Evidence and Cloud Computing: The Virtual Machine Introspection Approach[*]

Rainer Poisel[1†], Erich Malzer[2], and Simon Tjoa[1]
[1]*St. Poelten University of Applied Sciences*
St. Poelten, Austria
{rainer.poisel, simon.tjoa}@fhstp.ac.at
[2]*Open Networks*
Vienna, Austria
em@ong.at

Abstract

Cloud forensics refers to digital forensics investigations performed in cloud computing environments. Nowadays digital investigators face various technical, legal, and organizational challenges to keep up with current developments in the field of cloud computing. But, due to its dynamic nature, cloud computing also offers several opportunities to improve digital investigations in cloud environments. The enormous available computing power can be leveraged to process massive amounts of information in order to extract relevant evidence. In the first part of this paper we focus on the current state-of-the-art of affected fields of cloud forensics. The benefit for the reader of this paper is therefore a clear overview of the challenges and opportunities for scientific developments in the field of cloud forensics. As this paper represents an extended version of our paper presented at the ARES 2012 conference, we describe digital forensics investigations at the hypervisor level of virtualized environments in greater detail. cloud computing setups typically consist of several virtualized computer systems. Therefore we introduce the reader to the topic of evidence correlation within cloud computing infrastructures.

Keywords: Cloud Computing, Digital Forensics, Cloud Forensics, Hypervisor Forensics, Evidence Correlation

1 Introduction

In recent years, cloud computing has gained vastly in importance. It has been introduced to optimize the general usage of IT infrastructures. cloud computing is a technology that evolved from technologies of the field of distributed computing, especially grid computing [2]. According to NIST [3], "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e. g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

The European Union highlights in their digital agenda [4] the high value of cloud computing for businesses and the governmental sector. An important factor to reach the full potential delivered by cloud computing techniques is the reduction of uncertainty which is currently addressed by EU's cloud strategy [5].

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 4, number: 1, pp. 135-152

[*] This paper is an extended version of the work originally presented at the 7th International Conference on Availability, Reliability and Security (ARES'12), Prague, Czech Republic, August 2012[1].

[†] Corresponding author: Rainer Poisel, St. Poelten University of Applied Sciences, Matthias Corvinus-Straße 15, A-3100 St. Poelten, Austria, +43 2742 313 228 637, Email: rainer.poisel@fhstp.ac.at

Experts agree that there will be a substantial growth in the field of cloud computing over the next few years. According to Kazarian and Hanlon [6], 40% of small and medium businesses (SMBs) from different countries are expected to use three or more cloud services and migrate their data into the cloud. In 2010, Gartner [7] released a study which forecasted the cloud service revenues to reach 148.8 billion in 2014 (compared to 58.6 billion in 2009). Beside the usage of cloud computing in the economic sector it is increasingly used in a governmental context (e.g. [8, 9]). Paquette et al. discuss in their work [9] specific risks which have to be considered for cloud technologies in government use.

Carlton and Zhou [10] state that cloud computing is, from a technical point of view, a combination of existing technologies. People have difficulties to capture the big picture: for managers and customers of cloud services the idea is similar to exchanging information through web-based user interfaces. Others view the concept as being an extension of the timesharing concept from the 1960s. cloud providers sell services based on different business models (also referred to as “service models”): Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS) [11, 12]. With SaaS, the customer uses applications which are provided by the service seller (e. g. web-based e-mail services). With PaaS, the service seller then provides his infrastructure (servers, operating systems, network, etc.). The customer is able to write/use his own applications using the application programming interface made available by the provider. IaaS enables the user to use and run software of his choice (e. g. operating systems). The service seller provides the customer with the necessary infrastructure (servers, network, storage facilities, etc).

Depending on the level of access to the underlying cloud infrastructure the following types of clouds have been categorized [11, 13]: private clouds, community clouds, public clouds, and hybrid clouds. In “private clouds” the infrastructure is operated on behalf of a single entity. Usually the infrastructure is located in the premises of the organization. “Community clouds” refer to cloud deployments where the infrastructure is shared by several organizations. In “Public clouds” one or more providers run the infrastructure and make it available to anybody who wishes to pay for the service. “Hybrid clouds” refer to setups which are formed out of two or more cloud infrastructures. These in turn can be private, community, or public clouds.

Another upcoming trend is the usage of cloud infrastructures for criminal activities. In line with legal cloud business models, so called crime-as-a-service [14, 15] has been introduced as term for performing malicious activities in the cloud.

The shift in intercommunications and interaction between IT systems poses new challenges for digital forensics investigations. cloud Service Providers (CSPs) often do not let their customers look behind their “virtual curtains” [16, 17]. Vendor dependent implementations, multiple jurisdictions and proprietary data exchange formats [18] bring digital forensics into a deeper crisis as it is already facing [19]. Ruan et al. [20] defined cloud forensics as being a cross discipline between cloud computing and digital forensics. It is further recognized as a subset of network forensics [21]. Network forensics deals with investigating private or public networks and as cloud computing is based on broad network access it should follow the main phases of the network forensic process. Delport et al. [22] deem cloud forensics to be a subset of computer forensics as clouds consist of several nodes which are computers. This means that cloud forensics combines both, computer forensics and network forensics [23].

Ruan et al. [20] further extended the definition of cloud forensics across three major dimensions: technical, legal, and organizational. The technical dimension describes the set of procedures and tools which are utilized to carry out the digital forensics process in cloud environments. The organizational dimension refers to the fact that cloud computing involves at least two parties: CSPs and cloud customers. Further it is possible that CSPs outsource some of their services to other CSPs. The legal dimension refers to multi-jurisdiction and multi-tenancy challenges. Both fields have been exacerbated in cloud environments. Existing agreements and regulations have to be adopted for forensics activities to not breach any jurisdictions or confidentiality measures. Digital forensics can be divided into live and dead analysis [24]: while the former refers to investigations being performed while systems are running, the latter investigates systems that are in powered-off state. cloud forensics involves both disciplines as it is possible to acquire memory dumps without changing systems' states. Investigating hard disk or storage images from virtual machines can be performed with techniques known from the field of dead or post-mortem analysis.

This paper is structured into two parts. First we focus on the current State-of-the-Art of affected fields of cloud forensics. In the second part, based on the current State-of-the-Art, related challenges and opportunities are identified in order to derive and describe open research problems.

2 State of the Art of Cloud Forensics

This chapter describes the State-of-the-Art of affected fields of digital forensics investigations in cloud environments. Further, this chapter higlights possible challenges and opportunities for cloud forensics.

2.1 Existing Digital Forensics Frameworks

Digital investigations have to consider various perspectives (e.g. legal perspective, technological perspective) in order to be successful. In order to coordinate the efforts between the various stakeholders, there exist a variety of publications dealing with procedures how to handle, analyze, document and present digital evidence. The presented work in this subsection contains well-known and well-established guidelines which are not specifically tailored to cloud computing. To some extent the principles introduced are also valid for cloud technology. However, an adaption of the organizational frameworks has to be considered to deal with the new challenges arising from the usage of cloud computing.

In the First Responder's Guide for Electronic Crime Scene Investigations [25], the forensic process is split into the four phases, (1) collection, (2) examination, (3) analysis and (4) report. The first phase is dedicated to capture electronic evidence. Thereafter, in the examination phase content and state of evidence is documented and the evidence is examined concerning hidden and obscured information. The last step of the second step is to reduce the information. In the analysis phase the evidence is analyzed concerning the relevance to the case. While examination is a technical task, analysis is usually conducted by an investigation team. Finally, in the last step reporting takes place [25].

NIST SP800-86 [26] shows how digital forensics can support incident handling. This publication focuses tackles digital forensics mainly from an IT perspective, not a legal perspective. The forensics process uses the phases of [26].

Further widely-used digital forensic frameworks include the digital forensics framework of the Association of Chief Police Officers (ACPO) [27] and the DFRWS (Digital forensics Research Workshop) Investigative Process model [28]. Cohen proposes, in [29], a model consisting of the seven phases: identification, collection, transportation, storage, examination and traces, presentation, and destruction. Ke [30] describes the application of the SABSA model to the digital forensics process to obtain forensically sound evidence. More information on digital forensics frameworks can be found in [31].

[32] present in their paper an iterative framework based on the well-established and widely accepted work of [33] and NIST SP800-86 [26]. The proposed framework comprises the phases *Evidence Source Identification and Preservation*, *Collection*, *Examination and Analysis* and *Reporting and Presentation*. Possible iterations of the process are initiated by the Examination and Analysis step.

2.2 Investigation of Cloud Infrastructures

According to Zimmerman and Glavach [34], the technology of cloud computing is not new. It is a new way of providing applications and computing resources on demand. Therefore the technology seems a perfect solution for smaller businesses that do not have the necessary resources to completely fulfill their IT needs [35, 36]. Further, it allows private end users to utilize massive amounts of computing resources at affordable prices. However, the introduction of new technologies poses new challenges for the digital forensics investigator [37]. Grispos et al. show “how established digital forensic procedures will be invalidated in this new environment” [38]. They propose research agendas for addressing the new challenges depending on the investigation phase. As mentioned in the previous section there exist several organizational digital investigations frameworks. In the following the different investigation steps: identification, preservation, examination, and presentation are elucidated regarding their implementation for the investigation of cloud environments.

Identification, Preservation, and Acquisition: Grispos et al. outline in [38] the lack of frameworks to determine which elements were affected by IT specific crimes. The usage of conventional intrusion detection systems in the context of cloud computing infrastructures has been proposed by several authors [38]. The preservation and acquisition step deals with evidence collection from computer based systems. The increasing storage capacity of devices and computer systems are everlasting challenges in digital forensics investigations [38]. With the introduction of cloud computing systems this challenge is still ubiquitous: the elastic ability of cloud computing infrastructures allows the user to request additional data storage in a limitless fashion.

The chain of custody documents how evidence was handled in the context of the digital investigations process [39]. The documentation describes how evidence was collected, analyzed, and preserved to be approved in court. Due to the remote nature of cloud computing scenarios, assumptions that have been made with the investigation of traditional computer systems are not valid anymore [40]. Investigators usually had physical access to traditional computer systems [34]. Therefore they were able perform a live analysis or to remove storage devices for analyzing them in a forensics laboratory. Storage devices are accessed through a computer network. Digital investigators have to obtain control of cloud services before investigating them [20]. Depending on time an investigator requires to gain control of such a service, relevant evidence can be destroyed (deliberately or accidentially) by both, the service user and the cloud provider [38]. In this regard, IaaS deployments provide much more useful information for digital forensics investigations than PaaS or SaaS setups [16, 41]. With PaaS or SaaS deployment scenarios, customers do not have any control of the underlying operating infrastructure. The amount of information from servers is limited and therefore, the client has to contribute to the investigation process. Besides the technical challenges, the lack of regulatory and legal frameworks complicate meeting the chain of custody requirements [42].

In forensics, ’live’ acquisitions and investigations allow to obtain data stored in non-persistent memory such as process information or active network connections [43] as well as temporary data, such as file locks or web browsing caches [16, 38], RFC3227 [44] explains several best practices regarding live investigation of systems in case of security incidents.

However, traditional forensics guidelines require storage images to be forensically sound. Therefore bit-by-bit copies including a check sum are made from digital storage devices from instances in “dead” state (the system has been shutdown) to proof the unadulteratedness of digital evidence [27]. Traditional search and seizure procedures may be impractical for performing digital investigations in cloud computing environments. Digital evidence is stored in cloud data centres, desktop computers or mobile phones which could be out of physical control by the digital investigator [45]. As it is almost impossible to make a bit-by-bit copy of storage devices [34] the ACPO guidelines are rendered pointless when it comes to complete authenticity of digital evidence in cloud environments. Acquiring all storage devices from such a setup would be too time consuming for investigators and too disruptive for CSPs [38]. Usually cloud users are only offered remote access to the logical representation of their data. In most cases, the underlying physical infrastructure is transparent for the user. In the future, new methods will be needed to allow partial recovery of data from physical devices in accordance with accepted forensic principles.

Therefore, forensics tools have to be hybrid of the current live and post-mortem analysis methods [34]. There will be a need for intelligent tools that note and predict artefacts based on heuristics. Delport et al. outline in [22] that it might be necessary to isolate cloud instances in case they have to be investigated. The problem associated with isolating cloud instances is the integrity of data intended for digital forensics investigations [35].

Basically, methods for clearing include moving uninvolved instances or suspicious instances to other nodes. This way the CIA of other instances is protected, but it might result in loss of possible evidence. However, by moving instances, evidence is protected from being tampered by these moved instances. Delport et al. [22] presented different techniques to isolate instances of cloud environments.

Instance relocation means moving an instance inside a cloud environment by moving the data logically or by creating new and destroying old instances. Server farming refers to putting up a spare instance which offers the same functionality as the instance intended for digital investigations. By Sandboxing programs can run in an environment which they cannot escape. Man in the Middle (MitM) refers to placing an entity between a sender and a receiver. In the field of digital forensics this entity is placed between the cloud instance and the hardware of the cloud. Delport et al. [22] conclude that none of their presented approaches fulfils every requirement for the investigation of cloud environments. However, depending on the case techniques may be combined to gain explicit access to a cloud instance. In his paper, Yan [46] describes the basic architecture of a Cybercrime Forensic System which can be deployed in cloud computing set ups. It interacts as an extra layer which is placed between clients and actual services offered by the cloud.

The usage of cryptography in cloud environments poses additional challenges. CSPs offer encryption as a security feature to their customers. All data is encrypted on the client's side. The key to the encrypted data is never stored in the cloud environment [47].

Deleted data represents another major challenge due to the volatility and elasticity of cloud environments. On one hand, data that has remotely been requested to be deleted can be a rich source of evidence as it can still be physically existing [38]. On the other hand it depends on the CSP how to proceed in the event of a user requesting his data to be deleted [16, 34] (e. g. Google's policy includes the deletion of such data from both, its active and replication servers as well as of all pointers to this data).

Reilly et al. [40] also mentioned the lack of tool support for dealing with digital investigations with cloud data centres. Currently, most tools are intended for examining data from traditional computer setups such as office or home computers. Taylor et al. [45] recommended to update existing tool suites such as EnCase or FTK to account for new developments in the field of cloud computing.

Examination and Analysis: Forensic tool suites such as The SleuthKit, FTK or EnCase perform "pattern matching" and "filtering" of data that is existing in different types of memory. Evidence in cloud is manifold and will likely be similar to evidence found in traditional computer setups [38]: office application documents, file fragments, digital images, emails, and log file entries [48]. Checksums are used to verify the integrity of objects (disk images, files, log entries, etc.) in the cloud. Detecting file signatures of files in question or files which should be excluded from a digital forensics investigation are crucial for the filtering process. Hegarty et al. [49] describe a method for adapting existing signature detection techniques for their usage in cloud environments. To detect files with a specific hash value a so called "initialiser" submits the target buckets (storage units of a cloud customer) as well as the hash value to a so called "Forensic Cluster Controller" which in turn distributes the job of finding files with that has value to so called "Analysis Nodes".

In the future investigating cloud infrastructures may be a task performed by cloud deployments. However, cloud customers may access applications offered in the cloud from a myriad of different computer setups (mobile phones of different make, desktop PCs with different operating systems, etc.) [45].

Presentation: Digital evidence can be utilized in several ways: it can be submitted to court in the form of a report [28] or it may be used by an organization to improve corporate policies and support future investigations [50]. Grispos et al. [38] highlight the need for a standard evaluation method for cloud forensics so that cloud forensics investigation results pass the Daubert principles [51]. Another challenge arises from explaining the cloud computing concept to a jury in court [40]. It may be difficult for a jury member to comprehend the concept as jury members will usually only have basic knowledge of how to use home PCs.

2.3 Digital Investigations using Cloud Infrastructures

According to cloud security alliance [52], industry is heading forward to create Security-as-a-Service (SecaaS). The authors identified the following ten domains that are likely to interest consumer in the future: (1) Identity and Access Management Services; (2) Data Loss Prevention; (3) Web Security; (4) Email Security; (5) Security Assessments; (6) Intrusion Management, Detection and Prevention (IDS/IPS); (7) Security Information and Event Management; (8) Encryption; (9) Business Continuity and Disaster Recovery; (10) Network Security. Within one of these domains the authors identify the requirement to "...provide customers with forensics support...". This opinion is also supported by Ruan et al. [20] who derive from the emerging trend to security-as-a-service that forensics-as-a-service will gain importance in cyber criminal investigations by providing massive computing power.

Reilly et al. [40] take the discussion of the usage of cloud technologies for forensic investigations one step further and highlight the benefits delivered by the usage of cloud computing for digital investigations. The major advantages identified by the authors include large-scale storage, high availability and massive computing power. Roussev and Richard [53, 54] recognized the need for distributed forensics at an early stage. In their paper [55] they formulated the following requirements that should be satisfied by a distributed digital forensic toolkit: Scalability, platform-independence, lightweight, interactivity, extensibility and robustness. As cloud technologies can meet the abovementioned requirements, Roussev et al. evaluate in their paper [55] the feasibility and applicability of MapReduce for forensics applications. Map Reduce [56] was developed by Google in order to facilitate large scale computing. Phoenix [57] and Hadoop [58] are well known implementations of Google's MapReduce model. In their paper, the authors present their prototype, called MPI MapReduce (MMR), which is based on the Phoenix shared memory implementation. In order to test the performance of the prototype they implemented three Hadoop samples (wordcount, pi-estimator and grep) for MMR.

Cohen et al. introduce in [59] their GRR Rapid Response framework which pursues the objective to support live forensics within in an enterprise. The framework is designed to be highly scalable and is available for all common platforms. The proposed architecture is supported by an open-source prototype that is available [59].

Hegarty et al. present in their paper [49] the distributed calculation of file signatures if analyzing distributed storage platforms. Their proposed architecture consists of the three components: initializer, forensic cluster controller and analysis nodes.

Distributed computing power for password recovery or hash cracking is already well established. Various publications (e.g. [60]) and tools (e.g. Distributed Network Attack by AccessData [61, 62]) are devoted to this significant subject. eDiscovery applications which are also an important component in an digital investigator's daily business are already available for cloud implementations. An example is the open source eDiscovery software FreeEed [63].

2.4 Digital Evidence in Cloud Computing Environments

The introduction of cloud computing provided a change of paradigms to the distributed processing of digital data. In their paper Taylor et al. [64] focuses on the legal aspects of digital forensics investigations. They concluded that due to the increasing number of interacting systems the acquisition and analysis of digital evidence in cloud deployments is likely to become more complex. The data could be encrypted before being transferred to the cloud or it could be stored in different jurisdictions resulting in data being deleted before investigators have access to it [11].

Flaglien et al. [65] evaluated currently used formats for handling digital evidence against criteria identified in recent research literature. Recent developments with a focus on evidence exchange have been presented. Formats intended for storing evidence from highly dynamic and complex systems are characterized by incorporating additional information which can be processed by data mining tools.

Birk [16] and Wegener [41] mentioned digital evidence to be in one of three different states: at rest, in motion or in execution. Data at rest is stored on storage media. In this case it does not matter if the data is allocated to a file or if it has been deleted. Data in motion is usually data that is transferred over a computer network. Data that is neither in rest nor in motion is referred to as to be in execution. Usually this means process data that has been loaded into memory. In cloud environments evidence can be found on several sources: the virtual cloud instance (where the incident happened or originated), the network layer, and/or the client system [34, 16]. Especially in SaaS setups evidence can be found on client systems.

Lu et al. [66] proposed to adopt the concept of provenance to the field of cloud computing. As a data object is able to report who created it and modified its contents, provenance could provide digital evidences for post investigations. However, up to now, provenance is still an unexplored area in cloud computing. Provenance information would have to be secured in cloud environments as leaking this information could breach information confidentiality and user privacy. Marty [48] follows a similar approach. CSPs and application providers utilize logging facilities to generate and collect relevant data to support the digital forensics investigation process. The sources for logging can be manifold: "business relevant logging covers features used and business metrics being tracked" [48]. Operational logging covers errors that concern a single cloud customer, critical conditions that impact all users, system related problems, etc. forensics investigations are supported by security logging which focuses on login information, password changes, failed resource access and all activity that is executed by privileged accounts.

Cloud customers lose control over their data and executions in case they outsource the execution of business processes to the cloud [67]. Accorsi [68] stated that this problem could be overcome with remote auditing. Data analytics perform traditional audits remotely by assess and report on the accuracy of financial data. This requires the introduction of an additional service model: business-process-as-a-service (BPaaS). It is based on the SaaS provision model and provides methods for modelling, utilizing, customizing, and executing business processes in cloud infrastructures. Access to the physical systems is neither possible nor necessary: external auditors will have access to both the auditee's system and the auditee's compartment in the cloud. Then it is possible for the auditors to employ remote auditing, thus addressing the inherent loss of control.

3 Hypervisor Forensics

This section describes the process when acquiring data in a forensically sound manner from virtualized environments. New environments and technologies pose new challenges to researchers and digital investigators. On the other tack, hypervisors allow access to computing ressources on a low-level without chaning the system's state. Thus, traditional limitations known from the field of live data acquisition can be overcome using these new technologies.

3.1 Utilizing Virtual Machine Introspection in Forensics

Hypervisors (also referred to as "Virtual Machine Manager" or "VMM") can be understood as a host operating system which performs the allocation of computing resources such as memory, CPU, disk I/O and networking among operating systems that are running as "guest operating systems" [21]. As hypervisors build the bridge between guests and physical computer hardware, all data that is processed has to pass through the hypervisor before it can access physical devices (e. g. network interface cards, CPU . . .).

The usage of data from hypervisors to prove various actual situations has been proposed in previous research papers [69, 70]. The terminology has been referred to as "virtual machine introspection" (VMI) and data gathered from this level of access supported the operation of Intrusion Detection Systems (IDS). It is suitable for investigating cloud infrastructures as long as there is access to the Hypervisors. Thus, it is not suitable for investigation of Public clouds in case access to the hypervisor is denied or in case that infrastructure components are located in remote regions, such as given in the Amazon cloud or Google cloud Platform.

As known from digital forensics investigations on physically available devices volatile data might be lost in case cloud instances are shut down [16]. One example for such a scenario would be Amazon AWS EC2 cloud instances. Before shutdown persistent data would have to be stored in long time storage containers such as Amazon Simple Storage Service (S3) or Amazon's Elastic Block Storage (EBS). At the time of writing, approaches of how to interface Virtual Machine Monitors are product-specific. Currently, Xen is one of most widely used hypervisors [24]. Payne and Lee [71] focused on the development of an abstract monitoring architecture. Their programming library "XenAccess" has been released as an open-source project in 2006. Four years later the source-base has been forked: the project is currently released as another open-source programming library "LibVMI". The library is "focused on reading and writing memory from virtual machines" [72]. Therefore monitoring applications can access the memory state, CPU registers and disk activity of target operating systems in a safe and efficient manner. Memory can directly be read during runtime of virtual machines. Thus, it is possible to create memory dumps for further processing. In order to meet safety and integrity requirements target VMs can be paused in order to eliminate the chance of acquiring inconsistent snapshots. The library itself is written in C and comes with a Python wrapper to be able to integrate access to VMs to Python scripts.

Obtaining high-level information from low-level information found in memory images is referred to as the "semantic gap" problem [73] in recent research publications [74]. Essentially, the modes described by Pfoh et al. [74] differ in two ways: the place where the view-generation takes place (e. g. internally or externally), and the way in which semantic knowledge is incorporated. Pfoh et al. [74] considered three modes in their formal model in order to bridge the semantic gap:

- Out-of-Band delivery: The view-generating function is implemented so that semantic knowledge is received in advance before the actual VMI begins. VMs do not need to run while the view generation process takes place. Among the main disadvantages is that this approach cannot be implemented guest-portable [24]. However, it allows the integration of tools such as the usage of the Volatility Framework [75] to evaluate data structures from memory dumps acquired from virtualized environments.

- In-Band delivery: The view-generating function is internal and therefore it can make use of the guest OS' knowledge of the deployed software architecture. A disadvantage arises from this component being susceptible to compromise from malicous entities which have compromised monitored guest OSes [74]. Further, this method actually does not bridge the semantic gap, it rather avoids it.

- Derivation: In this case information is derived from the VMM through semantic knowledge of the hardware architecture. In their paper, Pfoh et al. [74] mention that "understanding a particular architecture and monitoring control registers within a CPU provides us with some semantic knowledge". Thus, the approach is guest-portable [24].

In the following we will describe different approaches depending on different common hypervisors. All of them follow the "out-of-band delivery" approach.

XEN Due to the requirement of direct memory access of virtualization hardware, the LibVMI framework runs within Dom0. When an application running in Dom0 accesses a specific address within a VM running in DomU, XEN has to translate this address into a physical address. This address in turn gets mapped back into the Dom0 address space. This procedure has to be repeated for every access to the different memory regions. As an example, access to the `task_struct` of the Linux operating system is described. The data structe mentioned before contains pointers to information required to manage tasks of the Linux operating system like the next and previous entry, the process ID offset, executable name offset or the `signal_struct` offset. To get all information about processes, relevant memory addresses have to be mapped forth and back between Dom0 and DomU address space. In order to "walk" through this process list, each entry of the list has to be mapped to an accessible memory region of Dom0 [76]. This procedure creates overhead for virtual machine introspection which is made necessary by the design of this virtualization approach. LibVMI enables investigators to access low-level information. Memory mapping procedures are abstracted away from users of the programming library.

Paravirtualization and hardware-assisted virtualization. Former requires the guest-operating system to be modified. In case of Windows the paravirtualized approach is supported through the "Xen Windows GplPv" drivers [77]. The introduction of technologies such as "Intel VT-x" and "AMD-V" allowed hardware-assisted virtualization which resulted in the ability to run unmodified (closed-source) operating systems such as Microsoft Windows.

In their work, Lengyel et al. [78] utilize information from Xen hypervisors in order to analyze malware samples. The major contribution of their project is an automated malware collection and analysis system by setting up hybrid honeypots that are exposed to unprotected public Internet. Baiardi et al. [79, 80] follow a similar approach ("PsycoTrace"), but malware samples are analyzed through both static and dynamic tools. The sequence of system calls is described by static tools that rely on a context-free grammar. Dynamic tools observe call traces of processes in order to check if they belong to static definitions. Conformity is further checked by the evaluation of assertions.

KVM Kernel-based Virtual Machine (KVM) [81] is a solution for Linux on x86 platforms that supports full virtualization. Hardware virtualization extensions such as Intel VT or AMD-V are supported through loadable kernel-modules that provide the core virtualization infrastructure as well as processor specific functionality. The virtualization infrastructure (such as computer devices like hard-disks, sound-card, etc.) is provided by Qemu [82]. LibVMI [72] offers functionality to inspect KVM-virtualized virtual machines in order to obtain forensically sound memory dumps. At the time of writing this feature is experimental. Patches [72] are available only for specific versions (QEMU-KVM 0.14.0) of Qemu. Another source of information for KVM virtualized environments is the Qemu monitor (reachable via "Ctrl-Alt-2" within a session window). E. g. the "memsave" command allows to dump memory in a read-only fashion of virtual machines. Memory dumps can then be processed further in order to obtain information about the processes running in virtual machines.

The "KvmSec" [83] project focuses on extending the Linux Kernel Virtual Machine in order to increase the protection of guest virtual machines against viruses and kernel rootkits. In contrast to most other projects this project's architecture is composed of multiple modules that live both in the host as well as the guest kernels. Thus, this project implements the in-bound delivery model [74]. Modules inside and outside of VMs communicate with each other through shared memory. Sharif et al. [84] follow a similar approach: in order to provide security monitors, that improve the security of executed processes, a general-purpose framework based on hardware-virtualization is installed into the VM that is protected.

VMware ESXi A prototype ("Livewire") for a VMI-based Intrusion Detection System (IDS) has been proposed by Garfinkel and Rosenblum [69]. They modified VMware Workstation on Linux for the x86 platform to offer hooks in order to gain access to memory, CPU registers and device states.

In 2008 VMware announced the VMsafe program [85]. This closed development is only available for chosen partners of VMware who develop security solutions in order to enhance the security of virtual machines. The software provided by this project provides interfaces to the hypervisor to enable the implementation of antivirus, firewall, and IDS/IPS solutions on an abstraction level close to the hypervisor. The VMsafe API is split into three parts: vCompute, vNetwork, and the Virtual Disk Development Kit (VDDK) API [86].

The vCompute API enables introspection of CPU states and registers and access to the VMs memory. The vNetwork API enables packet inspection between the virtual switch and the vNIC of running VMs. This allows for running firewalls right in front of one or more VMs, thus eliminating the need for a dedicated virtualized firewall, a physical appliance or personal firewall which simplifies the overall configuration effort for the network part. The VDDK API allows to manage virtual storage. It comes bundled with an API as well as a SDK and allows to implement e. g. malware or antivirus solutions [86] without the need for running several instances of antivirus software on each of the involved virtual machines.

The "CloudSec" project [87] focuses on active, transparent, and real-time monitoring of security properties of hosted VMs in IaaS cloud setups by an additional monitoring appliance. Access to physical memory of VMs is accomplished by performing Virtual Machine Introspection interfacing VMware's VMsafe APIs. It is not necessary to install security code inside VMs. Low-level information (bytes) is mapped into high-level data structures (OS data structures) that allow the detection of Dynamic Kernel Object Manipulation (DKOM) and Kernel Object Hooking (KOH) rootkits. The semantic gap is bridged by the so called "Semantic Gap Builder" (SGB). It reads specific physical memory pages according to definitions of OS global variables' addresses based on specified Kernel Structures Definitions (KSDs). Access to memory occurs through a back-end (the VMI component). Thus, this approach corresponds to the out-of-band delivery model. Triggers are installed to invoke functionality in the cloudSec front-end.

Microsoft Hyper-V In a blog entry on Microsoft's TechNet [88] Russinovich introduced "LiveCloudKd" which extends the existing "LiveKd" project [89]. LiveKd is a utility that allows to run Microsoft's kernel debuggers (Kd and Windbg) on locally live systems. "LiveCloudKd" further extends the project by supporting VMs powered by Microsoft Hyper-V. LiveCloudKd allows for pausing and resuming VMs and copying their memory to files. The tool runs within the Hyper-V server. LiveCloudKd is developed by M. Suiche and available for free [90]. "HyperTaskMgr" [91] allows to visualize all running Hyper-V VMs on a system and to extract further information like running processes and attached DLLs. Further it enables the user to elevate the privileges of any process or to kill specified processes from outside of virtual machines.

3.2 Correlation of Evidence Accross Cloud Environments

The resource pooling feature of cloud computing environments is implemented by utilizing virtualization techniques [3]. With virtualization being a key-technology involved in cloud computing, the Virtual Machine Introspection approach has to be taken one step further. Information and digital evidence acquired from different hypervisors in order to obtain the "big picture" of such highly distributed systems as given in cloud infrastructures.

Existing scientific research which was based on VM introspection and monitoring software focused mainly on the detection of and defence from malicious software. Ando et al. [92] modified Linux as guest operating system to be able to obtain event-driven memory snapshots. Heuristics developed in this project allowed the detection of unknown malware which could not be detected by characteristic signatures. Kuhn and Taylor [93] focused on capturing exploits in virtualized environments (such as cloud infrastructures). They concluded that there is no common collective base of root-kits, applications, and kernel versions for the forensic analysis of memory in virtualized environments to form a ground-truth for cross technology comparisons. Lempereur et al. [94] presented a framework which could be used to automatically evaluate live digital forensic acquisition tools on different platform configurations. Live digital forensics techniques play an important role in the area of virtualized environments. In their work they describe three classes of digital forensic evidence: stored information (high amount, slow access), information pending storage, and operational information. Operational information can help to narrow down the amount of searches to analyze stored information. This is true for both locally stored information (e. g. within an instance) and information stored on remote systems (e. g. cloud storage). Krishnan et al. [95] proposed a forensics platform that transparently monitored and recorded data access events within a virtualized environment by only using the abstractions which were exposed by the hypervisor. The developments focused on monitoring access to objects on disk and allowed to follow the causal chain of the accesses across processes even if objects were copied into memory. Transactions of data have then be recorded in an audit log which allowed for faithful reconstruction of recorded events and the changes that they induced. In their work the authors demonstrated how their approach could be used to obtain behavioral profiles of malware. Dykstra and Sherman [96] explain how to extract volatile and non-volatile data from cloud infrastructures such as Amazon EC2 with nowadays tools such as Guidance EnCase and AccessData Forensic Toolkit.

Recent projects combine Hypervisor forensics with techniques known from the field of Live/Memory forensics. Dolan-Gavitt et al. [97, 98] interlink the Volatility Framework [75] with techniques from the field of Hypervisor forensics. To improve the results of the acquisition and analysis process the authors developed a whole-system dynamic slicing technique for executables. The semantic gap is bridged by modeling the behavior of operating systems. From a high-level point of view, the process of modeling an OS' behavior occurs in three phases: the training, the analysis, and the runtime phase. During the training phase, in-guest programs start processes that are started so that as many execution paths of each process are executed as possible. Result of this procedure are instruction traces that can be used to analyze memory structures outside introspected VMs. During the analysis phase "noise" of the execution process is removed by performing dynamic data slice on each trace (footprint). The result of this step is a unified program that can be used for out-of-guest introspection. During the runtime-phase the generated program is then used to introspect running virtual machines. In case of the Virtuoso project, the base for obtaining memory dumps from VMs is "PyXa" [98, 24]. Access to VMs virtualized in Xen is provided through Python objects with a `read` method that reads single pages of physical memory when invoked. The Python wrapper allows for prototyping VMI applications. The Volatility Framework [75] is also implemented in Python, thus allowing for interoperability between developments.

Different acquisition modes from the formal models to bridge the "semantic gap" [74] have to be applied in order to determine relevant evidence from cloud infrastructures. In order to obtain information from within VMs it could be helpful to install additional software inside the VMs. This corresponds to the in-bound delivery model mentioned above. Carbone et al. [99] follow this approach by developing a secure and robust infrastructure called "SYRINGE". The monitoring application is protected because it is put into a separate virtual machine as known from the out-of-guest approach. Nevertheless, it is possible to invoke guest functions by utilizing the function-call injection technique. Another technique, localized shepherding, helps to verify the secure execution of invoked guest OS code. Localized Shepherding refers to a technique that allows to shepherd the thread of guest code executed after being injected for function calls. Shepherding in turn refers to verifying code in memory against pre-compiled white lists in order to prevent code-patching attacks. Instrumentation further helps to dynamically evaluate instructions that could be changed by attackers to divert the regular control flow.

In order to correlate information from different hypervisors, information from additional cloud infrastructure elements has to be considered. Figure 1 shows an example of the current OpenStack ("Folsom" release) architecture [100]. Circles refer to Linux services which are part of OpenStack. Rectangles refer to external components which are not part of the OpenStack project. Interactions between OpenStack components and external components are shown as solid line. Dashed lines represent interactions between external components [101]. Central elements of this architecture are the "Message queue" (RabbitMQ, Qpid, or ZeroMQ) as well as the "Database" (MySQL, PostgreSQL, or sqlite). As all tasks are coordinated through these components, it turns out that crucial meta-data about the overall information flow can be found there.

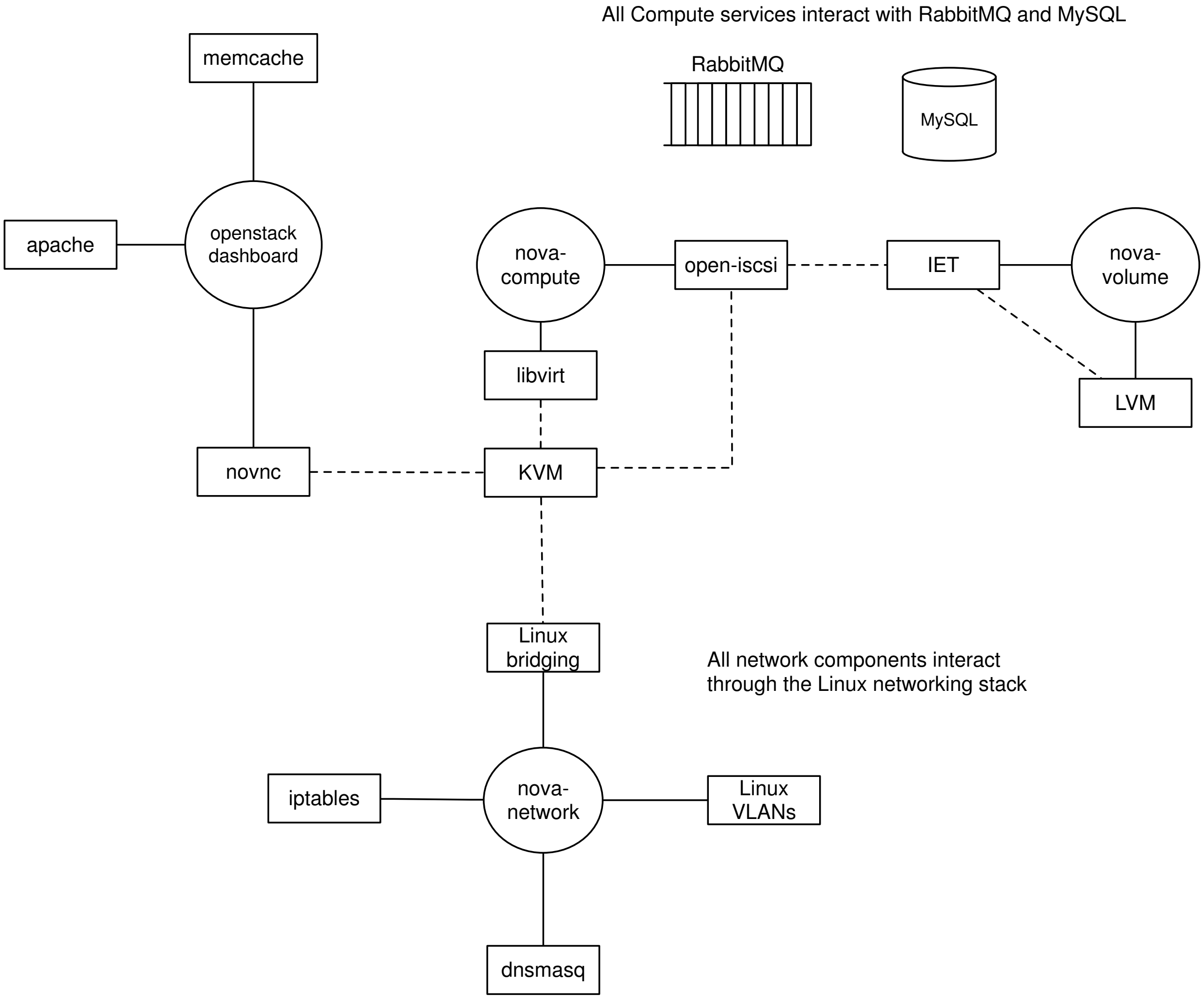


Figure 1: Architecture of the OpenStack Folsom Project [101]

4 Conclusion and Outlook

Within this paper the current State-of-the-Art in cloud forensics has been presented. We focused on existing digital forensics frameworks to show the lack of regulations when investigating cloud environments. Further we described the investigation of cloud infrastructures from technical, legal, and organizational points of view. Subsequently we elucidated how to perform digital investigations using cloud infrastructures. Due to the massive computing power available in cloud environments there are opportunities that can improve the forensics acquisition and analysis process. Main contribution of this part is an extensive discussion on acquiring digital evidence from hypervisors with a focus on cloud computing.

Current research results demonstrate the feasibility of information acquisition from virtual machine managers (Hypervisors) to support the digital forensics analysis process. However, most work is focused on smaller setups (e. g. single physical machine with several VMs). Therefore we propose that more research should be done to investigate the acquisition of digital evidence across multiple virtualized environments, as given in cloud computing. In regard to the OpenStack project we consider to perform more research in the area of this cloud's infrastructural components in order to improve the process of acquiring digital evidence from cloud computing environments. More specifically we consider to investigate the data structures used in both the queuing and the database backend of OpenStack. Further we intend to correlate data acquired from different hypervisors supported by information acquired from the backends mentioned before.

References

[1] R. Poisel and S. Tjoa, "Discussion on the challenges and opportunities of cloud forensics," in *Proc. of the 2nd Multidisciplinary Research and Practice for Information Systems (CD-ARES'12), Prague, Czech, LNCS*, vol. 7465. Springer-Verlag, 2012, pp. 593–608.

[2] I. T. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Proc. of the 2008 Grid Computing Environments Workshop (GCE'08), Austin, Texas, USA*. IEEE, November 2008, pp. 1–10.

[3] P. Mell and T. Grance, "The nist definition of cloud computing," http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, September 2011.

[4] European Commission, "Digital Agenda: New strategy to drive European business and government productivity via cloud computing," http://europa.eu/rapid/pressReleasesAction.do?reference=IP/12/1025, September 2012, [Online; Status: October 10^{th} 2012].

[5] R. Tolido, "Cloud uncertainty is the enemy of investment," http://www.ft.com/cms/s/0/fd41369a-fde6-11e1-9901-00144feabdc0.html, September 2012, [Online; Status: March 18^{th} 2013].

[6] B. Kazarian and B. Hanlon, "SMB Cloud Adoption Study Dec 2010 - Global Report," http://www.microsoft.com/Presspass/presskits/commsector/\docs/SMBStudy_032011.pdf, December 2010, accessed: December 30^{th} 2011.

[7] Gartner, "Gartner says worldwide cloud services market to surpass $68 billion in 2010," http://www.gartner.com/it/page.jsp?id=1389313, 2010, accessed: December 30^{th} 2011.

[8] P. Sinha, "India to connect sdcs via national cloud network," http://www.igovernment.in/site/india-connect-sdcs-national-cloud-network, June 2012, accessed October 15th, 2012.

[9] S. Paquette, P. T. Jaeger, and S. C. Wilson, "Identifying the security risks associated with governmental use of cloud computing," *Government Information Quarterly*, vol. 27, no. 3, pp. 245–253, July 2010.

[10] G. H. Carlton and H. Zhou, "A Survey of Cloud Computing Challenges from a Digital Forensics Perspective," *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 3, no. 4, pp. 1–16, March 2011.

[11] S. Mason and E. George, "Digital evidence and "cloud" computing," *Computer Law & Security Review*, vol. 27, no. 5, pp. 524–528, September 2011.

[12] T. S. Dillon, C. Wu, and E. Chang, “Cloud Computing: Issues and Challenges,” in *Proc. of The International Conference on Advanced Information Networking and Applications (AINA’10), Perth, Australia*. IEEE, April 2010, pp. 27–33.

[13] R. Krutz and R. Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. John Wiley & Sons, 2010.

[14] I-CIO, “Crime as a service,” http://www.i-cio.com/features/june-2010/crime-as-a-service, 06 2010, accessed October 15th, 2012.

[15] Fortinet, “2012 threat predictions,” http://blog.fortinet.com/2012-threat-predictions/, 2012, accessed October 15th, 2012.

[16] D. Birk, “Technical Challenges of Forensic Investigations in Cloud Computing Environments,” in *Workshop on Cryptography and Security in Clouds, Zurich, Switzerland*, March 2011.

[17] M. Damshenas, A. Dehghantanha, R. Mahmoud, and S. bin Shamsuddin, “Forensics investigation challenges in cloud computing environments,” in *Proc. of International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec’12), Kuala Lumpur, Malaysia*. IEEE, June 2012, pp. 190–194.

[18] N. Beebe and N. Beebe, “Digital Forensic Research: The Good, the Bad and the Unaddressed,” in *Proc. of the 5th IFIP WG 11.9 International Conference on Digital Forensics (ICDF’08), Orlando, Florida, USA, IFIP Advances in Information and Communication Technology*, vol. 306. Springer-Verlag, January 2009, pp. 17–36.

[19] S. L. Garfinkel, “Digital forensics research: The next 10 years,” in *Proc. of the 12th Annual Digital Forensics Research Conference (DFRWS’10), Portland, Oregon, USA, Digital Investigation*, vol. 7, no. Supplement, August 2010, pp. 64–73.

[20] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, “Cloud forensics,” *Advances in Digital Forensics 7*, vol. 361, pp. 35–49, 2011.

[21] T. Lillard, C. Garrison, C. Schiller, J. Steele, and J. Murray, *Digital Forensics for Network, Internet, and Cloud Computing: A Forensic Evidence Guide for Moving Targets and Data*. Elsevier, 2010.

[22] W. Delport, M. S. Olivier, and M. Koehn, “Isolating a Cloud Instance for a Digital Forensic Investigation,” in *Proc. of the 2011 Information Security for South Africa (ISSA’11), Johannesburg, South Africa*. IEEE, August 2011, pp. 25–33.

[23] BBC News, “Children warned against net predators,” http://news.bbc.co.uk/2/hi/uk_news/education/648156.stm, 2000.

[24] J. C. F. Cruz and T. Atkison, “Evolution of traditional digital forensics in virtualization,” in *Proc. of the 50th Annual Southeast Regional Conference, Tuscaloosa (ACM-SE’12), Alabama, USA*. ACM, March 2012, pp. 18–23.

[25] National Institute of Standards and Technology, “Electronic Crime Scene Investigation: An On-the-Scene Reference for First Responders,” http://www.ncjrs.gov/pdffiles1/nij/227050.pdf, 2001, Recommendations of the National Institute of Standards and Technology.

[26] K. Karen, S. Chevalier, T. Grance, and H. Dang, “NIST-SP800-86: Guide to Integrating Forensic Techniques into Incident Response,” http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf, August 2006, recommendations of the National Institute of Standards and Technology.

[27] ACPO, “Good Practice Guide for Computer-Based Electronic Evidence,” http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence.pdf, August 2007, [Online; Status: March 18^{th} 2013].

[28] B. D. Carrier and E. H. Spafford, “Getting Physical with the Digital Investigation Process,” *International Journal of Digital Evidence*, vol. 2, no. 2, pp. 1–20, September 2003.

[29] F. Cohen, *Digital Forensic Evidence Examination - 2nd Ed.* Fred Cohen & Associates, 2010.

[30] L. Ke, “Design of a forensic overlay model for application development,” Master’s thesis, University of Canterbury, College of Engineering, 2011.

[31] M. M. Pollitt, “An ad hoc review of digital forensic models,” in *Proc. of the 2nd International Workshop Systematic Approaches to Digital Forensic Engineering (SADFE’07), Seattle, Washington, USA*. IEEE, April 2007, pp. 43–54.

[32] B. Martini and K.-K. R. Choo, “An integrated conceptual digital forensic framework for cloud computing,”

Digital Investigation, vol. 9, no. 2, pp. 71–80, November 2012.

[33] R. McKemmish, “What is forensic computing?” *Trends and issues in crime and criminal justice*, vol. 118, pp. 1–6, June 1999.

[34] S. Zimmerman and D. Glavach, “Cyber Forensics In the Cloud,” *IAnewsletter*, vol. 14, no. 1, pp. 4–7, 2011.

[35] S. Biggs and S. Vidalis, “Cloud Computing: The impact on digital forensic investigations,” in *Proc. of the International Conference for Internet Technology and Secured Transactions (ICITS’09), London, UK*. IEEE, November 2009, pp. 1–6.

[36] M. Pollitt, “Blue skies and storm clouds,” *Journal of Digital Forensic Practice*, vol. 2, no. 2, pp. 105–106, 2008.

[37] S. D. Wolthusen, “Overcast: Forensic Discovery in Cloud Environments,” in *Proc. of the 5th International Conference on IT Security Incident Management and IT Forensics (IMF’09), Stuttgart, Germany*. IEEE, September 2009, pp. 3–9.

[38] G. Grispos, T. Storer, and W. B. Glisson, “Calm before the storm: The challenges of cloud computing in digital forensics,” *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 4, no. 2, pp. 28–48, March 2012.

[39] E. Casey, *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*. Academic Press, 2011.

[40] D. Reilly, C. Wren, and T. Berry, “Cloud computing: Forensic challenges for law enforcement,” in *Proc. of the 5th International Conferenc for Internet Technology and Secured Transactions (ICITST’10), London, UK*. IEEE, November 2010, pp. 1–7.

[41] D. Birk and C. Wegener, “Technical Issues of Forensic Investigations in Cloud Computing Environments,” in *Proc. of the 6th International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE’11), Oakland, California, USA*. IEEE, May 2011, pp. 1–10.

[42] K. Wang, “Using a Local Search Warrant to Acquire Evidence Stored Overseas via the Internet,” in *Advances in Digital Forensics VI, IFIP Advances in Information and Communication Technology*, K.-P. Chow and S. Shenoi, Eds. Springer, 2010, vol. 337, pp. 37–48.

[43] D. Barrett and G. Kipper, *Virtualization and Forensics: A Digital Forensic Investigator’s Guide to Virtual Environments*, ser. Syngress Media. Syngress/Elsevier, 2010.

[44] D. Brezinski and T. Killalea, “Guidelines for Evidence Collection and Archiving,” IETF RFC 3227, February 2002, http://www.ietf.org/rfc/rfc3227.txt.

[45] M. Taylor, J. Haggerty, D. Gresty, and D. Lamb, “Forensic investigation of cloud computing systems,” *Network Security*, vol. 2011, no. 3, pp. 4–10, 2011.

[46] C. Yan, “Cybercrime forensic system in cloud computing,” in *Proc. of the 2011 International Conference on Image Analysis and Signal Processing (IASP’11), Wuhan, Hubei, China*. IEEE, October 2011, pp. 612–615.

[47] I. Agudo, D. Nuñez, G. Giammatteo, P. Rizomiliotis, and C. Lambrinoudakis, “Cryptography goes to the cloud,” in *Proc. of the 1st International Workshop on Security and Trust for Applications in Virtualised Environments (STAVE’11), Seoul, Korea, Communications in Computer and Information Science*, vol. 187. Springer-Verlag, June 2011, pp. 190–197.

[48] R. Marty, “Cloud application logging for forensics,” in *Proc. of the 2011 ACM Symposium on Applied Computing (SAC’11), Taichung, Taiwan*. ACM, March 2011, pp. 178–184.

[49] R. Hegarty, M. Merabti, Q. Shi, and B. Askwith, “Forensic analysis of distributed service oriented computing platforms,” in *Proc. of the 12th Annual Post Graduate Network Symposium (PGNet’11), Liverpool, UK*, June 2011.

[50] Y. Wang, J. Cannady, and J. Rosenbluth, “Foundations of computer forensics: A technology for the fight against computer crime,” *Computer Law & Security Review*, vol. 21, no. 2, pp. 119 – 127, May 2005.

[51] C. V. Marsico, “Computer evidence v. daubert: The coming conflict,” Center for Education and Research in Information Assurance and Security, Purdue University, CERIAS Tech Report 2005-17, 2005.

[52] Cloud Security Alliance, “Security guidance for critical areas of focus in cloud computing v3.0,” https://cloudsecurityalliance.org/research/security-guidance/, 2011.

[53] V. Roussev and G. G. Richard, "Breaking the performance wall: The case for distributed digital forensics," in *Proc. of the 2004 Digital Forensics Research Workshop (DFRWS'04), Baltimore, Maryland, USA*, August 2004.

[54] G. G. Richard and V. Roussev, "Next-generation digital forensics," *Communications of the ACM*, vol. 49, no. 2, pp. 76–80, February 2006.

[55] V. Roussev, L. Wang, G. G. Richard, and L. Marziale, "MMR: A platform for large-scale forensic computing," in *Proc. of the 5th Annual IFIP WG 11.9 International Conference on Digital Forensics, Florida, USA*, January 2009.

[56] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Magazine Communications of the ACM - 50th anniversary issue: 1958*, vol. 51, no. 1, pp. 107–113, January 2008.

[57] J. Talbot and R. Yoo, "The Phoenix System for MapReduce Programming," http://mapreduce.stanford.edu/, [Online; Status: December 30th 2012].

[58] "Hadoop - mapreduce," http://hadoop.apache.org/mapreduce, 2013, [Online; Status: March 18th 2013].

[59] M. Cohen, D. Bilby, and G. Caronni, "Distributed forensics and incident response in the enterprise," *Digital Investigation*, vol. 8, Supplement, no. 0, pp. 101–110, August 2011.

[60] A. Zonenberg, "Distributed Hash Cracker: A Cross-Platform GPU-Accelerated Password Recovery System," http://www.cs.rpi.edu/~zonena/papers/cracker.pdf, 2009.

[61] AccessData, "Decryption and password cracking software," http://accessdata.com/products/computer-forensics/decryption, [Online; Status: March 18th 2013].

[62] G. Starcher, "Accessdata dna & amazon ec2," https://www.georgestarcher.com/?tag=amazon-ec2, 2011.

[63] "Freeeed.org - open-source ediscovery engine," http://www.freeeed.org/, 2013, [Online; Status: March 18th 2013].

[64] M. Taylor, J. Haggerty, D. Gresty, and R. Hegarty, "Digital evidence in cloud computing systems," *Computer Law & Security Review*, vol. 26, no. 3, pp. 304–308, 2010.

[65] A. O. Flaglien, A. Mallasvik, M. Mustorp, and A. Arnes, "Storage and exchange formats for digital evidence," *Digital Investigation*, vol. 8, no. 2, pp. 122–128, November 2011.

[66] R. Lu, X. Lin, X. Liang, and X. S. Shen, "Secure provenance: the essential of bread and butter of data forensics in cloud computing," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10), Beijing, China*. ACM, 2010, pp. 282–292.

[67] R. Chow, P. Golle, M. Jakobsson, R. Masuoka, and J. Molina, "Controlling Data in the Cloud:Outsourcing Computation without Outsourcing Control," in *Proc. of the 2009 ACM Workshop on Cloud Computing Security (CCSW'09), Chicago, Illinois, USA*. ACM, November 2009, pp. 85–90.

[68] R. Accorsi, "Business Process as a Service: Chances for Remote Auditing," in *Proc. of the 35th Annual Computer Software and Applications Conference Workshops (COMPSACW'11), Munich, Germany*. IEEE, July 2011, pp. 398–403.

[69] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proc. of the 10th Annual Network and Distributed System Security Symposium (NDSS'03), California, USA*. ISOC, February 2003, pp. 191–206.

[70] K. Kourai and S. Chiba, "Hyperspector: virtual distributed monitoring environments for secure intrusion detection," in *Proc. of the 1st ACM/USENIX international conference on Virtual execution environments (VEE'05), Chicago, Illinois, USA*. ACM, June 2005, pp. 197–207.

[71] B. D. Payne and W. Lee, "Secure and flexible monitoring of virtual machines," in *Proc. of the 23rd Annual Computer Security Applications Conference (ACSAC'07), Florida, USA*, December 2007, pp. 385–397.

[72] Sandia National Laboratories, "LibVMI," http://code.google.com/p/vmitools/, 2011, [Online; Status: September 1st 2012].

[73] P. Chen and B. Noble, "When virtual is better than real (operating system relocation to virtual machines)," in *Proc. of the 8th Workshop on Hot Topics in Operating Systems (HotOS'01), Elmau, Germany*. IEEE, May 2001, pp. 133–138.

[74] J. Pfoh, C. Schneider, and C. Eckert, "A formal model for virtual machine introspection," in *Proc. of the 1st ACM Workshop on Virtual Machine Security (VMSec'09), Illinois, USA*. ACM, November 2009, pp. 1–10.

[75] A. Walters, “The Volatility Framework: Volatile memory artifact extraction utility framework,” https://www.volatilesystems.com/default/volatility, [Online; Status: October 10^{th} 2012].

[76] B. Hay and K. Nance, “Forensics examination of volatile system data using virtual introspection,” *ACM SIGOPS Operating Systems Review*, vol. 42, no. 3, pp. 74–82, April 2008.

[77] L. Kurth, “Xen windows gplpv,” http://wiki.xen.org/wiki/Xen_Windows_GplPv, 2012, [Online; Accessed: October 10^{th} 2012].

[78] T. Lengyel, J. Neumann, S. Maresca, B. Payne, and A. Kiayias, “Virtual machine introspection in a hybrid honeypot architecture,” in *the 5th USENIX conference on Cyber Security Experimentation and Test (CSET’12), Washington, USA*, August 2012.

[79] F. Baiardi, D. Maggiari, D. Sgandurra, and F. Tamberi, “PsycoTrace: Virtual and Transparent Monitoring of a Process Self,” in *Proc. of the 17th Euromicro International Conference on Parallel Distributed and Network-based Processing, Weimar, Germany*. IEEE, February 2009, pp. 393–397.

[80] ——, “Transparent Process Monitoring in a Virtual Environment,” in *Proc. of the 3rd International Workshop on Views On Designing Complex Architectures (VODCA’08), Bertinoro, Italy, ENTCS*, vol. 236. Elsevier, November 2009, pp. 85–100.

[81] “Kernel Based Virtual Machine,” http://www.linux-kvm.org/, [Online; Status: October 15^{th} 2012].

[82] “Qemu - Open Source Processor Emulator,” http://www.qemu.org, [Online; Status: October 15^{th} 2012].

[83] F. Lombardi and R. D. Pietro, “KvmSec: a security extension for Linux kernel virtual machines,” in *Proc. of the ACM symposium on Applied Computing (SAC’09), Hawaii, USA*. ACM, March 2009, pp. 2029–2034.

[84] M. I. Sharif, W. Lee, W. Cui, and A. Lanzi, “Secure in-vm monitoring using hardware virtualization,” in *Proc. of the 16th ACM conference on Computer and Communications Security (ACM CCS’09), Chicago, Illinois, USA*. ACM, October 2009, pp. 477–487.

[85] “New VMware VMsafe^{TM} Technology Allows the Virtual Datacenter to Be More Secure Than Physical Environments,” http://www.vmware.com/company/news/releases/vmsafe_vmworld.html, VMware, [Online; Status: October 15^{th} 2012].

[86] “What actually is VMsafe and the VMsafe API?” http://blogs.vmware.com/vcloud/2010/04/what-actually-is-vmsafe-and-the-vmsafe-api.html, VMware, [Online; Status: October 15^{th} 2012].

[87] A. Ibrahim, J. Hamlyn-Harris, J. Grundy, and M. Almorsy, “CloudSec: A security monitoring appliance for Virtual Machines in the IaaS cloud model,” in *Proc. of the 5th International Conference on Network and System Security (NSS’11), Milan, Italy*. IEEE, September 2011, pp. 113 –120.

[88] M. Russinovich, “LiveKd for Virtual Machine Debugging,” http://blogs.technet.com/b/markrussinovich/archive/2010/10/14/3360991.aspx, October 2010, [Online; Status: October 15^{th} 2012].

[89] ——, “LiveKd,” http://technet.microsoft.com/en-us/sysinternals/bb897415.aspx, October 2012, [Online; Status: October 15^{th} 2012].

[90] M. Suiche, “LiveCloudKd - Debugging the clouds from the Moon,” http://www.moonsols.com/2010/08/12/livecloudkd/, August 2010, [Online; Status: October 15^{th} 2012].

[91] ——, “MoonSols HyperTaskMgr v1.0,” http://www.moonsols.com/2011/07/19/new-utility-moonsols-hypertaskmgr-v1-0/, July 2011, [Online; Status: October 15^{th} 2012].

[92] R. Ando, Y. Kadobayashi, and Y. Shinoda, “Asynchronous Pseudo Physical Memory Snapshot and Forensics on Paravirtualized VMM Using Split Kernel Module,” in *Proc. of the 10th International Conference on Information Security and Cryptology (ICISC’07), Seoul, Korea, LNCS*, vol. 4817. Springer-Verlag, November 2007, pp. 131–143.

[93] S. Kuhn and S. Taylor, “A survey of forensic analysis in virtualized environments,” Dartmouth College, Hanover, New Hampshire, Tech. Rep., 2011.

[94] B. Lempereur, M. Merabti, and Q. Shi, “Pypette : A framework for the automated evaluation of live digital forensic techniques,” in *Proc. of the 11th Annual PostGraduate Symposium on The Convergence of Telecommunications Networking and Broadcasting, Liverpool, UK*, June 2010.

[95] S. Krishnan, K. Z. Snow, and F. Monrose, “Trail of bytes: efficient support for forensic analysis,” in *Proc. of the 2010 ACM Conference on Computer and Communications Security (ACM CCS’10), Illinois, USA*. ACM, October 2010, pp. 50–60.

[96] J. Dykstra and A. T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques," in *Proc. of the 12th Annual Digital Forensics Research Conference (DFRWS'12), Washington, DC, USA, Digital Investigation*, vol. 9, August 2012, pp. 90–98.

[97] B. Dolan-Gavitt, T. Leek, M. Zhivich, J. Giffin, and W. Lee, "Virtuoso: Narrowing the Semantic Gap in Virtual Machine Introspection," in *Proc. of the 2011 IEEE Symposium on Security and Privacy (SP'11), Oakland, California, USA*. IEEE, May 2011, pp. 297–312.

[98] B. Dolan-Gavitt, B. Payne, and W. Lee, "Leveraging forensic tools for virtual machine introspection," Georgia Institute of Technology. College of Computing and Georgia Institute of Technology. School of Computer Science, Tech. Rep. SCS Technical Report: GT-CS-11-05, 2011.

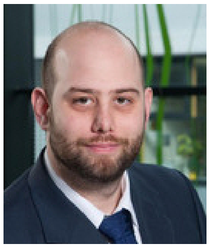
[99] M. Carbone, M. Conover, B. Montague, and W. Lee, "Secure and Robust Monitoring of Virtual Machines through Guest-Assisted Introspection," in *Proc. of the 15th international conference on Research in Attacks, Intrusions, and Defenses (RAID'12), Amsterdam, The Netherlands, LNCS*, vol. 7462. Springer-Verlag, September 2012, pp. 22–41.

[100] K. Pepple, "Openstack folsom architecture," http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/, September 2012, [Online; Accessed: October 10th 2012].

[101] L. Hochstein, "OpenStack - Underlying Technologies," http://docs.openstack.org/trunk/openstack-compute/install/apt/content/externals.html, 2012, [Online; Status: October 15th 2012].

Rainer Poisel received a master's degree in computer science management from the Technical University of Vienna as well as a master's degree in telecommunications from St. Poelten University of Applied Sciences. He has been working as a scientific researcher in the field of information security since 2007. During his last research project he developed a file carver for the recovery of multimedia files from various storage media. Besides the publication of various papers for different security conferences such as D-A-CH Security, IEEE SIBIRCON, DeepSec and Chaos Computer Club he holds certifications from the Linux Professional Institute (LPIC) as well as for AccessData (ACE, PSA) and Cisco (CCNA) products.

Erich Malzer received a master's degree in information security from St. Poelten University of Applied Sciences. His recent research projects focused on the fields of analogue voice encryption and forensics in virtualized environments. Within his diploma thesis about Hypervisor forensics he developed an approach for automated memory analysis and investigations of virtualized machines. He works as system engineer in the field of virtualization, datacenter and application delivery besides holding certifications of Cisco, F5 and VMware.



Simon Tjoa is associate professor at St. Poelten University of Applied Sciences. He received his Ph.D. in informatics from University of Vienna. His research interests include critical infrastructure protection, digital forensics, business continuity management and business process security. He is programm committee and organizing committee member of several security related international workshops and conferences. Furthermore, he currently serves as secretary of IEEE SMC Austria Chapter and holds professional security certifications such as AMBCI, ACE, CISA, CISM.