Towards a Distributed, Self-organising Approach to Malware Detection in Cloud Computing

Michael R. Watson, Noor-ul-Hassan Shirazi, Angelos K. Marnerides,
Andreas Mauthe, and David Hutchison

School of Computing and Communications, Lancaster University
Lancaster, UK, LA1 4WA
{m.watson1,n.shirazi,a.marnerides2,a.mauthe,d.hutchison}@lancs.ac.uk

Abstract. Cloud computing is an increasingly popular platform for both industry and consumers. The cloud presents a number of unique security issues, such as a high level of distribution and system homogeneity, which require special consideration. In this paper we introduce a resilience architecture consisting of a collection of self-organising resilience managers distributed within the infrastructure of a cloud. More specifically we illustrate the applicability of our proposed architecture under the scenario of malware detection. We describe our multi-layered solution at the hypervisor level of the cloud nodes and consider how malware detection can be distributed to each node.

1 Introduction

Cloud environments are in general made up of a number of physical machines hosting multiple virtual machines (VMs) that provide the resources for the cloud's services. The datacentre has an internal network and is connected through one or more ingress/egress routers to the wider Internet. In order to provide resilience within a cloud environment it is necessary to observe and analyse both system and network behaviour, and to take remedial action in case of any detected anomalies.

Detection in this scenario has to happen at various points throughout the cloud; resilience managers need to exchange information and co-ordinate a reaction to any observed anomalies. Since cloud environments are highly distributed structures with no prescribed hierarchy or fixed configuration resilience managers need to have the ability to flexibly organise themselves taking into account architectural considerations as well as system state. Each resilience manager needs to be a self-organising entity within a larger resilience management framework, which acts autonomously but in a coordinated manner in order to maintain overall system operability.

In this paper we present the architecture of a Cloud Resilience Manager (CRM) and the overall architecture arising from a network of CRMs under the

W. Elmenreich, F. Dressler, and V. Loreto (Eds.): IWSOS 2013, LNCS 8221, pp. 182–185, 2014.

same conceptual autonomic properties followed by previous work [1, 2]. Overall, we discuss the self-organising aspect of each element and how each CRM interacts to form the overall resilience framework.

2 System Architecture

The overall system architecture can be seen in Figure 1 with *A* representing a single hardware node in the cloud. For simplicity only three nodes are shown and the network connections between each node have been omitted. Each node has a hypervisor, a host VM (or dom0 under Xen terminology[3]) and a number of guest VMs. Within the host VM of each node there is a dedicated CRM which comprises one part of the wider detection system. The internal structure of the CRM is shown in more detail by *B* in Figure 1.

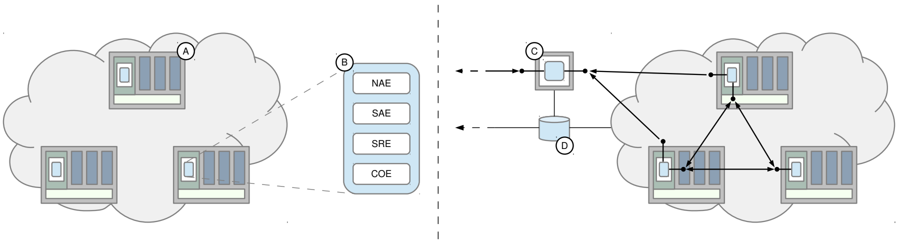

Fig. 1. An overview of the detection system architecture

The software components within *B* are, in order: the Network Analysis Engine (NAE), the System Analysis Engine (SAE), the System Resilience Engine (SRE) and the Coordination and Organisation Engine (COE).

The role of the SAE and NAE components is to perform local malware detection based on the information obtained through introspection of VMs and local network traffic capture respectively. In the NAE observations from lower layers, such as the transport layer, are correlated with observations from higher layers, such as the application layer, in order to attribute anomalies across layers to the same source. In the SAE features such as memory utilisation are extracted from the processes within each VM using introspection[4] and analysed using anomaly detection techniques.

The SRE component is in charge of protection and remediation actions based on the output from the NAE and SAE. The SRE is designed to alleviate the COE of any responsibility regarding system state due to its potentially heavy workload.

Finally, the COE component coordinates and disseminates information between other instances and, in parallel, controls the components within its own node. The COE is required to correlate NAE and SAE outputs by mapping statistical anomalies found in the network to end-system state as reported by the SAE. An example of this is the identification of protocols and ports used by anomalous traffic and the attribution of these to a particular process executing within a guest VM. In this way it is possible to attribute anomalies at disparate locations in the architecture to a single threat.

In addition to node level resilience, the detection system is capable of gathering and analysing data at the network level through the deployment of network CRMs as shown by C in Figure 1. D in the figure represents an ingress/egress router of the cloud; the monitoring system is directly connected to router D and as such can gather features from all traffic passing through it.

Self-organization in a system of CRMs is achieved through the dissemination and exchange of meaningful information with respect to the system and network activities of each VM, as well as with the router(s) that connect the datacenter network to the Internet. In practice, and as depicted in Fig. 1, there are various system/network interfaces that act as information dispatch points[1] in order to allow efficient event dissemination.

3 Resilience Manager Self-organisation

A system of Cloud Resilience Managers (CRMs) is required to be self-organising in order to allow the system to make autonomous decisions regarding end-system and overall cloud resilience. This is achieved through an internal peer-to-peer network in combination with hierarchical interactions between internal and external network interfaces.

3.1 Network Architecture

In Figure 1 the system architecture is shown as consisting of two levels of communication. The interfaces between the CRMs within the cloud correspond to an internal peer-to-peer network where peers can perform push/pull actions with other peers. The interfaces between system level CRMs and those in the network correspond to external interfaces that only allow push actions, from low level CRMs (i.e. system level) to higher level CRMs (i.e. network level).

Due to the hierarchical nature of the system the information sent to C is under a filtered, event-based format resulting from the malware analysis and detection achieved internally by the SAE and NAE components. Thus, the COE in C will only receive meaningful input from a remote COE such that it is able to correlate the VM-level anomalous activity with the traffic it captures on the ingress/egress router(s) (D). For instance if a destination within the cloud infrastructure is locally flagged as suspicious by its CRM, traffic to that destination can be thoroughly analysed by the detection component in C, as in [2].

3.2 Peer Communication

Peer-to-peer communication of data between CRMs enables each individual CRM to make local decisions which are influenced by the activity experienced on remote cloud nodes. This direct communication between peers results in the ability of the CRMs to exchange information with respect to the current health of their respective virtual environments.

The peer network itself is a simple message exchange system, whereby healthy peers advertise their presence in the network. Peers experiencing anomalous behaviour exchange the type of anomaly and pertinent information on how this will affect other peers. The other COEs in the cloud will receive this data and take action by invoking their local SRE.

The benefits of this information exchange versus a centralised system are the ability to notify vulnerable systems in a single exchange, and the ability to reduce the amount of data flowing over communication channels. In a centralised detection system it would be necessary to export all data relating to the state of every physical machine in the cloud to a single point. This scenario puts a higher demand on network links than the solution proposed in this paper. Moreover, a single point of analysis reduces the resilience of the cloud due to a single point of failure. This fact, coupled with the inherently distributed nature of clouds indicates that self-organisation is a better fit architecturally.

4 Conclusion

Cloud environments present unique challenges in terms of security and resilience. These challenges need to be confronted through the synergistic analysis of both system and network-level properties in order to more effectively utilise available information. In this paper we have proposed a solution to these challenges by introducing the concept of a Cloud Resilience Manager (CRM) which combines self-organisation with a distributed approach to detection.

Acknowledgments. The authors would like to thank Fadi El-Moussa and Ben Azvine of BT Research for their contribution to this work. This work is sponsored by the IU-ATC (EPSRC grant number EP/J016675/1) and the EU FP7 SECCRIT project (grant agreement no. 312758).

References

[1] Marnerides, A.K., Pezaros, D.P., Hutchison, D.: Detection and mitigation of abnormal traffic behaviour in autonomic networked environments. In: Proceedings of ACM SIGCOMM CoNEXT Conference 2008 (2008)

[2] Marnerides, A., Pezaros, D., Hutchison, D.: Autonomic diagnosis of Anomalous network traffic. In: Proceedings of IEEE WoWMoM 2010 (2010)

[3] Citrix Systems, Inc., Xen, `http://www.xen.org/`

[4] Payne, B.D.: LibVMI,
`http://code.google.com/p/vmitools/wiki/LibVMIIntroduction`