A Multi-Level Resilience Framework for Unified Networked Environments

Angelos K. Marnerides*†, Akshay Bhandari ‡, Hema Murthy‡ and Andreas U. Mauthe†
*School of Computing & Mathematical Sciences, Liverpool John Moores University, Liverpool, UK
a.marnerides@ljmu.ac.uk
†InfoLab21, School of Computing & Communications, Lancaster University, Lancaster, UK
a.marnerides2,a.mauthe@lancaster.ac.uk
‡Department of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai, India
akshayb,hema@cse.iitm.ac.in

***Abstract*—Networked infrastructures underpin most social and economical interactions nowadays and have become an integral part of the critical infrastructure. Thus, it is crucial that heterogeneous networked environments provide adequate resilience in order to satisfy the quality requirements of the user. In order to achieve this, a coordinated approach to confront any challenges is required. However, there is additional complexity since challenges manifest themselves under different circumstances in the various infrastructure components. The objective of this paper is to present a multi-level resilience approach that goes beyond the traditional monolithic resilience schemes that focus mainly on one infrastructure component. The proposed framework considers four main aspects, i.e. users, application, network and system. The latter three are part of the technical infrastructure while the former profiles the service user. Under two selected scenarios this paper illustrates how an integrated approach coordinating knowledge from the different infrastructure elements allows a more effective detection of challenges and facilitates the use of autonomic principles employed during the remediation against challenges.**

***Index Terms*—Resilience, Autonomic Networks, Network Architectures, Anomaly Detection, Security**

I. Introduction

Computer networks constitute the backbone of today's information society by providing connectivity between people as well as ICT (Information Communication Technology) systems. Consequently, they are increasingly mission-critical, especially when used as part of always-on services and applications (e.g., Web-services, Internet Television, Cloud applications, etc.), domain specific safety-critical services (e.g., Air Traffic Control (ATC) networks), critical management services for operators (e.g., Utility networks), and critical real-time financial services (e.g., stock-market systems). The security and resilience of such infrastructures is therefore paramount but at the same time becomes increasingly difficult to achieve.

Hence, the development of resilience mechanisms has to be a prime objective within the design and engineering process of any system or network [1]. However, in the past, availability was the main concern in the design and operation of computer networks [1] and less emphasis was placed on resilience aspects. Moreover, within the actual deployment of networks and ICT systems resilience aspects have also often been treated as add-on and resilience mechanisms have been implemented without reference to a generic resilience framework [1]. Trying to increase system resilience later by deploying such a generic resilience framework leads to monolithic solutions that mainly consider a part or a particular communication layer only. Thus, they usually focuses on a particular resilience subdomain (e.g., security [2], or survivability [3]) and do not look at the overall system resilience. More advanced resilient schemes that propose cross-layering methods tend to neglect higher-layer features that express the explicit requirements and characteristics of service users. Hence, their formulation results in one-dimensional performance-oriented solutions that strictly focus on traditional network performance metrics (e.g. throughput, delay, jitter) but avoid mapping these metrics onto the overall end-user QoE and QoS. Therefore, such approaches lead to what we consider "single-level" approaches.

In this paper we first present a *multi-level* resilience framework that allows the construction of case-specific resilience architectures that consider the various infrastructure levels and further allowing the construction of user related metadata to better control and identify challenges. User-specific requirements are considered in order to ensure that QoE objectives related to a given resilience strategy are met. Due to the *multi-level* persona of this framework it overcomes the drawbacks of the traditional monolithic, single system approaches as currently employed in some ICT infrastructure. This is achieved by the joint analysis of challenge indicators and co-ordinated detection actions that also help to coordinate the remediation process at the different system elements.

The remainder of this paper is structured as follows: Section II is dedicated at presenting the concepts behind our resilience framework and Section III illustrates the practical aspect of our framework within two case studies. Finally Section IV concludes and summarises this work.

II. Multi-Level (ML) Resilience Framework

As evidenced by Fig. 1, the most important functional block within our design is the adequate representation of a user of an infrastructure. Thus, we aim at describing a user based on three levels of observation; the application/service, network and system levels. Our design argues that measurements of features related with any of these three levels is feasible under the assumption that a given resilience architecture that

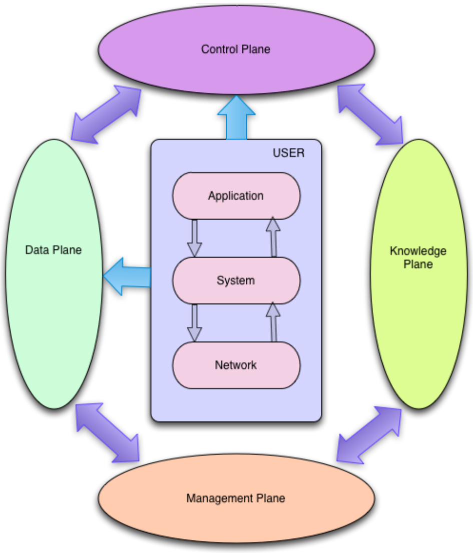

Fig. 1. Conceptual representation of Multi-Level Resilience

follows our framework will be deployed under standardized monitoring and measurement methods (e.g. SNMP, NetFlow, syslog etc.).

A. *Self-awareness & self-defence*

All gathered information from all the three different levels will be pre-processed at the management plane and further analysed on the knowledge plane. Fig 2 provides a visual example of how initially a client is meta-represented by the three different levels within the knowledge plane. The exemplar profiling case of Fig 2 illustrates the scenario of a ramp-up behaviour in all the three levels of observation due to the byte consumption caused by a particular application/service.

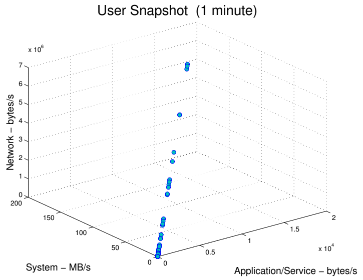

Fig. 2. An example of representing a single user based on the three levels of system, network and application/service.

Given an initial user (or group of users) profiling, the knowledge plane will enforce a dedicated component within its internal structure to perform a statistical characterisation of a user's activities within the observational timeframe. Apart from developing a user-specific profile, this statistical characterisation will also be aggregated for all the users and further correlated with the monitoring and measurement components of the control and the data plane. Hence, an overall characterisation of the environment is achieved and self-awareness is ensured.

Naturally, the overall characterisation complies with a particular mathematical model which is in a position to determine the levels of normality, thus detecting abnormal characteristics in real-time. Under the scenario of detecting an anomalous pattern, components within the management plane will be in charge of informing the knowledge plane. Consequently, the knowledge plane will update the overall statistical characterization of its profiling on a set of users and further trigger fine-grained analysis in order to diagnose the exact cause of the anomaly. Given the outcome of this fine-grained analysis, the knowledge plane will inform the management plane in order to trigger remediation techniques and dimension the environment resources accordingly. Thus, the property of self-defence is accommodated.

B. *Self-management & self-optimization*

Remediation of challenges is resulted after a coordinated act by the management and knowledge planes. In particular, the management plane is the actual co-ordinator at the onset of an event. Thus, the decision regarding the type of anomaly produced by the knowledge plane is sent over to the management plane. Subsequently, the management plane initiates a policies component that holds all the rules regarding the optimisation and management procedures that need to be taken within the ICT environment.

According to the rules provided by the policies component, immediate policies regarding traffic engineering actions (e.g. refined routing decisions, blocking) will be triggered and further received by the control plane. Eventually, the internal mechanisms of the control plane will re-configure the associated settings on the hardware (e.g. routers, sensors) and subsequently update the forwarding schemes within the data plane. Given all the remediation operations described, the properties of self-management and self-optimization for the ICT environment are empowered by our generic multi-level resilience framework.

III. ML Resilience Framework : In practise

A. *Case Study 1: ML Resilience over the Cloud for Malware Detection*

Based on the generic ML resilience framework we have derived a prototype resilience architecture explicitly for the identification and detection of malware over cloud environments where its detailed description can be found in [4]. The overall architecture of our approach can be seen in Figure 3 where for simplicity only three nodes are shown and the network connections between nodes are omitted. Each node has a hypervisor, a host Virtual Machine (VM) and a number of guest VMs. Within the host VM of each node there is a dedicated Cloud Resilience Manager (CRM) which comprises one part of the wider detection system. The software components within the CRM are the Network Analysis Engine (NAE), the System Analysis Engine (SAE), the System Resilience Engine (SRE) and the Coordination and Organisation Engine (COE). Nevertheless, the case study of this section is explicitly addressing the operations of the

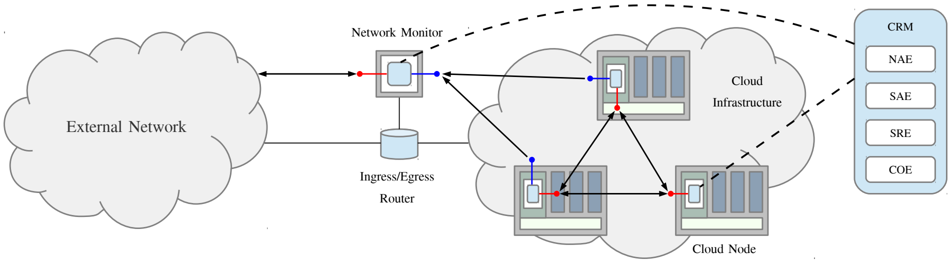

Fig. 3. Resilience Architecture over a Cloud Scenario for Malware Detection

NAE and SAE engines using the example of characterising and detecting the Kelihos malware [5].

Both SAE and NAE are composed by mechanisms that automate all the processes referring to the pre-processing of either network or system data. Their pre-processing outputs are subsequently fed as the primitive input for the analysis algorithms which are also integrated and they employ Principal Component Analysis (PCA) [6] and statistical timeseries on the jointly gathered system and network features. Hence the produced outputs are also considered as important aiding elements towards the establishment of the *self-awareness* and *self-defence* properties where the former enables the adequate characterisation of the three levels of *network*, *system* and *application/service* and the latter achieves malware detection.

In our experiments, we aimed at assessing the fundamental property of VM/service "live" migration of an HTTP server as initiated in todays' cloud environments and further investigate on how malware can be detected in such a scenario within a controlled experimental testbed. The experiment lasted for 20 minutes and the Kelihos malware strain, *Trojan.Kelihos-5*, was injected on the 9^{th} minute in one of the HTTP servers whereas on the 10^{th} minute the infected VM was migrated to the second "clean" physical host as manually commanded by the management host. Throughout the whole experiment there was the consistent aggreagted monitoring of system-related features (e.g. counts of processes) and network packets for all VMs on both physical hosts from the hypervisor level using custom monitoring scripts embedded in the NAE and SAE. Finally, the aggregated system and network features where subsequently fed to our implemented PCA algorithm in order to firstly chartacterize the joint dataset and further pinpoint possible anomalous characteristics.

1) **Case Study 1: Results***:* As evidenced by Fig. 4, each joint dataset is divided into 3-second bins, and each bin is converted into a feature vector per each VM node. The combined feature vector was submitted to PCA to obtain the *k-subspace* which corresponds to the normal behaviour of the traffic, and spans from a principal component pc_1, through pc_k, whereas the remaining subspace with the less significant principal components (i.e, pc_{k+1} through pc_m) maps the anomalous behaviour with respect to the variance of the dataset. Subsequently, we compute a distance metric that describes the magnitude of the projection of the original data points into the anomalous subspace to quantify their malicious behaviour which we use to produce the anomaly score graph (ASG) in Fig. 4. In practise, this plot that is generated by the NAE is a time-series representation which summarizes the anomalous score of each bin in the trace and thus indicates the level of how anomalous is each tested timebin with respect to the other measurement bins. Overall, the PCA performed extremely well and was able to show a sharp increase on the ASG plot as demonstrated by Fig. 4 as soon as the Kelihos malware was injected ($\approx 160^{th}$ bin in Fig. 4). Moreover, the ASG plot also shows that the PCA algorithm could also identify anomalous activity after the VM migration performed right after the 200^{th} time bin.

B. Case Study 2: Multi-level Resilience in Access Networks for Characterization & Detection of Systematic Downloads

Given the properties of the ML resilience framework we have derived timeseries analysis formulations in order to adequately detect systematic downloads that have become commonplace and consequently lead academic institutional campus networks being blacklisted.

In this work, the number of requests per second made to a specific publisher as obtained from the proxy logs captured at a proxy server on the Indian Institute of Technology Madras campus is used to model the time series. In order to build robust models, the data is obtained for 106 such publishers where each one is represented by a different timeseries since license agreements with each is likely to be different. In particular we have considered the number of downloaded files, the files downloaded at random intervals, the file sizes and the random ordering of files of different sizes as our data features. We formed the timeseries by computing the number of requests at different polling intervals varying from 5-30 seconds in 5 second bins. The time-dependent AR process was modelled with a framelength of 3 minutes and we further computed the AR process roots.

1) **Case Study 2: Results***:* The type and order of our model is obtained by first considering the autocorrelation function (ACF) of the time series. It was observed that the ACF of the differenced timeseries was stationary and could be modeled as an Auto Regressive (AR) process

$$\Delta(x_n) \quad = y_n = \quad x_n - x_{n-1} \tag{1}$$

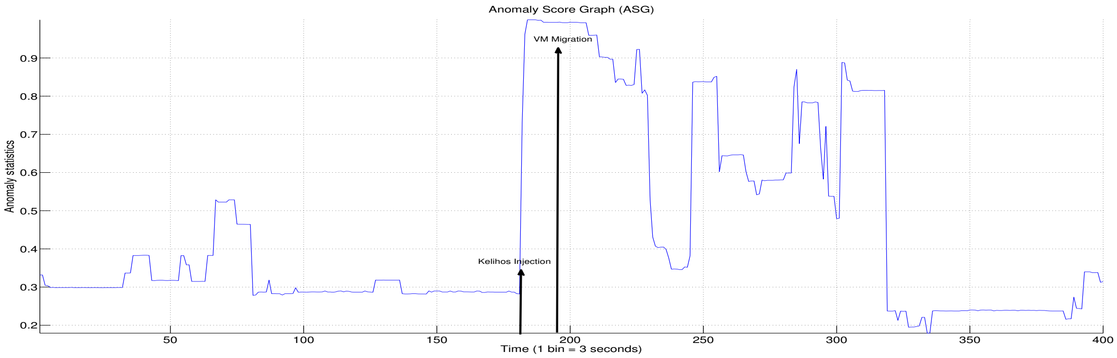

Fig. 4. Output of the resulting PCA-based anomaly detection as jointly performed on network and system data that were monitored by the NAE and SAE.

$$y_n = \sum_{k=1}^{p} a_k y_{n-k} + e_n \tag{2}$$

where e_n is the prediction error assumed to be a generated by a white noise process. In parallel, the order of the model is estimated again from the ACF of the differenced series as shown in Figure 6. The differencing operation aims to address the non-anomalous structural breaks disclosed within network data where average statistics can vary depending upon the downloading time (prime versus nonprime time).

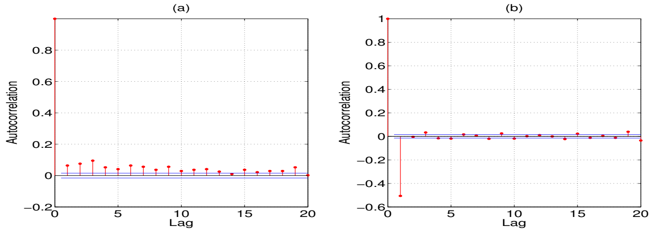

Fig. 5. Autocorrelation function of differenced data before (left) and after differencing (right) in order to estimate the order of the model regarding a publisher(s) timeseries.

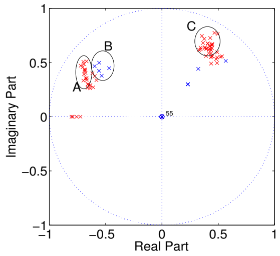

Fig. 6. Roots of the time-varying AR model during periods of both systematic downloads (clusters A,B) and normal downloads (cluster C)

We have identified a specific root that characterised the anomaly and was obtained by using ground truth data (i.e. the location of the region when systematic downloads are in progress), and feature selection was performed. The AR process roots are shown in Figure 5 where different colours are associated with the roots corresponding to that of systematic download and normal traffic. It is evidenced that the roots associated to that of systematic downloads (marked with A and C) are closer to the unit circle and belong to a different cluster compared to a normal download (marked with B).

IV. Conclusions

Today's networked ICT environments are increasingly challenged by misuse and security issues manifested at different levels of the system architecture and are so far dealt with independently. This paper introduces a multi-level resilience framework in which the resilience activities (such as anomaly detection) are co-ordinated in order to provide better and early defence and awareness regarding threats and challenges. This paper illustrates that architectures that comply with the requirements derived from the generic multi-level resilience framework can adequately relate several types of information with respect to application, system and network-specific characteristics. Further, we show how mechanisms that confront particular challenges at the different levels at which they are likely to manifest themselves can be co-ordinated and hence produce a better result. This is demonstrated through case studies focusing on the central aspects of analysis and aggregation of heterogeneous types of information.

Acknowledgments

The authors would like to thank the UK EPSRC and DST India funded IU-ATC research project, the EU FP7 SECCRIT project and the EPSRC Situation-Aware Information Infrastructure project for supporting this work.

References

[1] Smith, P., Hutchison, D., Sterbenz, J., P., G., Scholler, M., Fessi, A., Karaliopoulos, M., Lac, C., Plattner, B., *Network Resilience: A Systematic Approach*, in IEEE Communications Magazine, Vol. 49, I: 7 , pp 88-97, July 2011

[2] 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, Student Workshop, 2008

[3] Sterbenz, J., P., G., et. al., *Resilience and Survivability in Communication Networks: Strategies, Principles and Survey of Disciplines*, in Elsevier Computer Networks (COMNET), Special Issue on Resilient and Survivable Networks, vol. 54, no. 8, June 2010

[4] Watson, M., Shirazi, N., Marnerides, A., K., Mauthe, A., Hutchison, D., Towards a Distributed, Self-Organizing Approach to Malware Detection in Cloud Computing , in 7th IFIP International Workshop on Self-Organizing Systems, IFIP/IFISC IWSOS 2013, May 2013

[5] Garnaeva, M. "Kelihos/Hlux Botnet Returns with New Techniques." Securelist, http://www.securelist.com/en/blog/655/Kelihos_Hlux_botnet_returns_with_new_techniques.

[6] Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E. D., Taft, N., Structural analysis of network traffic flows. in ACM SIGMETRICS Perform. Eval. Rev. 32, 1, June 2004