Cloud Accounting Systems, the Audit Trail, Forensics and the EU GDPR: How Hard Can It Be?

George Weir
Department of Computer and Information Sciences
University of Strathclyde
Glasgow, UK
Email: george.weir@strath.ac.uk

Andreas Aßmuth
University of Applied Sciences
OTH Amberg-Weiden
Germany
Email: a.assmuth@oth-aw.de

Mark Whittington
Business School
University of Aberdeen
Aberdeen, UK
Email: mark.whittington@abdn.ac.uk

Bob Duncan
Business School
University of Aberdeen
Aberdeen, UK
Email: bobduncan@abdn.ac.uk

Abstract

Ahead of the introduction of the EU General Data Privacy Regulation (GDPR), we consider some important unresolved issues with cloud computing, namely, the insecure cloud audit trail problem and the challenge of retaining cloud forensic evidence. Developing and enforcing good cloud security controls is an essential requirement for this is to succeed. The nature of cloud computing architecture can add additional problem layers for achieving cloud security to an already complex problem area. Historically, many corporates have struggled to identify when their systems have been breached, let alone understand which records have been accessed, modified, deleted or ex-filtrated from their systems. Often, there is no understanding as to who has perpetrated the breach, meaning it is difficult to quantify the risk to which they have been exposed. The GDPR seeks to improve this situation by requiring all breaches to be reported within 72 hours of an occurrence, including full identification of all records compromised, failing which the organisation could be subject to punitive levels of fines. We consider why this is such an important issue, identifying what desirable characteristics should be aimed for and propose a novel means of effectively and efficiently achieving these goals. We have identified a range of issues which need to be addressed to ensure a robust level of security and privacy can be achieved. We have addressed these issues in both the context of conventional cloud based systems, as well as in regard to addressing some of the many weaknesses inherent in the internet of things. We discuss how our proposed approach may help better address the identified key security issues.

Index Terms

Cloud security and privacy; cloud audit; cloud forensics; Internet of Things.

1. Introduction

In this work, we consider the use of cloud based systems for running accounting systems in the light of the unresolved issues of weak audit trail and vulnerable forensic data in cloud systems. Both of these form vital components of ensuring the veracity of cloud accounting systems data. In the light of the forthcoming EU GDPR, EU [2017], it is incumbent on companies to ensure that company data is maintained securely. In the event that the company is subject to a security breach, there is a requirement in the new legislation to report the breach within 72 hours of the occurrence.

It is well known that in cloud systems, once an intruder breaches a system, it can be particularly easy for them to obfuscate the trail of their ingress by tampering with or deleting both audit trail and forensic data. While some intruders have the finesse to do this without leaving a trace of their presence, many others are less skilled. The resultant impact of their actions, including the loss of audit trail and forensic data, can have catastrophic implications for the company, both in terms of satisfying statutory audit requirements and business continuity.

The audit trail and forensic data are targeted in all systems, but usually, in traditional distributed systems, it is easier to spot than in cloud systems, where instances are often spooled up and down with great frequency. Incorrect configuration can cause both audit trail and forensic data to be very easily lost, without the possibility of recovery. In any event, accessing and abstracting forensic data from a virtual hard drive can be all but impossible when the cloud service provider (CSP) often has no idea of where the data is physically stored within their datacenter. However, once an instance is shut down, there is also very little chance of recovery, since the CSP will seek to sell the freed up space to another customer as quickly as possible. This often happens in seconds in a busy datacenter.

When we consider the results presented by security breach companies, it is clear that there is a considerable time for the majority of companies between breach and discovery. While this was estimated some 5 years ago by Trustwave Trustwave

[2012], at 6 months, Verizon suggest that today, typical time-to-compromise continues to be measured in minutes, while time-to-discovery remains in weeks or months, Verizon [2017]. Given that time to discovery is on average considerably greater than 72 hours, this certainly remains a major concern.

The GDPR introduces a requirement to report on any breach within 72 hours of that breach and the company is also required to explain which records were accessed, modified or deleted in that breach. Clearly, this would seem to present a difficulty for the vast majority of companies. Given that 27% of companies still do not detect a breach themselves, Verizon [2017], but instead, are informed by third parties, this places them immediately in a position of non-compliance. With fines for a first offence pitched at the greater of €10 million or 2% of Global turnover, this will attract some serious attention. Considering that some companies are breached every day, it is clear that these costs might accumulate rather rapidly, especially as fines can double for persistent non-compliance.

Thus our approach is to ensure a good level of security can be achieved, and in Section 2, we discuss how earlier work on the use of an immutable database Duncan and Whittington [2017], may be deployed to help us achieve this difficult goal. On its own, it will be unlikely to solve the cloud security problem, but it will at least allow us to retain some usable audit trail and forensic data. In Section 3, we consider how adding a forensic detection system based on earlier work Weir and Aßmuth [2017], would attempt to catch the point of entry of the intruder. In Section 4, we discuss why the combination of these two techniques can provide a far greater level of security in the cloud. In Section 5, we discuss our conclusions.

2. The Power of the Immutable Database

In any business, there are many areas of activity which need diligent checking and verification by an objective external person or organization. Some of these audits are undertaken voluntarily by the firm, while others such as the audit of financial systems and results are mandatory. Cloud computing audit is a new, immature field and it would be surprising if there were not lessons to learn from the experiences — and failures — of audit processes and practices in other fields that have been honed over decades if not centuries Duncan and Whittington [2014].

It is hard to find people with the appropriate skillset to undertake appropriate quality audit whenever any new technical area emerges, as they need a technical knowledge of the area combined with competency in carrying out an audit. Being commercial organisations, audit companies have sought to extend their audit competence into new technical areas, not just cloud audit, but also environmental audit as another example. Over a century of experience in the development of audit tools and practices is then required to be applied to a new technical domain. Alternatively, computing specialists might pick up an audit skillset. A logical outcome would be for audit firms to recruit computer cloud experts and seek to harmonise their skills with those of audit already embedded in the firm. The culture clash between accountants and cloud experts would be a potential side effect from such a strategy Duncan and Whittington [2017].

One tool long used by accountants is the audit trail and this is a phrase already in the cloud computing literature courtesy of the National Institute of Standards and Technology (NIST) Guttman and Roback [2011], for example. However, the same phrase may not carry the same meaning in both settings. Quoting from the definitions of "audit trail" in the Oxford English Dictionary (OED) OED [2016]: "(a) Accounting: a means of verifying the detailed transactions underlying any item in an accounting record; (b) Computing: a record of the computing processes that have been applied to a particular set of source data, showing each stage of processing and allowing the original data to be reconstituted; a record of the transactions to which a database or a file has been subjected". This disparity of definition is thus recognized by the OED. Accountants are members of professional bodies (some national, some global) that limit membership to those who have passed exams and achieved sufficient breadth and length of experience that they are deemed worthy to represent the profession. Audit is a key feature of these exam syllabi and the tracing back to the source of each accounting activity (the trail) is a foundational aspect of audit. Attacks involving insiders are not considered, but even people who "are deemed worthy to represent the profession" might be corrupted or threatened to help adversaries, a point worth bearing in mind.

Whilst a clear explanation of an audit trail in a computing security setting is provided by Guttman and Roback [2011], but in keeping with the OED definition (b), the understanding of the term in cloud audit research seems less precise and consistent. For example, Bernstein et al. [2009], see the trail including: events, logs, and the analysis of these, whilst Chaula [2006], gives a longer, more detailed list: raw data, analysis notes, preliminary development and analysis information, processes notes, and so on. Indeed, Pearson and Benameur [2010], accepts that the attaining of consistent, meaningful audit trails in the cloud is a goal rather than reality. More worryingly Ko et al. [2011], points out that it is quite possible for an audit trail to be deleted along with a cloud instance, meaning no record then remains to trace back, understand and hold users to account for their actions and Soares et al. [2014], then detail the requirements for accountability. Indeed, the EU Article 29 Working Party Data Protection Working Party [2012], highlights poor audit trail processes as one of the security issues inadequately covered by existing principles.

Once established, an audit trail must have its contents protected from any adjustment. As Anderson [2008], points out, even system administrators must not have the power to modify it. This is not only good practice even with well-trained and ethical

individuals, but it is always possible that an attacker might be able to gain access, followed by leveraging their permissions to the point where they can attain administrator status to alter or delete the audit trail. Therefore, the audit trail must be protected through the establishment of an immutable database (i.e., one that only records new activities but never allows adjustment of previous ones).

In previous work, Duncan and Whittington [2016], warned of the problems that might follow from failure to recognise the importance of the audit trail, and in Duncan and Whittington [2017], a solution was proposed to this problem for cloud by utilising an immutable database to support the provision of an independent audit trail, to which they suggested that all system logs should also be added. This would satisfy the long understood accounting requirements of an audit trail, namely that no user, not even system administrators should be able to gain access to the audit trail. Thus there is a requirement to utilise an immutable database (i.e., one that only records new activities but never allows adjustment of previous ones).

This satisfies the primary goal for the successful development of a system to preserve both the audit trail and system logs. Secondly, relying on the principle of separation of duties, by placing this immutable database into a completely separate system, this removes the danger of a leveraged attack once a system is compromised.

Naturally, this will make the immutable database itself a prime target for attack. However, the machine on which the immutable database runs would be set up to have no external web services or other means of direct access. There should be full security on this machine, including an intrusion detection system (IDS), and it should also run a permanent monitoring system to instantly detect anomalies, alerting the necessary people in authority of such activities.

But we already produce an audit trail in our accounting system, we hear you say. Why would we require another one? There is a perfectly good explanation for that. First, we have already highlighted the issues with retaining an audit trail in cloud, particularly where instances are freely spooled up, and down, to follow demand. Few people make proper provision to retain the full audit and forensic trail in these cases, meaning the full trail is frequently lost when the instances are shut down. Secondly, we recommend the original audit trail and forensic trails are retained. Intruders expect to be able to find the audit trail and the forensic trail, so that they can erase their tracks, however poorly they might carry this out. Absence of these will signal to any potential intruder that someone has already been in and they may well quickly depart. However, where the intruder thinks they are in control, they will remain for longer periods, leaving greater amounts of forensic evidence behind.

This leads us to the next Section where we will consider the addition of a forensic attack detection method, based on earlier work Weir and Aßmuth [2017], which attempts to catch the point of entry of the intruder.

3. Forensic Attack Detection

As a result of the associated economy and convenience, many mission-critical services are moving to Cloud implementations. This is also regarded as a strategy for limiting security concerns and assuring greater resilience. The virtual nature of Cloud services means that system recovery or replacement can be quick, reliable and cost-effective Benatallah et al. [2003].

For many Cloud service users, security is limited to concerns of authorised access, continuity of service and data maintenance. In contrast, if the customer uses the Cloud service as a means of computation, security extends to all traditional aspects, including data protection, access authentication, service misappropriation and service availability. The Cloud Service Provider (CSP) has ultimate management of the infrastructure that affords all of the higher-level service provision. 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 relied 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. This leads to the issue of digital forensic readiness as applied to the Cloud context.

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 Oliner et al. [2012], or key file signature monitoring Kim and Spafford [1994].

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, and in the Cloud setting, this OS may be virtual.

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 provide post-event insight and system logging affords the baseline for generating auditable records of system, network and user activity. Such monitoring is well understood and in the event of intrusion is likely to be a primary target in order to compromise any traces of the illicit activity.

In a Cloud context, each node uses its own logging daemon or agent to record important events. But in comparison to a single computer, the log information might be essential and therefore relevant for the whole Cloud infrastructure. For this reason, Cloud infrastructures use a centralised log server that receives log information from all attached nodes. The centralised log server not only records log files from all nodes but also has a role in monitoring the Cloud infrastructure. In case of a network attack, the log server ideally detects the attack (maybe assisted by an intrusion detection system) and starts countermeasures. Of course, this exposed role of the log server makes it an attractive target for cyber-attacks itself. Since the hardware of such a log server might also break down without any cyber-attack, in practice more than one log server is used at a 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, this third monitoring server is a single point of failure and is thereby an attractive target for any attack on the Cloud infrastructure. If an adversary manages to take out the monitoring server and tamper with the log information on at least one of the two log servers, the Cloud provider might be unable to distinguish which log files are correct and which are manipulated.

Example Monitoring Approach

By employing cryptography, Message Authentication Codes (MACs) can be used to address this monitoring issue. MACs are commonly derived by applying a mathematical function to a given data item, in combination with a secret key or password. Where a password is used, then a key derivation function is needed to produce a "good" key from the password. The resultant MAC can accompany the data item and affords a way of checking the authenticity and integrity of that data.

Authenticity is determined when the received MAC matches a newly generated MAC created using a key that corresponds to the expected sender. Integrity is determined (i.e., the content of the data has not changed since the accompanying MAC was created) when the received MAC matches a MAC that has been newly computed by applying the mathematical function to the incoming data item using the secret key.

Our proposed solution starts with a secure boot process for each node of the Cloud infrastructure. During boot, the common logging agent starts recording events in various log files. We suggest to compute a MAC for each event and to store these MACs 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 any adversary would 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 one step towards a solution to the problem of robust monitoring. An adversary could simply delete or falsify the log files (including the MACs). This may make it impossible to reconstruct the cyber-attack in a post-event analysis.

We deal with this issue and make use of the benefits of a Cloud infrastructure, by proposing the additional step of using secret sharing techniques. A suitable technique is available in 'threshold schemes' Shamir [1979]. This mathematical approach would divide some data D into n pieces $D_1, ..., D_n$ 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 (op. cit.) 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 $k = \lceil \frac{n+1}{2} \rceil$ parts D_i to reconstruct D. A lesser number of $\lfloor \frac{n}{2} \rfloor = k-1$ parts makes the reconstruction impossible.

Shamir introduced a "(k, n) threshold scheme" based on polynomial interpolation whereby the data D is 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. Given k points in the 2-dimensional plane, $(x_1, y_1), ..., (x_k, y_k)$ 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 $i = 1, ..., , k$. At first, the coefficients $a_1, ..., a_{k-1}$ are chosen at random and $a_0 = D$, which leads to the polynomial $q(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{k-1} x^{k-1}$.

The n different pieces of D are computed as $D_1 = q(1), D_i = q(i), ..., D_n = q(n)$. Provided that their identifying indices are known, any subset of k elements D_1 can be used to compute the coefficients a_i of the polynomial q which allow for 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.

In our solution to the problem of providing robust access to forensic information, D is the data to be written in a log file (the plaintext message of the event), n is a randomly chosen number of nodes in the Cloud infrastructure and we also deploy a MAC (computed from the concatenation of the event message, the previous MAC and the addresses of the n nodes). The n pieces D_i that are derived from D (using the appropriate secret sharing algorithm), and D are 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.

If there is a cyber-attack and a post-hack analysis is required, 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 distributed stored pieces of logging information are used to reconstruct D from any k of the n parts. Thereby, even if an intruder seeks to remove all trace of their presence and affects some of the nodes and the centralised logging system, the logged events can still be reconstructed. In addition, the integrity and authenticity of these events can be checked using the MAC chain (as described above).

Importantly, our proposed approach could identify and retain information on an intruder's actions that may have resulted in stolen, modified or deleted data. This feature will become vital, as legislative demands on data protection increase with the EU General Data Protection Regulation.

By combining Message Authentication Codes with secret-sharing techniques, we can provide a robust mechanism that affords secure logging information for post-event analysis.

As an added measure of security, we also pass a copy of this forensic information to our immutable database. While many might say this is additional duplication, if an intruder gets into a system and removes the forensic trail, it is highly unlikely that the intruder will successfully be able to gain entry into all elements of the system. Whereas, without the immutable database, if the intruder removes all the forensic data — it is game over.

4. Why the Combined Approach is Much More Secure

Now, we have a system which provides an additional means of recording both the audit trail and the forensic trail for the whole cloud system, regardless of the size of the system, or the frequency with which instances are spooled up or down. To this, we have added a neat forensic detection system, which will alert us to any attempt to breach our system. This ensures that our system will be robust. No system is ever likely to be foolproof, but in this case, we will be able to identify the point of entry by any intruder. We will have an independent audit and forensic trail retained in a separate secure location.

This means that contrary to most systems currently in use, we will be aware that we have been breached. We will not have to wait the seemingly normal weeks or months to find out from somebody else that we have been breached. However, we will also be in a much stronger position if we are breached, because we will know exactly which records have been accessed, which may have been modified, and which may have been deleted, thus providing us with all the information we need to comply with the reporting requirements of the GDPR, which we can now easily perform within the required 72 hours.

The added advantage, is that we are also in possession of a complete forensic trail, and this means steps may be taken by the relevant authorities to pursue the criminal intruders. Also, building up forensic attack profiles will be invaluable in preparing better defences against such attacks.

5. Conclusions

As we have seen, ensuring a proper audit and forensic trail is properly maintained in the light of using cloud accounting systems, is very hard indeed. For the vast majority of companies, compliance with the GDPR is currently nothing more than a pipe dream, thus exposing them to unsustainable levels of fines.

We believe that the state of readiness for companies who use cloud based accounting systems to comply with the forthcoming GDPR legislation which comes into effect in May next year EU [2017], is appallingly poor. For the vast majority of companies who will not even be able to tell that they have been breached for an average of 146 days /citeVerizon2016, they are likely to be in breach from the moment the regulation takes effect. This means they are effectively placing themselves at risk of receiving potentially huge levels of fines.

We propose a simple, straightforward and relatively inexpensive approach to addressing this potential ticking timebomb. Make no mistake, this regulation has serious teeth, and any company unprepared to achieve instant compliance is walking into a potential financial minefield.

REFERENCES

Anderson, R. J. (2008). *Security Engineering: A Guide to Building Dependable Distributed Systems*, volume 50. Wiley.

Benatallah, B., Sheng, Q. Z., and Dumas, M. (2003). The self-serv environment for web services composition. *IEEE Internet Computing*, 7(1):40–48.

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., and Morrow, M. (2009). Blueprint for the intercloud - Protocols and formats for cloud computing interoperability. In *Proceedings of the 2009 4th International Conference on Internet and Web Applications and Services, ICIW 2009*, pages 328–336.

Chaula, J. A. (2006). *A Socio-Technical Analysis of Information Systems Security Assurance: A Case Study for Effective Assurance*. PhD thesis.

Data Protection Working Party (2012). Opinion 05/2012 on Cloud Computing.

Duncan, B. and Whittington, M. (2014). Compliance with standards, assurance and audit: does this equal security? In *Proceedings of the 7th International Conference on Security of Information and Networks - SIN '14*, pages 77–84, Glasgow. ACM.

Duncan, B. and Whittington, M. (2016). Enhancing Cloud Security and Privacy: The Power and the Weakness of the Audit Trail. In *Cloud Computing 2016: The Seventh International Conference on Cloud Computing, GRIDs, and Virtualization*, number April, pages 125–130, Rome. IEEE.

Duncan, B. and Whittington, M. (2017). Creating an Immutable Database for Secure Cloud Audit Trail and System Logging. In *Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 54–59, Athens. IARIA, ISBN: 978-1-61208-529-6.

EU (2017). EU General Data Protection Regulation.

Guttman, B. and Roback, E. A. (2011). NIST Special Publication 800-12. An Introduction to Computer Security: The NIST Handbook. Technical Report 800, NIST.

Kim, G. H. and Spafford, E. H. (1994). The design and implementation of tripwire: A file system integrity checker. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pages 18–29. ACM.

Ko, R. K. L., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., and Lee, B. S. (2011). TrustCloud: A framework for accountability and trust in cloud computing. *Proceedings - 2011 IEEE World Congress on Services, SERVICES 2011*, pages 584–588.

OED (2016). Oxford English Dictionary.

Oliner, A., Ganapathi, A., and Xu, W. (2012). Advances and challenges in log analysis. *Communications of the ACM*, 55(2):55–61.

Pearson, S. and Benameur, A. (2010). Privacy, Security and Trust Issues Arising from Cloud Computing. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, number December, pages 693–702. Ieee.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

Soares, L. F. B., Fernandes, D. a. B., Gomes, J. V., Freire, M. M., and Inácio, P. R. M. (2014). Security, Privacy and Trust in Cloud Systems. In Ko, R., editor, *Security, Privacy and Trust in Cloud Systems*, chapter Data Accou, pages 3–44. Springer.

Trustwave (2012). 2012 Global Security Report. Technical report.

Verizon (2017). Verizon Security Breach Report 2017. Technical report.

Weir, G. and Aßmuth, A. (2017). Strategies for Intrusion Monitoring in Cloud Services. In *Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 1–5, Athens.