Forensic Recovery and Intrusion Monitoring in the Cloud

George R. S. Weir
Department of Computer and Information Sciences
University of Strathclyde
Glasgow, UK
e-mail: george.weir@strath.ac.uk

Andreas Aßmuth and Nicholas Jäger
University of Applied Sciences
OTH Amberg-Weiden
Germany
e-mail: {a.assmuth,n.jaeger}@oth-aw.de

***Abstract*—As organisations move away from locally hosted computer services toward Cloud platforms, there is a corresponding need to ensure the digital forensic integrity of such instances. This need is largely motivated by the locus of responsibility and also by the associated risk of legal sanction and financial penalty. Effective monitoring of activity and events is an essential aspect of such forensic readiness. A major concern is the risk that monitoring systems may themselves be targeted and affected by intruders, thereby nullifying the prospective benefits of such internal software surveillance facilities. In this paper, we outline an approach to intrusion monitoring that aims to ensure the credibility of log data and provide a means of data sharing that supports log reconstruction in the event that one or more logging systems is maliciously impaired. In addition, we identify and describe the multi-level interpretation problem as an inherent challenge to managing forensic recovery in the Cloud.**

Keywords-Cloud security; forensic readiness; intrusion monitoring; multi-level interpretation; secure data retention.

I. INTRODUCTION

In the virtual world of interactive software systems, as in the physical world, we often aim to observe and detect behaviour and events that may represent risks or threaten damage to the environment or those within that environment. The primary purpose of such surveillance is to determine the cause and likely consequences of such crucial events. In the event of a security incident, we want to record data that may later have evidential value, shed light on the nature of the occurrence, its context (including significant precursors) and its consequences. Capturing such data in a covert manner aims to reduce the likelihood that the recording facility will be detected and thereby, minimise the prospect that the data collection will be deliberately impaired and the telling data subverted.

While surveillance affords no immediate defence against security breaches, it does illustrate the desirability of establishing auditable data in order that light may later be shed on unauthorised or anomalous events that initially have gone undetected by relevant human agency. With varying degrees of transparency, the logging features in computer operating systems, individual computer applications, network operations and Cloud environments go some way toward addressing this requirement by recording data that may subsequently be consulted, in a process of digital forensics, as evidence of past events.

Although considerable efforts are directed in computer security toward protection and prevention of illicit access and system misuse, digital forensic readiness is increasingly recognised as a necessary measure toward recovery, understanding vulnerabilities and pursuit of those responsible for cyber-misdeeds. In this context, the present paper details the complex problem of managing Cloud forensic recovery [1] and affords a proposed response through application of techniques to bolster digital forensic readiness in the Cloud [2].

In the following, Section II reviews the characteristics of Cloud services and the facilities available to the customer. Section III characterises the attack context, with reference to likely intruder behaviour. In Section IV, we consider the context of Cloud security, with associated network security issues and Cloud security risks addressed in Sections V and VI, respectively.

In Section VII we elaborate upon the role of monitoring as a basis for forensic readiness in Cloud Services, with specific attention to the variety of strategies that may be employed. The effectiveness of such mechanisms for event reconstruction and on-going resilience, is a key consideration. Section VIII presents our proposed monitoring approach that we believe contributes toward a solution to the forensic readiness problem in the Cloud setting.

This is followed by Section IX on Cloud forensic readiness, in which we introduce the multi-level interpretation problem. The paper ends with conclusions in Section X.

II. CLOUD SERVICES

In this section, we briefly review the characteristics of Cloud Services and highlight the security concerns associated with different use contexts.

For many users and organisations, their primary engagement with Cloud computing is remote data storage. To this end, most major online Cloud service providers offer such facilities. Offerings in this area include iCloud for Apple users, as a supplement to local storage capacity and emergency backup for system configuration. Similar service offerings include Google Drive, Microsoft OneDrive and Amazon Drive.

For instance, Dropbox offers a familiar model whereby users may register for a free account with limited storage capacity and a pay option for extended storage capacity and further features. The appeal and benefits from such services are clear from the proliferation of such offerings, as

underlined by the fact that many home broadband contracts include a measure of Cloud storage as standard. Home broadband users often rely on remote storage and backup facilities and may be unaware that Cloud services are the basis for such operation.

Although consumers have been quick to adopt Cloud-based services, there is some concern with security issues that may arise in the Cloud setting [3]-[6], with particular concern for the availability and privacy of data [7].

As a basis for understanding Cloud Services, a taxonomy has been developed by the US National Institute of Standards and Technology (NIST) [8]. Three typical service models are described:

- Software as a Service (SaaS);
- Platform as a Service (PaaS); and
- Infrastructure as a Service (IaaS).

In Software as a Service, the customer is given access to applications running on the service provider's Cloud infrastructure, usually through a variety of client devices and software interfaces. In this arrangement, the customer has no control over the underlying Cloud infrastructure (op. cit., p.2) and this level of service extends from simple file storage, through hosted Web sites and database management to specific Web services, including RESTful applications [9] and use of 'containers' [10].

In Platform as a Service, the customer can deploy their own applications on to the provider's Cloud infrastructure and customer control extends to configuration and management of these Cloud-hosted applications. As before, the customer has no facility to control any other aspects of the underlying Cloud infrastructure [8, p.2].

In Infrastructure as a Service, the customer has more scope for software deployment to the Cloud infrastructure, extending to 'arbitrary software, which can include operating systems and applications' (op. cit.). In this arrangement, the customer's control is still limited to the deployed software applications, including operating systems (e.g., virtual machines) and associated networking features (such as software firewalls) [8, p3].

These three service models characterise typical Cloud Service Provider (CSP) offerings with the increasing levels of access and software capability that are reflected in increasing cost levels. In each of these contexts, management and control of the Cloud infrastructure resides with the CSP, who must be relied upon to manage most security aspects that may impinge upon services purchased by the customer.

Cloud services afford an extensive range of applications and software facilities and many mission-critical services are moving to Cloud as a means of limiting security concerns and assuring greater resilience. Since Cloud services are virtual, system recovery or replacement can be quick, reliable and low-cost [cf. 11]. Cloud-based outsourcing of software applications is recognised as commercially attractive for factors, such as:

- Cost (reduction in local expertise and local infrastructure);
- Reliability (service-level agreements can assure availability);
- Resilience (speedy recovery in the event of data or service loss);
- Technical extensibility (support for multiple instances of applications with increasing availability of service to meet growing demand).

We may broadly differentiate two end-user contexts of Cloud usage. In the first case, the customer employs the Cloud service as a data storage facility. (This is a specific instance of the Software as a Service.) Here, security for the customer is limited to concerns of authorised access, continuity of service and data maintenance. In the second case, the end-user employs the Cloud service as a means of computation. This broadly covers all other Cloud interaction. Here, security for the customer extends to all traditional aspects, including data protection, access authentication, service misappropriation and service availability. While some of these issues may lie within the control of the consumer, the CSP has ultimate management of the infrastructure that affords all of the higher-level service provision.

The security risks associated with these service levels in Cloud provision are elaborated further in Section VII, below.

The extent to which the CSP can reliably manage the security and associated integrity of provided services, depends ultimately upon the availability of techniques for detecting and recording the details of any illicit operations that take place within the Cloud service context. Without recourse to such facilities, the CSP cannot be counted upon to maintain consumer services in a satisfactory fashion since there is lack of assurance that such services have not been infiltrated, impaired or subverted. In addition, ability for the CSP to restore services to pre-compromise level depends largely upon the CSP's facility to identify any delta between pre- and post-intrusion services. Inevitably, this leads back to the issue of digital forensic readiness as applied to the Cloud context.

III. THE ATTACK CONTEXT

Successful cyber-attacks can be construed as having three phases. The first is reconnaissance and information gathering, followed by infiltration and escalation and, finally, exfiltration, assault and obfuscation.

In the first phase, the adversary gathers any information needed to gain access to the system, e.g., open ports, versions of operating systems and software services, security measures (such as firewalls, IDS, etc.) [12]. Using this information, the adversary gains access to the system in the second phase [13].

The process of gaining access might consist of several steps, for example, if the adversary has to comprise another system first, in order to get into the actual target. In this process, the adversary also tries to escalate available privileges in order to gain super-user access to the system.

In the third phase, the adversary extracts any information from the system that might prove to be useful [13]. If the goal of the attack is stealing confidential data, such as user accounts, passwords or credit card information, this data is extracted by the adversary and possibly sold to third parties.

If the cyber-attack has another goal, e.g., sabotage, the adversary extracts the data needed to launch the actual assault, often triggered by a certain date or specific event. In any case, the adversary can be expected to perform whatever action is required to cover their tracks. Among other actions, they may install a rootkit that exchanges current files and services within the system with modified versions of these particular files and services. Such system modifications may extend to altering process information, e.g., a program to list all running processes on the system may be modified to list all running processes except for the processes run by the adversary. Additionally, the adversary may target existing log files that might contain traces of the intrusion.

Such strategies are reflected in many network-based intrusions since, in many instances, network vulnerability is predicated upon known weaknesses in networked hosts.

IV. NETWORK SECURITY RISKS

In non-Cloud systems, the principal ingredients in management responses to security take three general forms:

- System hardening
- Software defences
- Data backup

Firstly, system hardening is an attempt to render known threats ineffective. This includes 'conventional' measures that reduce vulnerability, such as authentication, identity management and access control [14], as well as acting to disable unnecessary services, applying regular software updates (patches) and gauging of the relevance and associated risks from newly published exploits [15]. Modern work-s have also been adapted to meet known cyber threats. Counter measures, like address space randomisation, mandatory access control or maybe sandboxing, are state of the art. In addition, advanced users might even build their own operating system and use selected kernel parameters to further harden their system. The second variety of response to address security issues is the application of software defences. This ranges from antivirus provision to firewalls and may also include some variety of intrusion detection, usually rule-based [16] or anomaly-based [17].

Any computing system may be described by a simple layer-based model. Obviously, security on any higher layer strongly depends on access control mechanisms of lower layers. Even if users or service providers only aim for access control on a higher level to secure their application, these access control mechanisms in practice are more complex than those on lower layers. In addition, vulnerabilities or inadequate configuration on lower levels may lead to bypassing security measures on higher layers. Therefore, appropriate countermeasures are necessary on all layers.

A third security measure is the provision of regular data backup, as a means of ensuring that any system failure or intrusion does not result in irretrievable data loss.

V. CLOUD SECURITY RISKS

Perhaps unsurprisingly, Cloud configurations are subject to levels of security risk that go beyond those affecting conventional networked computer systems. In consequence, the security measures outlined above may not be sufficient in the Cloud setting. In elaborating this claim, the Cloud issues are best illustrated with reference to the differing Cloud service offerings mentioned above [8].

These models for Cloud service provision are helpfully elucidated by Gibson et al. [18], as follows:

- "IaaS provides users with a web-based service that can be used to create, destroy and manage virtual machines and storage. It can be used to meter the use of resources over a period of time, which in turn, can be billed back to users at a negotiated rate. It alleviates the users of the responsibility of managing the physical and virtualized infrastructure, while still retaining control over the operating system, configuration and software running on the virtual machines" [op. cit., p. 199].
- "Platform-as-a-Service providers offer access to APIs, programming languages and development middleware which allows subscribers to develop custom applications without installing or configuring the development environment" [op. cit., p. 200].
- "Software-as-a-Service gives subscribed or pay-per-use users access to software or services that reside in the Cloud and not on the user's device" [op. cit., p. 202].

Our earlier noted approaches to system security are equally applicable to Cloud-based systems. With an eye specifically on Cloud security, we can consider how each of these service offerings may be at risk and what precautions may be anticipated in response to these risks.

1. Infrastructure as a Service

This kind of service seems most prone to the types of exploit that one would expect with conventional networked computers, principally, because, in most cases, such virtual machines will be presented to the Internet as networked hosts. Here, the customer is deploying a virtual machine with associated operating system and on-board software applications. This raises the prospect of vulnerabilities at network level, as well as application level issues, e.g., with Web systems and Database servers, Cross-Site Scripting (XSS) or SQL injections. Denial of service attacks are also a legitimate concern, especially since this kind of attack can achieve enormous bandwidths by using IoT devices for their purpose [19]. For these reasons, *system hardening* (especially, defending against known vulnerabilities) and *software defences* are appropriate for IaaS, including precautions such as anti-malware, firewalls and Intrusion Detection Systems. Provision of these features may be the responsibility of the Cloud Service Provider (CSP), who determines what OS and defensive capabilities are made available. In some settings, the

customer may be in a position to bolster the native defences on the virtual system provided by the CSP.

In similar vein, *data backup* is likely to be required by the IaaS customer. Indeed, the protection of customer data may jointly be the concern of the customer and the CSP. The former may enable off-Cloud backup, to avoid a single source of failure. While the CSP may also offer data backup to a separate Cloud data storage facility.

Despite reasonable expectation of such measures, there are indications that Cloud software infrastructure components are not always adequately secured from known vulnerabilities at the virtual machine level [20].

2. Platform as a Service

Computing facilities afforded to the customer of PaaS, are limited to the development of specific middleware or functional components. These services employ technologies such as Docker [21], Containers [22], DevOps [23] and AWS Lambda [24], in order to host customer-defined remote functionality. From a Cloud customer perspective, *system hardening* seems to be irrelevant in this context in relation to the host operating system. On the other hand, any code developed for use on the Cloud platform must be protected from illicit operations, e.g., process hijacking, output redirection or the elevation of privileges.

Software defences of the variety outlined above seem less relevant to the PaaS context since the operations supported by the middleware are limited to specific data processing and do not afford full operating system access or modification. The primary concern should be the operational effectiveness and resilience of the customer-defined operations. Clearly, such services may also be impaired through illicit access, e.g., stealing authentication details in order to alter code on the host system. Managing this area of concern lies primarily in the hands of the Cloud customer, with the assumption that the CSP will prevent unauthorised access to customer account details.

3. Software as a Service

SaaS provides the Cloud customer with remote access to third-party data processing facilities via micro-services [25] or RESTful services [26]. Aside from network level attacks, such services should be protected from most other security concerns by having the host system hardened and equipped with suitable software defences. From the customer perspective, so long as their remote Cloud services operate effectively, without interruption or data loss, there would seem to be little cause for concern. Of course, the risk of aberrant customer-side behaviour may arise through social engineering exploits or disgruntled employee actions.

This summary of security concerns affecting the three varieties of service has treated each Cloud model as an isolated networked computing facility. In reality, since the essence of Cloud provision is the virtualisation of services, our overview lacks one further important consideration, i.e., the possibility of service impairment as a result of activity at adjacent, upper or lower levels of the Cloud implementation.

Clearly, any security aspects that affect the operational resilience of the underlying Cloud infrastructure is of direct concern to the CSP and can have a knock-on effect upon customer services. The underlying Cloud technology, i.e., the hardware and software configurations that provision our three Cloud models, may be subject to attack or deliberate manipulation in a fashion that impinges detrimentally upon the Cloud services supported by that particular hardware and software ensemble. This may be construed as a service attack 'from below'. The scope for such attacks are precisely the characteristic exploits that may affect any networked host (listed earlier).

Attacks 'from the side' are a growing concern in Cloud security. 'Side channel attacks', originate with co-hosted customers who manipulate the behaviour of their virtual system to influence the behaviour of the host system and thereby affect co-hosted customers. Several studies suggest that such 'co-tenancy', an essential feature of IaaS and PaaS, carries dangers. Thus, "Physical co-residency with other tenants poses a particular risk" [27], such as "cache-based side-channel attacks" [28] and "*resource-freeing attacks* (RFAs)" in which "the goal is to modify the workload of a victim VM in a way that frees up resources for the attacker's VM" [29]. Most worrying are contexts where one customer's 'malicious' virtual machine seeks to extract information from another customer's virtual machine on the same Cloud platform [30]. Such risks to Cloud facilities are fundamental to their service provision.

A final attack vector that threatens some Cloud systems is 'from above'. In this case, poorly implemented virtual systems may afford scope for customers to 'break free' of their virtual system and access or directly affect the underlying operating system or middleware/hypervisor. Clearly, it must be ensured that there is no information leakage from virtual machines and that attackers or malicious customers are not capable of breaking out of the virtual machine and gaining access to the host OS or the virtual machines of other customers [31].

The characteristics of these Cloud service offerings with associated security measures and the likely risk conditions are captured in Table I. The prospect of action from one Cloud user affecting another is described as intra-platform interference.

VI. DIGITAL FORENSIC READINESS

Indications are that the number of cases of network intrusion and data breach is on the rise: "there is a massive increase in the records being compromised by external hacking – from roughly 49 million records in 2013 to 121 million and counting in 2015" [32].

One positive effect of this growth in unauthorized data access is the raised awareness of digital forensics (DF) and a marked change in its perception from a solely post-event reactive investigative tool to a pro-active policy to establish intelligence capabilities in advance of any incidents. This change in role reflects the concept of digital forensic readiness. Thus, "Pro-active DF management must ensure that all business processes are structured in such a way that essential data and evidence will be retained to ensure successful DF investigations, should an incident occur" [33, p.18].

One might define digital forensic readiness as 'having facilities in place to ensure the comprehensive capture and retention of all system event and user activity data that would be required post-incident in order to determine the precise nature of any data-loss, system modification or system impairment that results from intrusion, system misuse or system failure".

Naturally, this concept of digital forensic readiness is equally applicable to Cloud systems and novel techniques have been proposed to facilitate the data collection that this entails [34]. Yet, the Cloud context introduces particular problems with respect to forensic readiness.

Table I. Summary of features, security measures and risks

Service model	**Main features**	**Security Measures**	**Risks**
Infrastructure (IaaS)	Virtual machines, Operating systems, Storage, Software applications	System hardening, Software defences, Data backup	Social engineering, Intrusion, Malware, Denial of service, Elevation of privileges, *Intra-platform interference*
Platform (PaaS)	APIs, Programming languages, Development middleware, (Containers, Dockers, AWS Lambda, DevOps)	System hardening, Software defences	Social engineering, Elevation of privileges, *Intra-platform interference, Information leakage*
Software (SaaS)	Remote applications, Micro-services, RESTful services	System hardening, Software defences	Social engineering, *Intra-platform interference*

VII. MONITORING STRATEGIES

As previously noted, digital forensic readiness requires the monitoring and recording of events and activity that may impinge upon the integrity of the host system. Much of this capability is provided natively by the local system, using standardly available operating system logging, perhaps with additional active security monitoring, such as dynamic log analysis [35] or key file signature monitoring [36].

The situation for Cloud-based services reflects in many respects the context of a networked host. Where a customer employs Cloud purely as a storage medium, minimum security requirements will seek to ensure authenticated access and secure data backup. In turn, the monitoring requirements associated with this service must capture details of user logins (including source IP, username and success or failure of login attempts). Additionally, any file operations that change the status of data stored under the account of that customer must also be recorded. In the event of unauthorised access (e.g., stolen user credentials), such default monitoring may offer little protection, aside from identifying the identity of the stolen credentials and recourse to subsequent backup data recovery. Such monitoring is essentially operating system-based, albeit that in the Cloud setting, this OS may be virtual.

This context of Cloud usage faces the same challenges in monitoring and security that confront any networked host, with the added complication that a Cloud-based virtual host may face added vulnerability via its hosting virtualiser [37]. Furthermore, Cloud services are often configured to provide new virtual OS instances automatically to satisfy demand and in turn, shut these down when demand falls. A side-effect of such service cycling is that system logs are lost to the customer and subsequent digital forensic analysis may be unavailable.

In the 'traditional' network setting, numerous techniques have been devised to afford post-event insight on system failures and unwelcome exploits. In all major operating system contexts, whether virtualised, Cloud-based or native, system logging affords the baseline for generating auditable records of system, network and user activity. Such system level monitoring is well understood and in the event of intrusion is likely to be a primary target in order to compromise the record and eliminate traces of illicit activity.

For networked hosts and, by extension, as a monitoring strategy for local area networks, a wide-variety of Intrusion Detection Systems (IDS) have been developed and deployed with a view to rapid determination of malicious activity. These techniques may be rule-based [e.g., 38]. In most cases, the IDS monitors and cross-correlates system-generated logs in order to identify anomalous event sequences. Many approaches to anomaly-based intrusion detection have been reported [39]-[44]. Inevitably, such systems may themselves become targets in order to inhibit their detection capability and maintain a 'zero-footprint' on the part of the intruder [45].

In a Cloud context, each node is using its own logging daemon or agent to log important events. But in comparison to a single computer, the log information might be essential and therefore relevant for the whole Cloud infrastructure. For that reason, Cloud infrastructures use a centralised log server that receives the log information of all attached nodes. The task of this log server is not only the recording of log files of all nodes but also to monitor the Cloud infrastructure. In case of a cyber-attack, the log server ideally detects the attack (maybe assisted by an intrusion detection system) and starts countermeasures. This exposed role of the log server makes

it a very attractive target for cyber-attacks itself, or, as described above, means that an adversary has to deal with the log server in phase 2. Since the hardware of such a log server might also break down even without any cyber-attack, in practice more than one log server is used at the same time to provide redundancy.

A practical solution might consist of two log servers in "active-active-mode" which means that both are operating at the same time, but in case of one system failure, the other takes over for the whole Cloud infrastructure. The operation of these two log servers might be supervised by a third server which in case of failure or attack sends an alarm to the administrator. Unfortunately, the problem stays more or less the same: this third monitoring server is a single point of failure and is therefore attractive as a target for any adversary attacking the Cloud infrastructure. If an adversary manages to take out the monitoring server and to tamper with the log information on at least one of the two log servers, the Cloud provider might not be capable of determining which log files are correct and which are manipulated.

Any logging service that is introduced in addition to the traditional daemons or agents has to meet several constraints, including the following:

1. the new logging service must not cause too much additional load, either on the nodes (concerning computation) or on the network (concerning network traffic) and;
2. the computation of additional security measures in order to provide authenticity and integrity must be efficiently feasible.

VIII. Example Monitoring Approach

Message Authentication Codes (MACs) as described in almost any textbook about cryptography can readily be used to address this monitoring dilemma. MACs can be constructed using cryptographic hash functions or using block ciphers, for instance. Either construction ensures efficient computation of the MACs under a secret key. MACs are used to provide authenticity and integrity; therefore, they meet both conditions.

A solution that we propose starts with a secure boot process for each node of the Cloud infrastructure. During boot, the common log daemon or agent is started and it starts recording events in various log files. We suggest to compute a MAC for each event and to store these additional bits with the plaintext message of the event in the log file. We assume that the plaintext message also contains a time stamp. For the next event to be recorded in a log file, the plaintext of the event is concatenated with the previous MAC before computing the MAC for this event. This leads to a MAC chain which can be checked for each step using the plaintext and MAC of the previous event - but only if the secret key is known. Since the adversary does not know the secret key, he is not capable of computing valid MACs and therefore not capable of tampering with the MAC chain in order to hide his tracks.

The use of Message Authentication Codes is only the first step towards a solution to the problem. An adversary could simply delete or deliberately falsify all log files (including the MACs). This would probably make it impossible to reconstruct the steps of the cyber-attack in a post-hack analysis.

In order to deal with this issue and to make use of the benefits of a Cloud infrastructure, we propose the additional step of using secret sharing techniques - or so-called threshold schemes - as published by Adi Shamir in 1979 [46].

The idea is to divide some data D into n pieces $D_1, \dots, D_n$ in such a way that:

(a) D can be reconstructed easily of any $k < n$ pieces D_i

(b) the knowledge of only $k-1$ or even fewer pieces D_i leaves the data completely undetermined.

Shamir named such a scheme a "(k, n) threshold scheme". He points out that by using such a (k, n) threshold scheme with $n = 2k - 1$, it is necessary to have at least $k = \left\lceil \frac{n+1}{2} \right\rceil$ parts D_i to reconstruct D. A lesser number of $\left\lfloor \frac{n}{2} \right\rfloor = k - 1$ parts makes the reconstruction impossible.

Shamir introduced a (k, n) threshold scheme based upon polynomial interpolation. The data D can be interpreted as a natural number and p is a prime number with $D < p$. All of the following computations are made in the prime field $\mathrm{GF}(p)$. Given k points in the 2-dimensional plane, $(x_1, y_1), \dots, (x_k, y_k)$ with distinct coordinates x_i, there is one and only one polynomial q of degree $k - 1$ such that $q(x_i) = y_i$ for all $i = 1, \dots, k$. At first, the coefficients $a_1, \dots, a_{k-1}$ are chosen at random and $a_0 = D$, which leads to the polynomial

$$q(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}.$$

The n different pieces of D are computed as $D_1 = q(1)$, $D_i = q(i), \dots, D_n = q(n)$. Provided that their identifying indices are known, any subset of k elements D_i can be used to compute the coefficients a_i of the polynomial q which allow the computation of the data $D = q(0)$. From any subset of less or equal $k - 1$ pieces D_i, neither the coefficients a_i nor the data D can be calculated. (For further details, we direct the reader to the original paper [46].)

In our proposed solution to the problem of providing additional forensic information for post-hack analysis, D is the data to be written in a log file: the plaintext message of the event, n randomly chosen nodes of the Cloud infrastructure and the corresponding MAC, computed from the concatenation of the event message, the previous MAC and the addresses of these n nodes. The n pieces D_i that are derived from D as stated before and D is sent to the traditional centralised log server. The n pieces D_i are additionally sent to the n nodes which store this information. For the next event, we repeat this procedure but choose n (possibly) different nodes.

In case of a cyber-attack and if a post-hack analysis is necessary, at first all pieces of logging information are gathered from all nodes. Using the time stamps and the MAC chains, the order of the logged events can be reconstructed. The decentralised stored pieces of logging information are put together to reconstruct D from any k of the n parts. This

means, even if an adversary succeeds in manipulating some of the nodes and the centralised logging system, the events can be reconstructed. Finally, the integrity and authenticity of these events can be checked using the MAC chain.

The proposed approach may identify and retain information on an intruder's actions that result in stolen, modified or deleted data. This is a feature with growing importance, as legislative demands on data protection increase. For instance, the EU General Data Protection Regulation that is due to come into force in May 2018, will require companies to notify all breaches within 72 hours of occurrence, with a potential penalty of up to 4% of global turnover based on the previous year's accounts.

Note that this solution is not proposed as a general basis for monitoring the Cloud infrastructure. Rather, its purpose is to provide secure logging information for a post-hack analysis by distributing their parts randomly over all nodes. Thereby, reliable system monitoring can be established by means of multiple log servers, with the added assurance of Message Authentication Codes.

Now that we have a workable means of addressing the log data collection, robust storage and recovery of such data, we move to consider significant residual issues with digital forensic recovery in the Cloud context.

IX. Issues in Cloud Forensic recovery

Forensic readiness in the Cloud is complicated by the variety of contexts in which Cloud services are deployed and the diversity of software settings in which security risks may arise. Forensic readiness must accommodate these complexities and, in turn, this suggests that a single infrastructure-based digital forensic readiness solution may be infeasible.

The primary reason for concern is the need to capture relevant data on system operation at the various operational levels of the Cloud system and any potential interaction across these levels. This means capturing program logs, system logs and user activity logs. In any end-customer Cloud facility, the data protected may not extend beyond any currently live information and data held in associated database systems. The ready recycle capability of Cloud services also has implications for the persistence of digital forensic evidence. An intrusion that steals data from a virtual machine and then seeks to reset that machine may well succeed in destroying evidence of the intrusion, thereby removing any forensic traceability on the nature and quantity of stolen data.

Neither is it sufficient to provide each distinct operational layer of Cloud systems with its own comprehensive forensic readiness. At best, this condition will allow for forensic data recovery for that operational layer. But there is no one-size-fits-all solution that can capture all state, interaction and performance data such as would ensure full Cloud forensic recovery. In fact, this insight reveals a fundamental problem that may impact upon Cloud forensic readiness.

There are parallels here with issues in distributed systems and software architecture. Thus, "distributed software systems are harder to debug than centralized systems due to the increased complexity and truly concurrent activity that is possible in these systems" [47, p. 255]. Regardless of whether the Cloud setting is truly distributed in its realisation, its interconnected software functional layers represent a unique challenge when attempting to interpret the relationship between events or changes actioned at one functional level and the operational impact of such changes on other functional aspects of the services afforded by that Cloud.

When considering Cloud systems, from the perspective of software architecture there may be an assumption of 'a component- and message-based architectural style' in which there is 'a principle of limited visibility or *substrate independence*: a component within the hierarchy can only be aware of components "above" it and is completely unaware of components which reside "beneath" it' [48, p.825].

This multi-level interpretation problem is complicated by the fact that events considered anomalous at one level of service offering may arise through actions considered legitimate at a 'lower' level of software implementation. From the digital forensic readiness perspective, this underlines the requirement to go beyond capture of significant events across the Cloud service software and functional levels, since significance is an aspect that may cross the boundaries between such layers in the system as a whole. A few hypothetical examples may clarify this issue.

In our first example, a CSP customer may contract access to specific functional components (e.g., a Web service). The operational characteristics of the service are under the control of the CSP and not the customer. An authorised employee of the CSP may modify the algorithmic process and thereby affect the outcome of any service use by the customer. While a change in operational behaviour of the service (i.e., an anomaly) may eventually be detected by the customer, there may be no anomalous activity evident at the level of CSP employee activity. The focus of subsequent forensic investigation may light initially on the nature of customer activity, since this is where the anomaly is apparent, but proper understanding of the issue requires that events across different functional levels of the Cloud system be apprehended.

In our second example, an employee of the CSP illicitly establishes a clone of the live customer system, with all data records in the customer system continuously duplicated, updated and available to the CSP employee. Here, data records are being accessed without authorisation and this fact is both unknown and unavailable to the Cloud customer. Insight from the operational level of the CSP would be required in order to expose this situation. Yet, the Cloud customer may come under scrutiny or be subject to litigation if critical customer data is made available on the Dark Web.

An informative view on this issue of informational levels may be borrowed from Granular Computing [49], which aims to develop computational models of complex systems, such as human intelligence. A key characteristic of this work is that it 'stresses multiple views and multiple levels of understanding in each view' [op. cit., p.85]. Here, the emphasis is upon 'holistic, unified views, in contrast to isolated, fragmented views. To achieve this, we need to consider multiple hierarchies and multiple levels in each hierarchy' [op. cit., p.88].

Our proposal for adequate Cloud forensic readiness has two components (detailed above). Firstly, there is a requirement for data capture. This is the obvious need to record any data at each layer of Cloud facility that may have a role to play in subsequent digital forensic analysis. Secondly, the captured data must be stored off the system being monitored in a manner that both ensures the integrity of the logging and minimises the likelihood that the stored data can be compromised, either as a result of hostile action or 'friendly fire'.

Our requirement for secure and resilient log storage can build upon default system logging that will be present within the Cloud implementation but this must be supplemented to achieve log reliability.

Instead of using centralised log servers, which of course are attractive targets and easy to spot for attackers, we propose a different approach. In order to prevent adversaries from manipulating log files to hide their tracks, we use chained Message Authentication Codes (MACs) for each entry to the log file on each node. If state-of-the-art MACs are used, this makes it impossible to delete or manipulate text in the log files. Next, each node uses secret sharing techniques, such as that proposed by Adi Shamir [46], to divide the log file into parts. These parts are then sent to random other nodes which store these log data. Even if an adversary succeeds in taking over some of the nodes, he will need a certain number of these fragments to reconstruct the log data. But since for each log entry different nodes are chosen randomly as stated before, the attacker effectively needs to control the whole Cloud ecosystem to stay hidden.

X. CONCLUSIONS

Recognising the importance of securing log data as a basis for digital forensic reconstruction in the event of system intrusion, a multiple server solution combined with Message Authentication Codes affords a mechanism that allows for safe deposit and reconstruction of monitor data. This can operate in a Cloud setting in which each logging node is a virtual server.

An important benefit from this distributed solution is that digital forensic reconstructions are possible for virtual machines that are 'cycled', since their native OS logs can be maintained in a recoverable and verifiable form beyond the OS of those machines. This provides the safeguard of digital forensic readiness for Cloud customers in the event that an intruder accesses private data on the Cloud service and causes that system to cycle as an attempt to delete all traces of illicit data access.

The possibility, however slight, that an intruder may gain access to and potentially compromise all peers in this configuration, can be mitigated by also allowing log data to transfer 'upwards' to one or more 'superior' systems (e.g., the parent operating systems in which the peer log servers are virtualised).

As organisations move increasingly away from locally hosted computer services toward Cloud platforms, there is a corresponding need to ensure the forensic integrity of such instances. The primary reasons for concern are the locus of responsibility and the associated risk of legal sanction and financial penalty. In the first place, while Cloud service providers (CSPs) are responsible for the availability and robustness of their commercial offerings, they will not be responsible for the management of such services by their customers, nor for the data security associated with customer-level use of the Cloud services. Responsibility for these aspects resides with the CSP's customers, whose data processing and data management are built upon the purchased Cloud services. In the second place, legislative demands on data protection, such as the EU General Data Protection Regulation [50], requires companies to notify all breaches within 72 hours of discovery or face significant financial penalty.

These concerns can be addressed and the business risk mitigated through development of forensic readiness in customer-level Cloud systems (as described above). We have argued that this requires a range of logging and data capture facilities across the Cloud system software infrastructure that maintain the possibility of tracking activity at different levels of software abstraction (the multi-level interpretation problem). Our second proposition is that such digital forensic readiness must be combined with techniques to ensure that logged data is incorruptible and robust. We have previously proposed techniques for intrusion monitoring that ensure log data credibility and provide robust decentralised log storage and recovery for post-hack scenarios.

To achieve adequate data capture, we require 'state information' and data management across differing levels of any Cloud service, from the lowest software level up to the most abstracted 'user facing' software component. On their own, such records will not be sufficient to fully capture the potential interplay of differing software levels. For this purpose, subsequent digital forensic analytics will be required in order to establish a multi-dimensional representation of event chronology. This means that timestamps from events and data captured at different software levels of abstraction will need to be correlated in order to determine how events across the Cloud system are related.

Cloud service provision has a requirement for secure and robust monitoring with access to multiple levels of such monitoring data. If we are able to supplement our robust monitoring and logging approach with appropriate levels of Cloud operational information (e.g., as a feature of Cloud Service Level Agreements), this may in turn facilitate a solution to the multi-level interpretation problem and take us all the way to effective digital forensic readiness. Thereby, we may achieve a Cloud facility that is capable of successful recovery from accidents and incidents, to afford effective management of digital forensic recovery.

REFERENCES

[1] G. R. S. Weir, A. Aßmuth and N. Jäger, "Managing forensic recovery in the Cloud", In: Proc. *Cloud Computing 2018*, *The Ninth International Conference on Cloud Computing*, GRIDs and Virtualization, IARIA, Barcelona, Spain, 2018.

[2] G. R. S. Weir and A. Aßmuth, "Strategies for Intrusion Monitoring in Cloud Services", In Proc. *Cloud Computing 2017, The Eighth International Conference on Cloud*

Computing, GRIDs and Virtualization, IARIA, Athens, Greece, 2017.

[3] M. Nanavati, P. Colp, B. Aiello and A. Warfield, "Cloud security: A gathering storm", *Communications of the ACM*, *57*(5), pp. 70-79, 2014.

[4] S. S. Tirumala, H. Sathu and V. Naidu, "Analysis and prevention of account hijacking based incidents in Cloud environment", In Proc. *International Conference on Information Technology (ICIT),* pp. 124-129, 2015.

[5] M. Zhou, R. Zhang, W. Xie, W. Qian and A. Zhou, "Security and privacy in Cloud computing: A survey", In Proc. *Sixth International Conference on Semantics Knowledge and Grid (SKG),* pp. 105-112, 2010.

[6] Y. Chen, V. Paxson and R. H. Katz, "What's new about Cloud computing security", *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, *20*, 2010.

[7] BBC News, Available: http://www.bbc.co.uk/news/technology-29076899. [Accessed: Dec. 29, 2017].

[8] P. Mell and T. Grance, "The NIST definition of Cloud computing", NIST, 2011. Available from http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf, [retrieved: February, 2017].

[9] S. A. Shaikh, H. Chivers, P. Nobles, J. A. Clark and H. Chen, "Network reconnaissance", *Network Security*, vol. 11, pp. 12-16, 2008.

[10] L. Richardson and S. Ruby, *RESTful Web Services*. O'Reilly Media, Inc., 2008.

[11] B. Benatallah, Q. Z. Sheng and M. Dumas, "The self-serv environment for web services composition", *IEEE Internet Computing*, vol. 7, no. 1, pp. 40-48, 2003.

[12] B. F. Murphy, Network Penetration Testing and Research, NASA, 2013. Available from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140002617.pdf, [retrieved: February, 2017].

[13] J. Andress and S. Winterfeld, *Cyber warfare: techniques, tactics and tools for security practitioners.* Elsevier, 2013.

[14] H. Takabi, J. B. Joshi and G. J. Ahn, "Security and privacy challenges in Cloud computing environments", *IEEE Security & Privacy*, 8(6), pp. 24-31, 2010.

[15] M. Carroll, A. Van Der Merwe and P. Kotze, "Secure Cloud computing: Benefits, risks and controls", In *Proc. Information Security South Africa* (ISSA), 2011, pp. 1-9, 2011.

[16] K. Ilgun, R. A. Kemmerer and P. A. Porras, "State transition analysis: A rule-based intrusion detection approach", *IEEE Transactions on Software Engineering*, vol. 21, no. 3, pp. 181-199. 1995.

[17] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Computers and Security*, vol. 28, no. 1, pp. 18-28, 2009.

[18] J. Gibson, R. Rondeau, D. Eveleigh and Q. Tan, "Benefits and challenges of three Cloud computing service models", In *Proc. Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, pp. 198-205, 2012.

[19] H. Sweetland Edwards, "How Web Cams Helped Bring Down the Internet, Briefly", Time Magazine, 25th October 2016. Available: http://time.com/4542600/internet-outage-web-cams-hackers. [Accessed: Dec. 29, 2017].

[20] S. Zhang, X. Zhang and X. Ou, "After we knew it: empirical study and modeling of cost-effectiveness of exploiting prevalent known vulnerabilities across IAAS Cloud", In *Proc. 9th ACM Symposium on Information, Computer and Communications Security,* pp. 317-328, 2014.

[21] S. Dhakate and A. Godbole, "Distributed Cloud monitoring using Docker as next generation container virtualization technology", In Proc. *Annual IEEE India Conference (INDICON),* pp. 1-5, 2015.

[22] C. Pahl and B. Lee, "Containers and clusters for edge Cloud architectures--a technology review. In Proc. *3rd International Conference on Future Internet of Things and Cloud (FiCloud),* pp. 379-386, 2015.

[23] A. Balalaie, A. Heydarnoori and P. Jamshidi, "Microservices architecture enables DevOps: migration to a Cloud-native architecture", *IEEE Software*, *33*(3), pp. 42-52, 2016.

[24] M. Villamizar et al., "Infrastructure cost comparison of running web applications in the Cloud using AWS lambda and monolithic and microservice architectures. In Proc. *16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid),* pp. 179-182, 2016.

[25] D. Namiot and M. Sneps-Sneppe, "On micro-services architecture", *International Journal of Open Information Technologies*, *2*(9), pp. 24-27, 2014.

[26] H. Han et al., "A RESTful approach to the management of Cloud infrastructure", In Proc. *IEEE International Conference on Cloud Computing,. CLOUD'09.,* pp. 139-142, 2009.

[27] Y. Zhang, A. Juels, A. Oprea and M. K. Reiter, "Homealone: Co-residency detection in the Cloud via side-channel analysis", In Proc. *IEEE Symposium on Security and Privacy (SP),* pp. 313-328, 2011.

[28] Y. Zhang, A. Juels, M. K. Reiter and T. Ristenpart, "Cross-tenant side-channel attacks in PaaS Clouds". In Proc. *ACM SIGSAC Conference on Computer and Communications Security,* pp. 990-1003, 2014.

[29] V. Varadarajan, T. Kooburat, T., Farley, T. Ristenpart and M. M. Swift, "Resource-freeing attacks: improve your Cloud performance (at your neighbor's expense)", In Proc. *ACM conference on Computer and communications security*, pp. 281-292, 2012.

[30] Y. Zhang, A. Juels, M. K. Reiter and T. Ristenpart, "Cross-VM side channels and their use to extract private keys", In Proc. *ACM conference on Computer and communications security,* pp. 305-316, 2012.

[31] T. Vateva-Gurova, N. Suri and A. Mendelson, "The Impact of Hypervisor Scheduling on Compromising Virtualized Environments", In Proc. *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pp. 1910-1917, 2015.

[32] Security Week, "Data Breaches by the Numbers", Available: http://www.securityweek.com/data-breaches-numbers. [Accessed: Dec. 29, 2017].

[33] C. P. Grobler and C. P. Louwrens, "Digital forensic readiness as a component of information security best practice", In Proc. *IFIP International Information Security Conference*, pp. 13-24, Springer, Boston, MA, 2007.

[34] V. R. Kebande and H. S. Venter, "A Cloud Forensic Readiness Model Using a Botnet as a Service", In Proc. *The International Conference on Digital Security and Forensics (DigitalSec2014)*, pp. 23-32, The Society of Digital Information and Wireless Communication, 2014.

[35] A. Oliner, A. Ganapathi and W. Xu, "Advances and challenges in log analysis", *Communications of the ACM*, vol. 55, no. 2, pp. 55-61, 2012.

[36] G. H. Kim and E. H. Spafford, "The design and implementation of tripwire: A file system integrity checker", Proceedings of the *2nd ACM Conference on Computer and Communications Security*, ACM, pp. 18-29, 1994.

[37] J. S. Reuben, *A survey on virtual machine security*. Helsinki University of Technology, vol. 2, no. 36, 2007.

[38] K. Ilgun, R. A. Kemmerer and P. A. Porras, "State transition analysis: A rule-based intrusion detection approach", *IEEE*

Transactions on Software Engineering, vol. 21, no. 3, pp. 181-199, 1995.

[39] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Computers and Security*, vol. 28, no. 1, pp. 18-28, 2009.

[40] C. Chapman, S. Knight and T. Dean, USBcat-Towards an Intrusion Surveillance Toolset, arXiv preprint arXiv:1410.4304, 2014.

[41] X. Wang, D. S. Reeves, S. F. Wu and J. Yuill, "Sleepy watermark tracing: An active network-based intrusion response framework", *Trusted Information*, Springer US, pp. 369-384, 2002.

[42] C. V. Zhou, C. Leckie and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection", *Computers and Security*, vol. 29, no. 1, pp. 124-140, 2010.

[43] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends", *Computer Networks*, vol. 51, no. 12, pp. 3448-3470, 2007.

[44] H. Sukhwani, V. Sharma and S. Sharma, "A Survey of Anomaly Detection Techniques and Hidden Markov Model", *International Journal of Computer Applications*, vol. 93, no. 18, pp. 26-31, 2014.

[45] G. Tedesco and U. Aickelin, Strategic alert throttling for intrusion detection systems, arXiv preprint, arXiv:0801.4119, 2008.

[46] A. Shamir, "How to share a secret", *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[47] P. C. Bates and J. C. Wileden, "High-level debugging of distributed systems: The behavioral abstraction approach", *Journal of Systems and Software*, *3*(4), pp. 255-264, 1983.

[48] N. Medvidovic, R. N. Taylor and E. J. Whitehead Jr, "Formal modeling of software architectures at multiple levels of abstraction", *ejw*, *714*, pp. 824-837, 1996.

[49] Y. Yao, "Perspectives of granular computing. In Proc. *IEEE International Conference on Granular Computing,* Vol. 1, pp. 85-90, 2005.

[50] Regulation, Protection. "Regulation (EU) 2016/679 of the European Parliament and of the Council." *REGULATION (EU)*(2016): 679, 2016.