Strategies for Intrusion Monitoring in Cloud Services

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

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

Abstract— Effective activity and event monitoring is an essential aspect of digital forensic readiness. Techniques for capturing log and other event data are familiar from conventional networked hosts and transfer directly to the Cloud context. In both contexts, a major concern is the risk that monitoring systems may be targeted and impaired by intruders seeking to conceal their illicit presence and activities. We outline an approach to intrusion monitoring that aims (i) to ensure the credibility of log data and (ii) provide a means of data sharing that supports log reconstruction in the event that one or more logging systems is maliciously impaired.

Keywords-Cloud security; intrusion monitoring; message authentication codes; secret sharing.

I. INTRODUCTION

A news report from a recent computer electronics trade show featured a light bulb with an in-built spy camera. Although the application of this device is the realm of physical security rather than the world of computer, Clouds and networks, we can derive two general lessons from this example technology. Firstly, the purpose of the device is surveillance. Secondly, the device aims for covert operation. These joint concepts of covert surveillance are important in the context of security, whether in the home, on a network or in the Cloud. The primary role for this spying light bulb is surveillance, i.e., in the event of a security incident, to record data that may later have evidential value. 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 covert 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 have initially 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. Thereby, ‘a forensic investigation of digital evidence is commonly employed as a post-event response to a serious information security incident.’ … ‘Forensic readiness is defined as the ability of an organisation to maximise its potential to use digital evidence whilst minimising the costs of an investigation’ [1, p.1].

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 (e.g., [2]).

In the following, Section 2 reviews the characteristics of Cloud services and the facilities available to the customer. Section 3 characterises the attack context, with reference to recognised phases and the likely associated intruder behaviour. In Section 4, 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, both overt and covert, as well as their likely effectiveness as mechanisms for event reconstruction and on-going resilience. Section 5 presents an example monitoring approach that contains specific aspects toward a solution to the forensic readiness problem in the Cloud context. As summarised in Section 6, our proposed approach would generate auditable information that can be used subsequently for digital forensics analysis in a post-hack scenario, within a setting of Cloud Services.

II. CLOUD SERVICES

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

The US National Institute of Standards and Technology (NIST), has provided a detailed account of Cloud Services [3]. This includes a description of typical service models:

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

In the first case, 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. Aside from specific application configuration options, in this arrangement the customer is given no control over the underlying Cloud infrastructure (op. cit., p.2). This level of service extends from simple file storage, through hosted Web sites and database management to specific Web services, including RESTful applications [4], and use of 'containers' [5].

In the second case, the customer is permitted to deploy their own applications on to the service provider's Cloud infrastructure. Customer control extends to configuration and management of these Cloud-hosted applications but, as before, the customer has no facility to control any other aspects of the underlying Cloud infrastructure [3, p.2].

In the third case, the customer has greater scope for software deployment on to the Cloud infrastructure, extending to 'arbitrary software, which can include operating systems and applications' (op. cit.). Still, in this arrangement, the customer's control is limited to the deployed software applications, including operating systems (e.g., virtual machines) and associated networking features (such as software firewalls) [3, p3].

These service models characterise typical Cloud Service Provider (CSP) offerings and the increasing levels of access and software capability, is reflected in increasing levels of cost to the consumer. Notably, 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 the purchased services.

The range of applications and software facilities afforded by Cloud services is extensive, and indications are that many mission-critical services are moving to Cloud implementations as a means of limiting security concerns and assuring greater resilience. The virtual nature of Cloud services also means that system recovery or replacement can be quick, reliable and cost-effective [cf. 6]. Such outsourcing of local 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).

To simplify the categories of Cloud uses, we may broadly differentiate two end-user contexts. In the first, 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 context, 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 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

In general, there are three phases to a successful cyber-attack:

1. reconnaissance and information gathering;
2. infiltration and escalation and, finally;
3. exfiltration, assault and obfuscation.

In phase 1, 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.) [6]. Using this information, the adversary gains access to the system in phase 2 [8].

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 phase 3, the adversary extracts any information from the system that might prove to be useful [9]. 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. 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 [10] or key file signature monitoring [11].

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 [12]. 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., 13]. 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 [14]-[19]. 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 [20].

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 which is introduced in addition to the traditional daemons or agents has to meet at least the following constraints:

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.

V. 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 [21].

The idea is to divide some data D into n pieces $\mathrm{D}_1, \ldots, \mathrm{D}_\mathrm{n}$ in such a way that:

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

(b) the knowledge of only $\mathrm{k} - 1$ 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, $(x_1, y_1), \ldots, (x_k, y_k)$ 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 $i = 1, \ldots, k$. At first, the coefficients $a_1, \ldots, a_{k-1}$ 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 $\mathrm{D}_1 = \mathrm{q}(1)$, $\mathrm{D}_\mathrm{i} = \mathrm{q}(\mathrm{i}), \ldots, \mathrm{D}_\mathrm{n} = \mathrm{q}(\mathrm{n})$. 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 $\mathrm{D} = \mathrm{q}(0)$. From any subset of less or equal $\mathrm{k} - 1$ 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 [21].)

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.

VI. 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 distrusted 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).

Evidently, Cloud service provision has a requirement for robust monitoring that is sufficient to withstand direct assault from an intruder within the host context. Conventional OS monitoring goes some way toward providing the equivalent of a light bulb with an in-built spy camera, but needs to be enhanced with a reliable mechanism for validating and reconstituting log data, such as we have outlined in this paper.

References

[1] R. Rowlingson, "A ten step process for forensic readiness", International Journal of Digital Evidence, vol. 2, no. 3, pp. 1-28, 2004.

[2] K. Reddy, H. S. Venter, and M. S. Olivier, “Using time-driven activity-based costing to manage digital forensic readiness in large organisations”, Information Systems Frontiers, vol. 14, no. 5, pp. 1061-1077, 2012.

[3] 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].

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

[5] L. Richardson and S. Ruby, RESTful web services. O'Reilly Media, Inc., 2008.

[6] 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.

[7] B. P. Rimal, E. Choi and I. Lumb, “A taxonomy and survey of cloud computing systems”, INC, IMS and IDC, pp. 44-51, 2009.

[8] 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].

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

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