Forensic Analysis of Distributed Service Oriented Computing Platforms

Robert Hegarty, Madjid Merabti, Qi Shi, Bob Askwith
School of Computing and Mathematical Sciences
Liverpool John Moores University
R.C.Hegarty@2006.ljmu.ac.uk, [M.Merabti, Q.Shi, B.Askwith] @ljmu.ac.uk

Abstract— **Cloud computing is quickly becoming pervasive. Millions of concurrent users are taking advantage of the flexibility offered by cloud computing platforms. The use of the large scale global storage provided by cloud computing presents a barrier to existing digital forensic techniques which were developed to target single hosts containing a small number of storage devices. By analysing computers to determine if they have been used in the commission of a crime or breach of policy. Various techniques are employed to analyse all aspects of a computer and/or network to determine if malicious activity has occurred. One such technique is signature detection, where signatures from know illicit files are searched for to determine their presence on a computer or storage device. We have identified that the volume and distribution of data in cloud platforms presents a barrier to the application of existing signature detection techniques. The focus of this paper is the development and implementation of a distributed signature detection framework that will enable forensic analysis of cloud storage platforms.**

Keywords; Digital Forensics, Cloud Computing, Signature Detection

I. INTRODUCTION

The distributed storage platforms provided by Cloud computing contain massive quantities of data belonging to many different users [1] [2]. Digital Forensics investigation aim to identify known target files stored in computer systems by identifying their signatures. Current digital forensic signature detection techniques were not developed to target distributed environments and therefore do not possess the capability to process data and signatures at cloud scale. Cloud scale systems contain exabytes of distributed data making analysis by a single computer infeasible. Therefore a distributed approach is required. Signature detection is the process of comparing a set of target signatures against signatures generated from a set of files to determine if any matches are present; it entails analysis of vast repositories of data to detect the presence of known target files [3]. For a distributed analysis process to be feasible the target signatures must be distributed for comparison to be carried out. In previous work [4] we developed a technique to create signatures of a suitable length for use in the signature detection process by selecting the first n bits of an MD5 hash value [5] based on the number of signatures in the target signature set and the number of files being analysed. This allowed us to reduce the overall amount of data required to store the target signatures and in turn reduce the network requirements associated with their distribution. This paper proposes a framework to distribute the reduced length signatures from our previous work and use them to identify target files stored in a distributed storage network. The main contribution of our work is enabling signature detection in large scale distributed storage platforms. The remainder of this paper is structured as follows; Section 2 provides background on digital forensics, cloud computing and distributed storage, the understanding of these areas was vital to the development of our distributed signature detection framework proposed in Section 4. Section 3 provides details of existing techniques. Section 4 contains details of our framework and implementation. In Section 5 we analyse the results of our system and its effectiveness in carrying out scalable distributed signature detection. Section 6 provides a conclusion and details of our future work.

II. BACKGROUND

This section provides the background necessary to understand how a distributed signature detection process can be developed within a cloud platform. Subsection A provides details of existing digital forensic techniques and explains why they are unsuitable for analysing distributed storage platforms. subsection B provides details of the infrastructure found in cloud storage platforms and enables an understanding of how a distributed signature detection process can be deployed in a cloud.

A. *Digital Forensics*

Digital Forensics is the formalised gathering of evidence from digital platforms [6]. This paper is concerned with the automated detection of known target files using a distributed signature detection technique. Signature detection is the comparison of unique features of a piece of data to determine if a match can be found it is typically carried out in digital forensics to identify target files on a storage device [3]. The process is as follows: a storage device is imaged [6] and hash values created for each file in the image. The hash values are then compared with a set of target hash values to determine if any matches can be found. The key property of hash functions that enable this process is that they produce hash values which are practically unique to each file the hash function was performed on. While cloud computing is efficient [7] and flexible it introduces barriers to current digital forensic analysis

ISBN: 978-1-902560-25-0

techniques, which typically rely on having physical access to a data storage device in order to image (create a bit by bit copy of the device) [6] and search the device for known illicit files. Storage device access is restricted in cloud computing as the distributed storage often spans multiple data centres worldwide and is used to store multiple concurrent users' data [8]. Aside from the distribution of data it is infeasible that single host could carry out signature detection on the volume of data found in a cloud platform in a timely manner.

B. Cloud Computing and Distributed Storage

Through a combination of distributed computing and virtualisation [9] Cloud computing provides on demand [10] provision of elastically scalable services [2] [11]. The three broad categories of service provided by cloud computing [12] are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS provides users with hardware and software as a service, and its resources (processing power, memory, and storage) are dynamically scalable, enabling adaptation to suit the users changing requirements and providing cost savings over conventional infrastructure procurement [7]. PaaS provides an environment in which developers can create distributed scalable applications using the API's provided by the service provider. SaaS provides end users with applications hosted on a distributed architecture which can be accessed by a web browser. This enables developers to create cross platform applications and for end users to access applications from any computer. All of these models provide storage as a service (SaS) in some form or other, storage is also offered as a service by providers such as Amazon via their S3 service [13] and Dropbox [14] a storage service for end users. Our work aims to perform signature detection on SaS. For the focus of our work an understanding of IaaS is essential as we require access to and control of the infrastructure in order to develop a distributed digital forensics framework within a cloud. IaaS can be further broken down into other services which are accessible to the end user. Computational resources provide the end user with access to virtualised computers within the cloud. The user can specify all aspects of the computer which they wish to provision including operating system, processor specification, quantity of memory, amount of storage etc. Storage is also provided as a service (SaS) this is useful when multiple virtualised computers need access to shared storage or a user requires storage without any computational resources.

The way in which this storage is accessed is of particular relevance to our research as it is SaS which we aim to analyse. The requirement for file access by multiple simultaneous users, found frequently in Cloud computing has lead to the need for a decentralised model to provide the required scalability and performance [15]. Metadata Servers (MDS) separate file metadata from the Object Storage Devices (OSD) used to store files. The OSDs replace block level storage devices by providing a CPU, network interface, local cache and storage combined to create a device capable of making decisions about storage allocation in a decentralised manner rather than relying on a generic server platform to process data storage [15] along with other tasks such as application execution etc which can lead to overloading. Clients communicate with the MDS for tasks such as finding, opening and renaming a file and with the OSD to perform direct file I/O as shown in figure 1. This decoupling has performance benefits as more than 50% of client requests are for metadata operations [16] using separate paths for disk I/O and metadata operations alleviates bottlenecks [8] associated with single centralised storage techniques where the operations are combined on a single path. Typical metadata operations are both frequent and small in size with only small amounts of data being written or modified. As the operations on files/objects are often much larger with large amounts of data being written, modified or read it is logical to handle the metadata operations on a cluster of dedicated MDS's and allow the OSDs to handle file/object storage. By assigning these tasks in such a manner both the MDS's and the OSD's can be tailored to handle these tasks in a scalable manner.

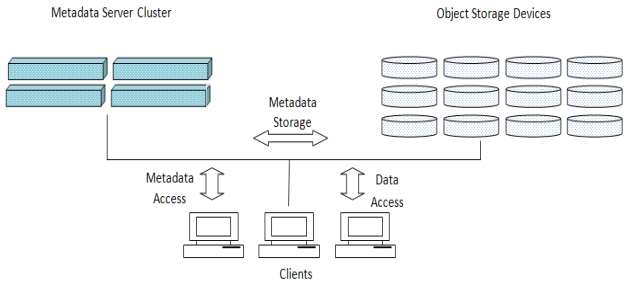

Figure 1. Decoupling of metadata and data

To preserve the integrity of files stored in cloud computing environments integrity checks are carried out and the associated hash value of each file is stored at the MDS as an etag (etags are the term used to describe the MD5 hash values stored in the cloud) this enables checks to be made by the user before and after uploading/downloading a file. It also enables the integrity of files at rest on the OSDs to be periodically checked. If an integrity check fails the file is replaced by one of its redundant copies stored within the cloud. This provides resilience as if any part of the cloud storage platform fails the redundant copies can be duplicated and used to provide failover.

III. RELATED WORK

Various authors have identified the potential to utilise distributed computing platforms to perform forensic analysis or store images [18] for use in digital forensic investigations. In [19] [20] the authors identify the need to apply distributed analysis techniques to the task of analysing computer hard drives. They cite the complexity of investigations and ever increasing storage capacities as barriers to existing standalone techniques. They also identify the features of forensic analysis techniques, which can benefit from informed design choices such as avoiding unnecessary memory-to-memory copies and disk I/O particularly writes [19]. However ultimately they posit that the performance gains alone from well-designed software will not be sufficient and a distributed approach is required. In [19] the acquisition phase of forensic analysis is identified as an area which needs improvement as capturing all possible data

is costly [19]. A system to load entire disk images into main memory, distributed across multiple worker nodes and coordinated by a central control node is proposed in [19]. This allows the disk image to be read into memory once for subsequent analysis by multiple worker nodes reducing the I/O overheads imposed through multiple reads. The author also criticises the current sequential query/response/query model used by current tools for their lack of concurrency. While the paper successfully identifies the requirements for a distributed processing technique it does not implement or test any techniques nor does it address the emerging trend of large-scale distributed storage. A distributed forensic analysis architecture proposed by [21] utilises a database wrapper to enable workstations to request the results of a distributed analysis technique from a parallel machine. The distributed analysis technique is not described and the overall architecture appears to be simplistic. They do not consider the requirements identified by previous work in this field. The Forweb technique proposed and tested in [22] utilises a more efficient block signature search technique Forsigs [23] to analyse blocks from images retrieved from the Internet by a web spider. The approach was developed to analyse image files uploaded and stored in distributed storage platforms accessible via the World Wide Web using a single analysis host to target specific websites. While the technique results in accurate signature detection when searching compressed file types it is not accurate when searching for non-compressed file types and does not scale due to its reliance on a single host for analysis.

IV. Distributed Signature Detection

The signature detection process described in sections 2 and 3 of this paper is not applicable to distributed storage platforms as it requires either the use of a single host which would be overwhelmed in a distributed environment by the scale of data undergoing analysis or the distribution of MD5 signature sets of the target files which may be many hundred of megabytes to many distributed analysis nodes. Distributed storage platforms contain data at a massive scale and provide storage to many concurrent users making imaging of the storage media infeasible. For this reason we require a new approach to performing signature detection at the scale and distribution of data encountered in cloud platforms. We propose and implement a distributed signature detection scheme to overcome the challenges posed by distributed storage. In this section we provide details of our design taking into account the infrastructure found in cloud platforms and our implementation using the APIs available for current platforms.

A. *Design*

Existing cloud platforms share common infrastructure components and provide similar methods for insuring the integrity of files. By targeting the use of these features in Amazon's S3 we have developed a generic model which can be applied to all existing cloud platforms that provide object storage and the majority of future platforms as they will require similar features. As detailed in section 3 cloud computing platforms allow on demand provisioning of compute resources. Our design makes use of this on demand provisioning to analyse the cloud platform by creating a number of components in the cloud as illustrated in figure 2. By following the analysis process we can determine the requirements for each component and describe their functionality. As most investigations will be instigated from outside the cloud under the instruction of the relevant authority an initialiser outside the cloud is required to prescribe the scope and target of the investigation. The first component of our framework the Initialiser fulfils this role by sending a list of target buckets to another component the Forensic Cluster Controller (FCC) along with a file containing the target MD5 hash values. Buckets are the term used in cloud computing to describe directories which belong to a specific user. The Initialiser is responsible for initiating the FCC by booting an image within the cloud. The FCC initialises and queries the third component of our system the Analysis Nodes (ANs) to determine how many files are contained in each target bucket. Each AN responds and the FCC calculates a suitable length signature for use at each AN taking into account the required round one false positive rate, the number of target MD5s and the number of files at the analysis node. The FCC then sends a file containing the appropriate length round one signatures to each of the ANs. Upon receiving the round one signature file each AN retrieves the etags of the bucket it has been tasked with analysing. It does this by either polling the target bucket (effectively the metadata server) for the etags or by analysing the file which it holds containing all the etags from the target bucket. The signatures in the round one signature file are compared with signatures generated from the etags by the AN. Any matches found at the ANs result in the full MD5 hash value of the file found to be a match being sent back to the FCC for the second round search to be carried out to confirm or deny the first round match. When all the ANs have reported their results to the FCC they are terminated by the FCC which is in turn terminated by the Initialiser upon completion of the analysis process. The process is illustrated in figure 2.

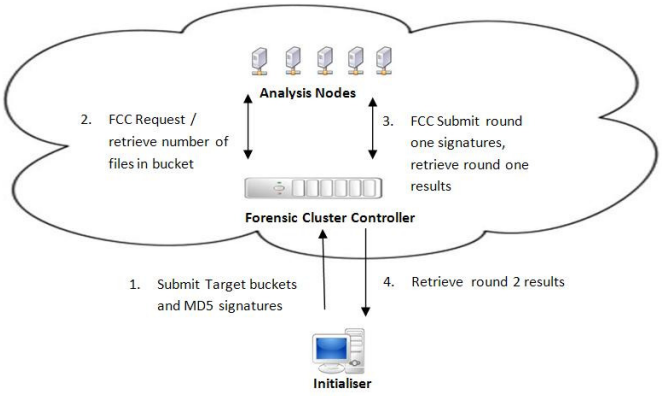

Figure 2. Distributed analysis

We designed our system to carry out forensic analysis in two separate modes. To enable it to be applicable to existing cloud platforms that have not had their infrastructure modified to support forensic analysis. While also illustrating how with small changes to the infrastructure forensic analysis can be supported by cloud computing platforms. The first mode sets up ANs as detailed above and polls the target S3 buckets to retrieve the etags that correspond to the files in each bucket.

This approach allows us to perform analysis on existing platforms without any changes being made to the underlying infrastructure. In the second mode our system emulates a cloud platform where the underlying infrastructure has been modified to support forensic analysis. As metadata servers already hold the etags which we use in the analysis process they could be augmented to support the forensic analysis process. Our system emulates this approach by analysing a list of etags stored at each AN. By supporting both modes of operation we can analyse existing platforms using the first mode of operation and measure the scalability we would achieve if we modified the infrastructure using the second approach.

B. *Implementation*

The implementation of our framework was carried out using Amazon Web Services (AWS) in particular Elastic Cloud Compute (EC2) and Simple Storage Service (S3) [18] as this provides a realistic real world test bed with all the required components for implementation of our system. Using the Boto [24] Python [25] interface to Amazon Web services we were able to access and control the IaaS offered by Amazon. We developed each of the components required by our design using a combination of compute, storage and queue services provided by AWS as described in the design section of this paper.

We used our framework to target a number of buckets in S3 containing various numbers of files using a signature set containing 10,000 signatures. This enabled us to determine what the size of the signature sets generated by our framework for each of the analysis nodes.

V. Analysis and Results

By targeting buckets in S3 which contained 5, 10, 50, 100, 500, 1000, 5000 and 10,000 files with 10000 target signatures and recording the amount of data required for analysis at each analysis node we were able to compare our framework with a theoretical framework which utilises the existing approach of comparing MD5 checksums. The results of our search are comparable in terms of accuracy with zero false positives or false negatives occurring, however as illustrated by the charts below we achieve a significant reducing in the amount of data required to carry out analysis.

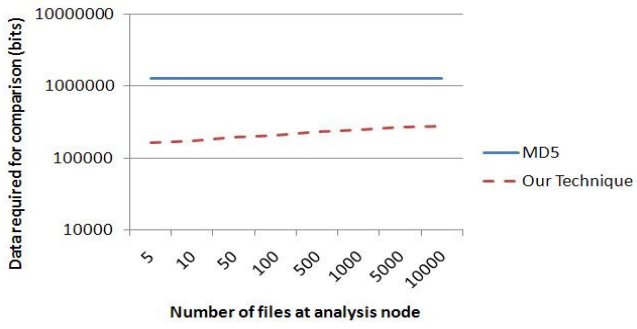

Figure 3. Varying amounts of data required by analysis nodes

It is clear from figure 3 (note the logarithmic scale) that the amount of data required by each analysis node varies significantly and that our technique provides the benefit of identifying how much data is required and generates/distributes the data to the analysis nodes.

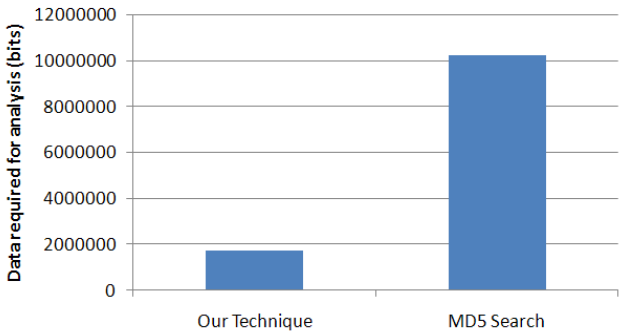

Figure 4. Total amount of data required by all analysis nodes

Taking into account the total amount of data required to analyse the files stored at each of the analysis nodes during our experiment as shown by figure 4 illustrates the benefit of generating and distributing suitable length signatures to each of the analysis nodes rather than using the approach applied in standalone digital forensics techniques of comparing the full length MD5 hash values.

VI. Conclusion and Further Work

Our framework performs distributed signature detection by assigning responsibility for the various system requirements to components within the framework designed to fulfil specific roles. By centralising the task of signature generation and distributing the task of signature detection we provide scalability while at the same time reducing the burden of signature dissemination on the system. By reducing the bandwidth requirements for signature dissemination we alleviate the bottleneck in the system posed by the network and enable signature detection to be carried out on distributed data in a timely manner something not previously feasible with existing techniques. The detection of known files is a key task often carried out in digital forensic investigations of standalone computers. Our work leverages the on demand nature of cloud computing to enable the detection of known files in distributed storage platforms with the same level of accuracy as that of standalone investigations. By utilising the MD5 hash values stored in distributed storage platforms as signatures we adopt the de-facto standard used by current digital forensic investigations to provide equivalent levels of forensic soundness in our approach. While our approach uses MD5 hash values it is equally applicable to all types of hash value to allow for the future use of other hash values.

We are carrying out larger scale testing to monitor the scalability of our framework and paying particular attention to load at the Forensic Cluster Controller. We may implement a method to increase the number of controllers as the number of analysis nodes increases to prevent overloading of the Controller. We also aim to create a modified metadata server

on a private cloud to better measure the real world scalability of our system when signature detection is carried out by a metadata server. We will employ more efficient algorithms at the analysis nodes to enable us to compare the time required for analysis using our technique with existing standalone techniques to determine if our technique can be of benefit to standalone investigations. We also intend to integrate a technique for retrieving and verifying malicious files which are detected into our framework. As the use of cloud computing and online social networking becomes more pervasive investigators will inevitably encounter an increase in the number of crimes involving the distribution of illicit files amongst communities of individuals. Our framework allows investigations to be run in parallel across the distributed data storage platforms on which such data resides and will be extended to analyse the linkage between suspects who commission such crimes.

REFERENCES

[1] W. Huang, J. Liu, B. Abali, and D.K. Panda, "A case for high performance computing with virtual machines," *Proceedings of the 20th annual international conference on Supercomputing - ICS '06*, Cairns, Queensland, Australia: ACM Press, 2006, p. 125.

[2] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," *Grid Computing Environments Workshop, 2008. GCE '08*, Nov. 2008, pp. 1-10.

[3] G.G. Richard III and V. Roussev, "Digital forensics tools: the next generation," *Communications of the ACM*, 2006, p. 75.

[4] R. Hegarty, M. Merabti, Q. Shi, and R. Askwith, "A Signature Detection Scheme for Distributed Storage," London, UK: 6th International Annual Workshop on Digital Forensics & Incident Analysis (WDFIA 2011), 2011.

[5] L. Rivest, R., "RFC 1321 - The MD5 Message Digest Algorithm," 1992.

[6] W.H. Allen, "Computer forensics," *Security & Privacy Magazine, IEEE*, vol. 3, 2005, pp. 59-62.

[7] K. a Delic and M.A. Walker, "Emergence of the Academic Computing Clouds," *ACM Ubiquity*, vol. 9, Aug. 2008.

[8] D. Reilly, C. Wren, and T. Berry, "Cloud computing: Forensic challenges for law enforcement," *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, London, UK: 2010, pp. 1-7.

[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *OSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles*, Bolton Landing, NY, USA: 2003, p. 164.

[10] M. Rappa, "The utility business model and the future of computing services," *IBM Systems Journal*, vol. 43, 2010, pp. 32-42.

[11] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, and I. Stoica, *Above the clouds: A berkeley view of cloud computing*, Citeseer, 2009.

[12] B.P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC*, Washington, DC, USA: IEEE Computer Society, 2009, pp. 44-51.

[13] "Amazon Simple Storage Service (Amazon S3)," *http://aws.amazon.com*.

[14] "Dropbox," *http://www.dropbox.com/*.

[15] S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," *Proceedings of the 7th symposium on Operating systems design and implementation*, 2006, p. 320.

[16] S.A. Weil, K.T. Pollack, S.A. Brandt, and E.L. Miller, "Dynamic Metadata Management for Petabyte-Scale File Systems," *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, 2004.

[17] Y. Zhu and H. Jiang, "HBA : Distributed Metadata Management for Large Cluster-Based Storage Systems," *Management*, vol. 19, 2008, pp. 750-763.

[18] S. Garfinkel, "Commodity grid computing with amazon s3 and ec2," *Usenix*, 2007, pp. 7-13.

[19] R. Golden G. Richard, Vassil, "Next-generation digital forensics," *Communications of the ACM*, vol. 49, 2006, pp. 76 - 80.

[20] V. Roussev and G.G. Richard III, "Breaking the performance wall: The case for distributed digital forensics," *Proceedings of the 2004 digital forensics research workshop (DFRWS 2004)*, DFRWS, 2004, pp. 1-16.

[21] L.M. Liebrock, N. Marrero, D.P. Burton, R. Prine, E. Cornelius, M. Shakamuri, and V. Urias, "A preliminary design for digital forensics analysis of terabyte size data sets," *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*, 2007, p. 190.

[22] J. Haggerty, D. Llewellyn-Jones, and M. Taylor, "FORWEB: file fingerprinting for automated network forensics investigations," *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*, Adelaide, Australia: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 1–6.

[23] J. Haggerty and M. Taylor, "'FORSIGS: Forensic Signature Analysis of Hard Drive Multimedia File Fingerprints'," *FIP TC11 International Information Security Conference*, Sandton, South Africa: 2006.

[24] Boto, "Boto," *http://boto.cloudhackers.com/*.

[25] "Python," *http://www.python.org/*.