouldn't have much interest in harming this customer as no direct competition between them exists.

B. Transborder Data Flow

Not entirely unlike the multiple tenants on the same physical infrastructure, there are usually multiple data centres forming this single infrastructure. For large providers, these facilities are usually distributed around the globe to provide georedundancy, locations closer to the customer (and thus less latency) and to obey potentially differing law regulations. The last point is particularly interesting from a privacy point of view as some countries have stronger privacy regulations than others, which might consequently not qualify as a suitable location for providing services [13]. In light of the recent fall of the "Safe Harbor" agreement[1], this issue might become more prominent in the near future. By signing the "Safe Harbor" agreement, companies from the US could declare to obey to the stronger regulations in the EU and thus allowing companies from European countries to store and process data in their data centres located in America.

C. Unencrypted Archived Data

In the sense of this paper, "archived data" is all data that is permanently stored outside the virtual instance and *not* being continuously accessed, such as backups of any sort or snapshots of virtual machines. This type of data is usually not stored on the same infrastructure as the running virtual machines, but completely different hardware (e.g. Amazon's S3[2]): Enforcing separate hosts per customer is therefore one thing, ensuring different storage locations something entirely different. The most obvious solution is encryption of the data in question, which is a reasonable option given the non-permanent access.

D. Data Leakage

Inadvertent leakage of sensitive information is not a problem solely found in Cloud Computing, but can occur on every machine on connected to the Internet. In fact, until a few years ago, data leakage was mostly associated with corporate on-site networks with secret data leaving an employee's computer; either due to malicious attempts or due to malware that had infected the machine. Detecting such leaks was therefore mostly done at network perimeter devices.

With the increasing move into the cloud, defining these perimeters becomes increasingly difficult and routing all traffic from the cloud to local detection devices lacks both efficiency and dynamic. Detecting these forms of data breaches is necessary in order to prevent more information from being lost, but doing so automatically on the side of the CSP can by itself be an intrusion to the customer's privacy [14], [15].

First steps towards protecting this privacy and providing *Data Leak Detection as a Service (DLDaaS)* have been made [16], [17]. Using such a framework would allow the CSP to help preventing leakage of sensitive information without himself violating the customer's privacy.

E. Data Access Violation

Due to the dynamic nature of Cloud infrastructures, it often becomes difficult to keep configurations and access lists up to date: Access rights are granted/revoked, configurations are

[1]Press Release of the Court of Justice of the European Union: http://curia.europa.eu/jcms/upload/docs/application/pdf/2015-10/cp150117en.pdf

[2]https://aws.amazon.com/s3/

changed and sooner or later inconsistencies between the various cloud instances will arise or privacy threatening vulnerabilities form. Simply speaking, it is thus required to regularly verify that the existing configurations are a) consistent across all deployed machines and b) consistent with the existing privacy policy defined.

IV. SOFTWARE DEFINED PRIVACY

This section will introduce the new concept of *Software Defined Privacy* by describing the technological architecture below and then highlighting the importance of the user interface. While the concept generally works throughout all layers of Cloud Computing, we will focus on IaaS for now, which has been identified to be the fastest-growing model [11].

A. Architecture

Software Defined Architectures, such as *Software Defined Networking*, are usually split in three distinct layers, namely *Application*, *Control* and *Infrastructure* [18]. Software Defined Privacy follows this approach by specifying the same three layers.

On the *Infrastructure Layer*, there are the various components within a typical Cloud Computing system, such as virtual machines, hypervisors, storage systems, networks or others. This is, where changes will be made to by above layers in order to comply with privacy policies.

To orchestrate and apply these changes, an agent-based approach is used on the *Control Layer*: Lightweight agents are specialised to do one particular task and can easily be deployed (or undeployed) on systems either temporarily or permanently to quickly reflect changes in policies. These agents can be placed on either of the above mentioned components and change configurations thereof or install additional software that might be required.

On the *Application Layer*, the user is required to define his privacy needs. This is currently done using a rudimentary user interface during the instance creation (see fig 2), where the user can select policies applicable to his needs (see below for details on the user interface). Using an XML-based interface, it is possible to attach other, more sophisticated interfaces or let specialised "Privacy Officers" design a suitable set of rules for a particular VM.

Figure 1 depicts the proposed architecture with all *Software Defined Privacy*-specific components highlighted in red. The whole system is designed to be as little intrusive to existing infrastructures as possible and instead playing well with widely used software, such as OpenStack to ensure easy adoption.

B. User Frontend

At least equally important as the technological framework is the user interface as it represents the interaction point between user and software and ultimately defines usefulness and acceptance of Software Defined Privacy.

In various studies, it has been shown that fine-grained configuration possibilities do not necessarily lead to better configured systems. Instead, more often than not, they promote misconfiguration and thus less protected systems as systems tend to get too complicated and users become unable to cope with too many choices [19], [20].

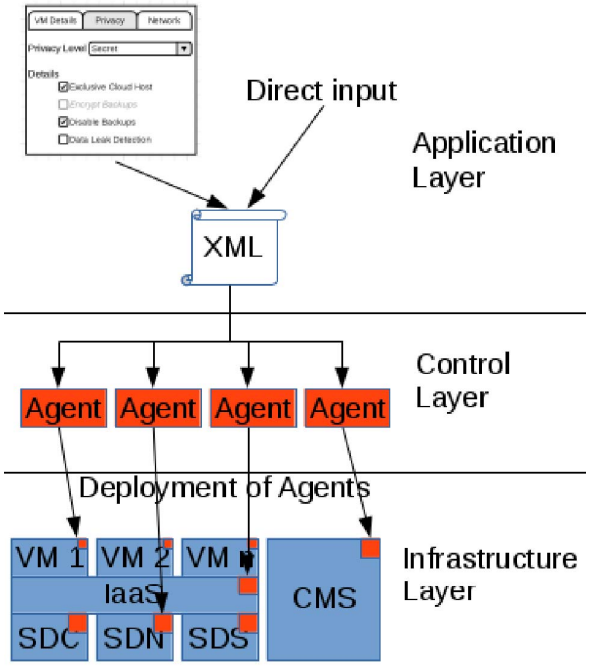

Fig. 1. Overview of the three layer architecture, which is similar to other Software Defined Services, such as SDN.

Building on these findings, we suggest a two-fold dialogue: On the first level, there is nothing more than a very limited number of options (aforementioned studies suggest three) to describe the overall sensitivity of a virtual machine, such as "secret"/"sensitive"/"unprotected" (in decreasing order of required protection). Based on the user's selection, appropriate defaults will be selected and correspondingly configured tools deployed. If the user then wishes to further define his needs in a more detailed way, this option will be given as well. The user is capable of either loosening the suggested mitigation methods or putting even more protection mechanisms in place.

Figure 2 shows a simplified dialogue during instance creation, where the user has chosen the highest default category ("secret"), which has automatically disabled backups and set the requirement of exclusive cloud hosts. However, the user has decided to not be in need of any sort of Data Leak Detection – neither as a service nor locally on the VM.

V. EVALUATION

Using the previously described privacy threats within Cloud Computing, we are now showing how *Software Defined Privacy* helps mitigating these risks.

A. Protection against Intra Host Attacks

To protect against attacks from other customers on the same machine, two protection mechanisms have been identified: a)

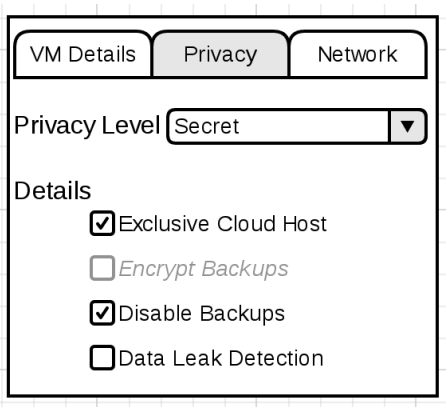

Fig. 2. VM creation dialogue that allows definition of privacy requirements for the given instance.

exclusive usage of physical resources for a single customer or b) semi-exclusive usage following a *Chinese Wall Policy Model*. While the former provides better isolation, it also removes many of the typical benefits of Cloud Computing and thus leads to potentially higher costs – for both CSP and cloud customer.

On the technical side, however, both approaches are similar in application: The original scheduler of the cloud management software, in our case OpenStack, gets replaced by a *PrivacyAwareDeploymentAgent*. More specifically, the agents provides a thin wrapper around the original scheduler and adds some critical functionality.

Before deploying a virtual machine, the Agent checks for any privacy policies associated with the instance. Following that, it looks for available hypervisors that match the requirement. In the first case, this would be either completely empty hypervisors or hardware only occupied by the same customer whereas in the second scenario any hosts without competing customer machines are valid.

```
for host in hosts:
  if host.vmCount == 0 or
    host.exclusivelyUsedByCustomer == True:
    validHosts.append(host)
```

```
for host in hosts:
  if host.vmCount == 0 or
    host.exclusivelyUsedByCustomer == True or
    mybusiness not in host.businessesDeployed:
    validHosts.append(host)
```

Finally, the virtual machine gets deployed on one of the limited set of hypervisors by the original scheduling algorithm.

B. Protection against Transborder Data Flow

The same *PrivacyAwareDeploymentAgent* can also be used to prevent virtual machines from accidentally being moved to other jurisdictions where the customer might not feel his data as well protected:

```
for host in hosts:
  if host.location in allowedLocations:
    validHosts.append(host)
```

Protection against transborder data flow as well as intra host attacks cannot be achieved by solely placing virtual instances correctly during their creation. All aforementioned verifications need also to take place, when instances are automatically moved (e.g. by the CSP to achieve better utilisation of the existing infrastructure). Moreover, the Agents must be capable of triggering a relocation by themselves: When the cloud customer removes a certain location from the list of allowed places while one of his instances resides there, this instance needs to be immediately transferred into a safe place

C. Protection against Unencrypted Archived Data

It has been shown that both backups and snapshots bear the potential to accidentally expose data. In order to prevent that, the *ArchiveProtectionAgent* will either enforce encryption of data or disallow creation of backups/snapshots completely.

To do so, it is again designed as a wrapper around the original functionality. In the scenario of complete denial, all incoming requests are blocked, so that neither the cloud customer can create backups/snapshots nor the cloud management software itself, e.g. when trying to automatically create regular backups.

In the first case, the *ArchiveProtectionAgents* requires the user to provide a public key. Using this key, the backup/snapshot is encrypted before being permanently stored on the provider's hardware. Even if the storage unit was compromised, the attacker would not be able to gain access to sensitive data within the backups.

D. Protection against Data Leakage

Data leakage can be detected in three different places, depending on the user's preference and resulting in different performance and perceived sensitivity: locally per instance, semi-globally across all customer instances or globally by the CSP for all customers' instances (*DLDaaS*). In either case, the ecosystem consists of two types of Agents: *DataCollectionAgent*s, that are used to collect and redirect outgoing data, and *DataAnalysisAgent*s, where the outgoing data is sent to and analysed.

Depending on the user's choice, the *DataAnalysisAgent* either resides on the same VM as the *DataCollectionAgent*, on a separate user instance dedicated to Data Leak Detection for the specific customer or is provided by the CSP. While more global approaches generally provide better performance (not least due to less redundancy), they also cause additional headaches as sensitive data is being analysed outside the customer's reach.

E. Protection against Data Access Violation

While many attacks and resulting data breaches are the result of evil forces (e.g. fraud by employees), a number of those accidents are possible to be traced back to erroneous configuration of systems, i.e. human failure. Keeping configuration files in sync with the original intent becomes

increasingly difficult as systems get more and more complex and distributed. Consequently, being able to detect misconfiguration early will allow to predict and prevent privacy breaches.

Let's consider the example policy "All web server traffic must be encrypted" (i.e. transmitted via HTTPs). Ideally, the web server would enforce this by not listening on port 80 at all. In this example, however, the company has recently switched its web server software (e.g. from Apache to nginX) and during that transition a new configuration file had to be created – which is now responding on port 80 as well and thus presents a potential threat to the system's privacy.

This improper configuration is detected by a *PrivacyBreachPredictionAgent* by parsing the configuration files in question and comparing them to the original privacy policy. A *PrivacyBreachDetectionAgent* will simultaneously analyse the web server's log files to determine if the privacy has already been breached or it is a theoretical leak to date. Finally, a *PrivacyBreachResolvingAgent* would give recommendations to fix the issue; in this case by changing the configuration file. Furthermore, last-mentioned agent is also capable of deploying temporary counter measurements, such as adding firewall rules to make the system comply with the intended policies.

In addition to user-defined policies, the Agents can optionally also verify compliance with existing best practises such as provided by the German Federal Office for Information Security for Apache web servers [21].

VI. CONCLUSION

In this paper, we have proposed a novel concept called "Software Defined Privacy" which is inspired by other software defined systems, but emphasises privacy. The systems allows orchestration and combination of various, already existing privacy protecting approaches by putting such tools in one single place and thus within easy reach of the user to allow him to perfectly reflect his privacy requirements in form of adequate policies. Despite having matching policies, the user doesn't need to be aware of any of the technical details or actual tools in use. Various examples in this paper have shown, how this enables the cloud customer to protect himself against against various attacks, such as Intra Host Attacks or transborder data flow.

Compared to other solutions geared towards security improvements, our approach is less intertwined with the SDN approach, but only shares the principle of decoupling configuration and enforcing of policies.

REFERENCES

[1] P. M. Mell and T. Grance, "SP 800-145. The NIST Definition of Cloud Computing," Jan. 2011.

[2] Top Threats Working Group, "The notorious nine: cloud computing top threats in 2013," *Cloud Security Alliance*, 2013.

[3] Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. Vouk, and A. Rindos, "Software defined cloud: Survey, system and evaluation," *Future Generation Computer Systems*.

[4] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2317–2346, Fourthquarter 2015.

[5] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2015.

[6] Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, and A. Rindos, "SDSecurity: A Software Defined Security experimental framework," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, Jun. 2015, pp. 1871–1876.

[7] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, "Security issues in cloud environments: a survey," *International Journal of Information Security*, vol. 13, no. 2, pp. 113–170, Apr. 2014.

[8] J. S. Reuben, "A survey on virtual machine security," *Helsinki University of Technology*, vol. 2, p. 36, 2007.

[9] N. Gruschka and M. Jensen, "Attack Surfaces: A Taxonomy for Attacks on Cloud Services." IEEE, Jul. 2010, pp. 276–279.

[10] A. R. Riddle and S. M. Chung, "A Survey on the Security of Hypervisors in Cloud Computing." IEEE, Jun. 2015, pp. 100–104.

[11] Y. Fairweather and D. Shin, "Towards multi-policy support for IaaS clouds to secure data sharing," in *2013 9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*, Oct. 2013, pp. 31–39.

[12] S. Yu, X. Gui, J. Lin, F. Tian, J. Zhao, and M. Dai, "A Security-Awareness Virtual Machine Management Scheme Based on Chinese Wall Policy in Cloud Computing," *The Scientific World Journal*, vol. 2014, pp. 1–12, 2014.

[13] J. S. Babu, K. Kishore, and K. N. Kumar, "Migration from Single-Cloud to Multi-Cloud Computing," in *International Journal of Engineering Research and Technology*, vol. 2. ESRSA Publications, 2013.

[14] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 1, pp. 51–63, Jan. 2011.

[15] J. Kim, J. Hwang, and H.-J. Kim, "Privacy Level Indicating Data Leakage Prevention System," *International Journal of Security and Its Applications (IJSIA)*, vol. 6, no. 3, pp. 91–96, 2012.

[16] X. Shu and D. D. Yao, "Data Leak Detection as a Service," in *Security and Privacy in Communication Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, A. D. Keromytis and R. D. Pietro, Eds. Springer Berlin Heidelberg, 2013, no. 106, pp. 222–240.

[17] X. Shu, D. Yao, and E. Bertino, "Privacy-Preserving Detection of Sensitive Data Exposure," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1092–1103, May 2015.

[18] Open Networking Foundation, "SDN architecture," Jun. 2014.

[19] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures," in *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC '09*, Dec. 2009, pp. 711–716.

[20] I.-H. Chuang, S.-H. Li, K.-C. Huang, and Y.-H. Kuo, "An effective privacy protection scheme for cloud computing," in *2011 13th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2011, pp. 260–265.

[21] BSI, "Sicheres Bereitstellen von Web-Angeboten mit Apache," 2008.