Assessing the Impact of Intra-Cloud Live Migration on Anomaly Detection

Noor-ul-hassan Shirazi, Steven Simpson, Angelos K. Marnerides, Michael Watson,
Andreas Mauthe and David Hutchison
InfoLab21, School of Computing and Communications
Lancaster University,UK
Email: {n.shirazi, s.simpson, a.marnerides2, m.watson1, a.mauthe, d.hutchison}@lancaster.ac.uk

***Abstract*—Virtualized cloud environments have emerged as a necessity within modern unified ICT infrastructures and have established themselves as a reliable backbone for numerous always-on services. 'Live' intra-cloud virtual-machine (VM) migration is a widely used technique for efficient resource management employed within modern cloud infrastructures. Despite the benefits of such functionality, there are still several security issues which have not yet been thoroughly assessed and quantified. We investigate the impact of live virtual-machine migration on state-of-the-art anomaly detection (AD) techniques (namely PCA and K-means), by evaluating live migration under various attack types and intensities. We find that the performance for both detectors degrades as shown by their Receiver Operating Characteristics (ROC) curves when intra-cloud live migration is initiated while VMs are under a netscan (NS) or a denial-of-service (DoS) attack.**

***Keywords*—Cloud computing, anomaly detection, live VM migration.**

I. Introduction

Cloud environments have evolved as the critical backbone for many ICT infrastructures due to their elasticity and resource transparency. As reflected in a recent report by the European Network and Information Security Agency (ENISA), cloud environments are becoming increasingly mission-critical [1]. Since they provide always-on services for many everyday applications (e.g. IPTV), safety critical operations (e.g., Air Traffic Control networks), critical manufacturing services (e.g., utility networks and industrial control systems), and critical real-time services (e.g., transportation and surveillance systems) [2]. Therefore, the ability of such cloud environments to remain operational in the face of anomalous activities becomes paramount.

Modern virtualised cloud environments support migration of services and virtual machines (VMs) to different physical nodes, and exploit the consequent elasticity and resource transparency for dynamic resource management. In particular, in contrast to cold migration, live migration allows a service or VM to move while retaining its network identity and connections, and without having to be powered off, by transferring its active memory and execution state. This makes live migration essential functionality for effective on-line resource management, allowing the workload to be balanced across physical nodes without major disruption to users, and is performed by the majority of cloud operators (such as VMware VSphere [3]).

While migration is a key feature of cloud environments, it introduces novel security and resilience challenges. For instance, an anomaly detector applied to network traffic visible at the cloud-infrastructure level[1] could be misled by the effects of migration on that traffic in two ways. First, legitimate migration could be misidentified as an anomaly (a false positive indication). Second, migration could occur simultaneously with a genuine challenge, and thus mask its detection (a false negative). Overall, despite the plethora of signature-based and anomaly-based detection solutions for a number of computer networks (e.g., [4], [5]), there has not yet been a thorough analysis on the impact of VM migration on state-of-the-art AD solutions. We aim to quantify this impact within an intra-cloud scenario where migration occurs between physical nodes in the same cloud environment.

We measured the performance of two AD techniques (Principal Component Analysis (PCA) and K-means clustering) that have proven to be effective in past and current literature [6], [7], [8]. These techniques were evaluated in a controlled cloud testbed in which an attack (either denial-of-service (DoS) or netscan) and a migration occur either simultaneously or separately, yielding metrics such as detection performance (i.e., true-positive rate; TPR) A broad observation is that the presence of migration affects the ability of both techniques to detect netscan more than DoS These outcomes, empower our argument that under certain attack-type and migration conditions, the number of attacks that are missed and false alarms generated by these techniques could render them unreliable and unusable respectively. Consequently, we argue that widely used AD techniques such as PCA and K-means clustering are directly affected by the live-migration aspect and, therefore, future designs of cloud-oriented anomaly detection components should consider this factor. To best of our knowledge, our work is the first to investigate the effect of VM migration on anomaly detection techniques.

The rest of the paper is organised as follows: Section II describes our experimental set-up. Section III discusses the experimental method, and the properties of the AD techniques used. Section IV describes the outcomes of our analysis and discusses the obtained results, while section V summarises and concludes the paper.

II. Experimental set-up

We established a testbed in which live Local Area Network (LAN) VM migration can take place while experiencing both

[1]We refer the reader to a SECCRIT whitepaper which presents an architectural model as a basis for the bringing critical-infrastructure (CI) services into cloud environments: https://www.seccrit.eu/whitepaper

normal and abnormal traffic conditions. Two hosts serve as nodes for running multiple VMs. Another host acts as a controller to initiate migrations, and also to generate background traffic. A fourth host generates attack traffic. All are connected to a LAN, as shown in Fig. 1.

Each physical node runs Kernel-based Virtual Machine (KVM)[2] as virtualization infrastructure, and the Quick EMUlator (QEMU[3]) provides hardware emulation. Migration is achieved with *libvirt*[4]. All VMs on a node are connected to a virtual bridge interface `virbr0`, so their own interfaces appear to be part of the LAN.

For the experiments presented here, each VM runs Apache HTTPd. The client host runs custom scripts to initiate random HTTP requests from the VMs. The 'challenger' host runs custom attack scripts to generate attack traffic directed towards the VMs' address range for a selected attack type and intensity (i.e., the volume of traffic it generates). Tcpdump[5] is used to simultaneously collect packet traces from the two virtual bridge interfaces, one in each physical node, and so these traces represent aggregate traffic to/from all VMs on a node.

This set-up allows us to run experiments in which the legitimate traffic of several web servers is continuously emulated, while anomalous traffic is emulated by overlaying the legitimate traffic with attack traffic from the attack scripts that run during part of the experiment. Independently, one of the VMs running a webserver can be migrated live between the nodes during a period of either normal or anomalous traffic. Subsequently, traces obtained at the virtual bridges can be fed into anomaly detectors to observe their reactions to normal/anomalous traffic with and without migration. Because we have control over when migrations and anomalies occur, we can confidently label our obtained traces with ground truth about both conditions, and therefore assess the impact of migration on anomaly detection that takes such traces as input.

In order to obtain a coherent view in every experiment we run, each monitored packet trace on every experimental iteration was summarised as depicted in Table I. As shown, every trace was described by the type of background traffic it captures, the anomaly/attack type, the attack intensity, migration overlap (whether migration happens during the normal or anomalous period), and migration direction (whether a VM is leaving or arriving at the node generating the trace). Section III explains characterization in more detail.

III. Evaluation method

To evaluate an anomaly detection technique in the face of migration, we first perform several experimental runs, where each yields a pair of packet traces which are labelled with the ground truth regarding the presence of attack traffic or migration in the trace. In each 10 minute run, background traffic occurs continuously at a fixed rate, and hence appears throughout a trace. At the first 5 minutes in the first run, an attack script starts, hence its traffic appears in each trace

[2]Kernel-based Virtual Machine: http://www.linux-kvm.org/
[3]Quick EMUlator: http://www.qemu.org/
[4]The virtualization API: http://libvirt.org/libvirt2
[5]Tcpdump: http://www.tcpdump.org/

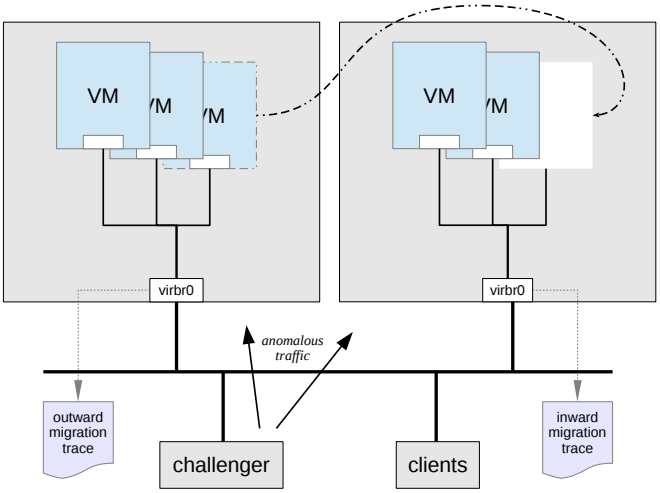


Fig. 1: Experimental test-bed set-up

from the midpoint. At either 2.5 minutes or 7.5 minutes, a migration of one of the VMs is initiated. A run can therefore be characterized by the attack type and intensity, and whether the migration occurs during the attack or during the normal period (i.e. 'migration overlap'). Each trace from a run can be further characterized by whether the node it was taken from experienced an outward (MDout) or inward (MDin) migration of the VM.

In parallel, the two anomaly types are denoted as NS for netscan and DoS for the denial-of-service where each type is employed under a high (AH) or low intensity (AL). DoS targets a VM that does not migrate. The Migration overlap is denoted by either NM (the migration occurred during the first half of the run, during the normal period), or AM (during the anomalous period). The fixed background traffic is denoted by BC0, to distinguish it from future runs with alternative background characteristics. BC0 involves five VMs running identical HTTP servers. Three run on one physical host, and two on the other. A host external to the VM infrastructure runs HTTP clients repeatedly connecting to each VM, two per VM. We performed one run for each of the 8 possible combinations of these characterizations, yielding 16 traces.

Each packet trace is filtered to eliminate the related management traffic between the VM host nodes. It is subsequently divided into 1-second bins, and each bin is converted into a feature vector, also labelled with ground truth. Tables of labelled feature vectors from related traces, i.e., the four in which the same attack type and intensity was applied (with NM/AM and MDin/MDout varying), are combined to form a dataset representing 40min of experimentation. Hence, our 16 traces yield 4 combined datasets. The processing from experiments to combine data sets is shown in Fig. 2.

Each examined detection technique is then applied to each dataset, by submitting them together to an evaluation process, as depicted by Fig. 3. The feature vectors from the dataset are submitted through a configured anomaly-detection engine, yielding an anomaly time series, a series of anomaly scores, one for each vector, indicating how anomalous the detector finds that vector with respect to the others. The anomaly scores

TABLE I: Summary of characterization

Background characterization	**Anomaly characterization**				**Experiment characterization**				**Detector characterization**	
Type	***Type***		***Intensity***		***Migration overlap***		***Migration direction***		***Type***	
BC0	**NS**	**DoS**	**AH**	**AL**	**NM**	**AM**	**MDin**	**MDout**	**PCA**	**KM**
5 HTTP servers; 2 clients each	Netscan	Denial-of-service	High	Low	Normal period	Anomalous period	Inward	Outward	Principal Component Analysis; unsupervised	K-Means; supervised

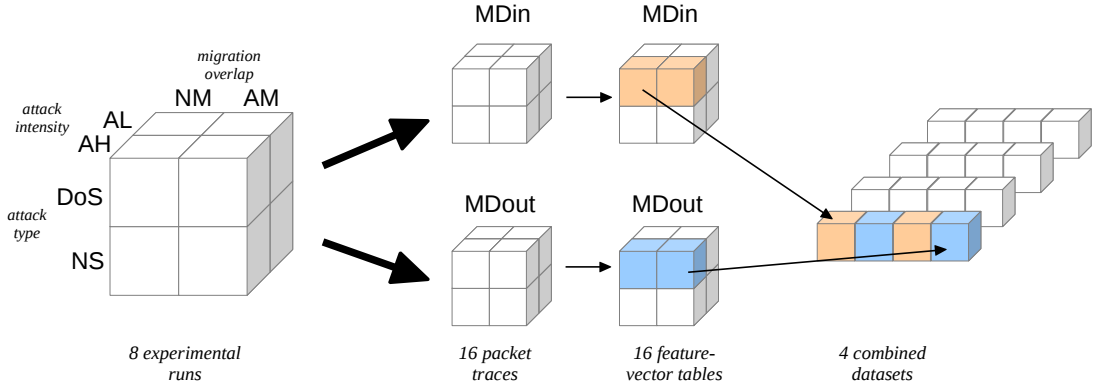


Fig. 2: Processing of data

are recombined with the corresponding ground truth, to yield a table of 3-tuples including anomaly score (AS), boolean anomaly ground truth (GT) and boolean migration ground truth (i.e. 'instantaneous migration intensity', IMI). The same dataset is also stripped of rows with IMI set, and submitted through the AD engine to yield a second anomaly-score table, which is similarly recombined with ground truth columns. Each labelled anomaly-score table then generates a Receiver Operating Characteristics (ROC) curve[6], and the two can be compared to derive an evaluation of the detection technique against the scenario represented by the dataset. In order to aid the visualization process of the dataset, we also present some of the anomaly-score tables visually as anomaly-score graphs (ASGs)[7].

C and the libpcap API are used to derive features from the packet traces. MATLAB is used to implement the anomaly detectors and the decision analyser.

In order to capture the dynamics of varying attack types, we extracted both volume-based features (e.g., count of bytes and packets) and distribution-based features (computed as the Shannon entropy of all values observed in the bin, as used in many seminal pieces of work [6]). We provide the following set of features computed for each bin:

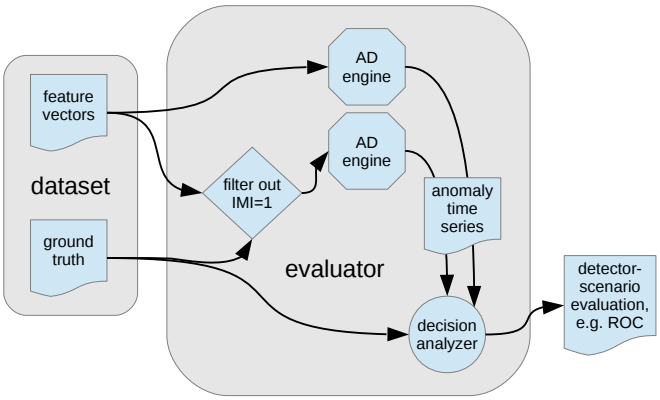
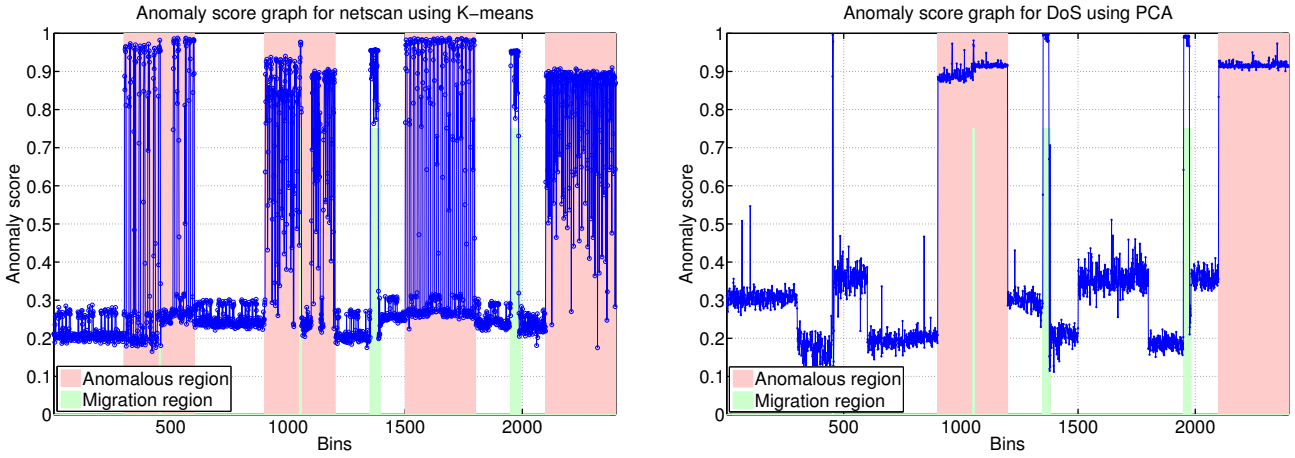


Fig. 3: Evaluation process

- Number of packets
- Number of bytes
- Number of active flows in each bin
- Entropy of source IP address distribution
- Entropy of destination IP address distribution
- Entropy of source port distribution
- Entropy of destination port distribution
- Entropy of packet size distribution

[6]A ROC curve is a plot of true-positive rate (TPR) against false-positive rate (FPR) for a range of thresholds. Points towards the bottom left correspond to high thresholds (or low sensitivity), and the top right to low thresholds (or high sensitivity). Better performance is indicated by curves that tend to occupy the top left, as these imply that sensitivity can be decreased to eliminate more FPs without degrading the TPR.

[7]An ASG displays a time series of outputs (anomaly scores) from a detector. Our particular ASGs are annotated with the periods during which attacks (GT $=$ 1 in pink) and migration (IMI $=$ 1 in pale green) occur. An ideal detector would show high scores only while GT $=$ 1, independently of IMI.



(a) ASG for netscan using K-means

(b) ASG for DoS using PCA

Fig. 4: Anomaly score graphs using K-means & PCA

A. *Principal Component Analysis*

As presented in [6], [9], [10], the PCA technique is not only used to reduce the dimensionality for a given dataset but also to separate the normal data from anomalous, based on the scores obtained by the newly computed principal components (PC). In practice, the produced PCs provide a transformation of the original data points to a new set of axes. Overall, by applying PCA to our combined dataset, the set X yields a set of m principal components $\{pc_i\}_{i=1}^{m}$. Assuming that data in X is zero-mean, the first principal component pc_1 is the vector that points in the direction of maximum variance in X and computed as follows:

$$pc_1 = arg \max_{\|pc\|=1} \| Xpc \| \tag{1}$$

The PCA employs the Singular Value Decomposition (SVD) method that obtains the k-subspace corresponding to the normal behaviour of the traffic, and spans from pc_1, through pc_k, whereas the remaining subspace (i.e, pc_{k+1} through pc_m) maps the anomalous behaviour with respect to the variance of the dataset. Within the context of anomaly detection, it is feasible to compute the magnitude of the projection of data point x_i into the anomalous subspace to quantify its anomalous behaviour, which we define as the anomaly score for that point, the time series of which produces an anomaly score graph.

B. *Clustering using K-means*

We have employed the commonly used K-means, as it has been successfully used within seminal anomaly detection techniques in past and current literature [11], [12]. In the greatest majority of clustering-based techniques, there is always the underlying assumption that normal data instances lie distance-wise closer to a given centroid of a cluster, whereas anomalous data points are recognised due to their much longer distance [13].

Simply enough, the first step within the K-means approach is to select k random time-bins from training data as centroids of the clusters $C_1, C_2, \ldots, C_k$, where a subset of those randomly selected bins is ensured to contain samples for migration, anomalous events and background normal traffic. An immediate following step is to partition the selected feature observations $x_1, x_2, \ldots, x_n$ into k sets until the centroids of clusters stabilise by minimising sum of squares within the cluster:

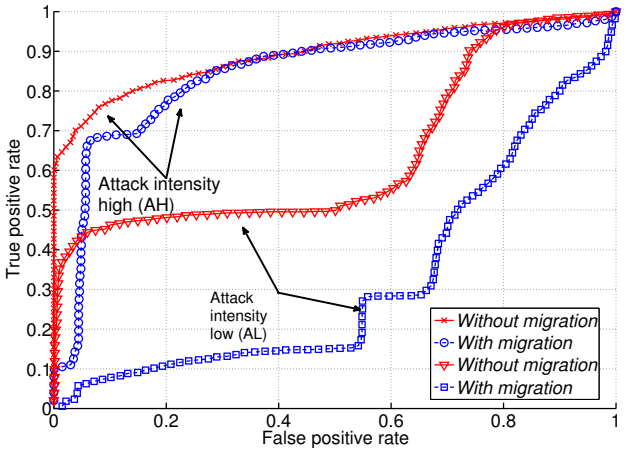
$$\sum_{j=1}^{k} \sum_{i=1}^{N} \|X_i^{(j)} - c_j\|^2 \tag{2}$$

where $\|X_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $X_{i^{(j)}}$ and the cluster centre c_j, is an indicator of the distance of the N data points from their respective cluster centre. Finally, for each test data instance, its distance to its closest cluster centroid is calculated as its anomaly score [14].

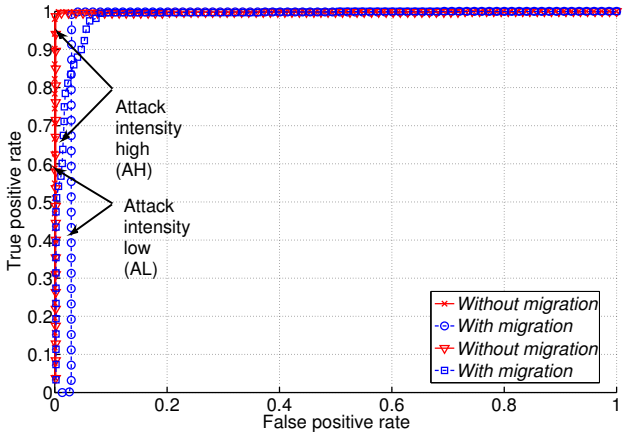
IV. Results and Analysis

Submitting each of our four 40 minutes datasets through the evaluation process produces one ASG and two ROCs, one including the effects of migration, and one without. By comparing these ROCs, we are able to assess the effect of migration on the AD technique that produced the table in the context of detecting the anomaly used to produce the traces that were submitted to the detector.

For illustration, two Anomaly score graphs (ASG) are presented here. Fig. 4a and Fig. 4b show some typical ASG obtained by using K-means for netscan and PCA for DoS respectively. High scores occurring within pink regions in each ASG plot indicate that each type of attack was successfully detected, i.e., these would be regarded as true positives. However, there are also high-scoring bins when only migration occurred (the green zones with no surrounding pink, at around bins 1350 and 1950), which would be regarded as false positives.

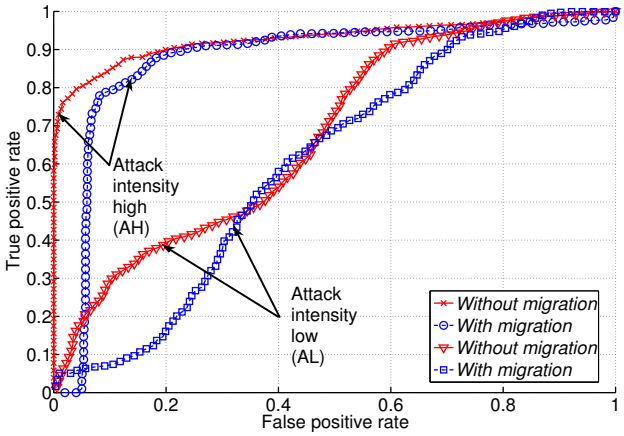


(a) ROC for netscan using K-means

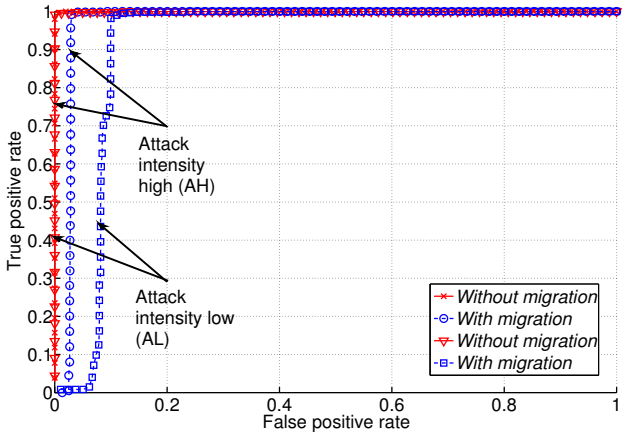


(b) ROC for DoS using K-means

Fig. 5: ROC for attacks using K-means



(a) ROC for netscan using PCA



(b) ROC for DoS using PCA

Fig. 6: ROC for DoS using PCA

We observe from our experimental runs that the presence of migration degrades the detection performance of both detectors for both the volume-based (DoS) and non-volume-based (NS) attack types.

Given the ROC curve outputs for K-means under netscan and DoS attack scenarios depicted at Fig. 5a and Fig. 5b respectively, we have noticed that the K-means method is sensitive to the chosen anomaly intensity and type. Also, in some instances its performance degrades even more while live migration occurs. However, for a volume-based attack such as DoS, K-means performs better in detecting both low and high intensity attacks but still its performance is negatively affected during migration. In particular, it can be observed that for low-intensity netscan, the true positive rate (TPR) is dropped by 35% in the event of a migration. In contrast under high attack intensity, the 65% TPR is achieved in face of migration with less than 10% false positive rates (FPR). Fig. 5b quantifies the effect of migration for volume-based attack (DoS), and it can be seen that more than 80% of the TPR is achieved with and without migration. There is a 5% rise of FPR when migration is introduced, which is acceptably small. In parallel, it can be seen in Fig. 6a that almost 80% of TPR is achieved under high-intensity netscan with and without migration for the PCA approach. Therefore, we conclude that migration has considerably less effect under high-intensity netscan on PCA. However, for the same attack type under low-intensity, the migration affect is more visible. In particular, the migration process impacts the performance by 30% when TPR reaches just about 20% yielding unacceptable number of false positives. For DoS, the PCA performs better comparatively for both attack types and is less affected under migration. As evidenced in Fig. 6b the FPR rate is less than 10% for both high and low intensities for more than 80% of the TPR.

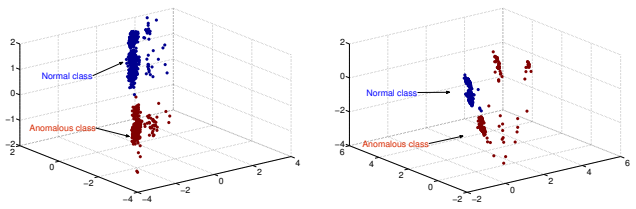
Finally, we argue that the detection sensitivity as evidenced by the ROC curves is truly affected by migration. This can also be illustrated by Fig. 7, where the changing behaviour of features' data points are illustrated. In more detail, Fig. 7a and Fig. 7b show the effect of the first three features *(packet size, byte size and active flows)* using scatter plots. Clearly, the clusters have dispersed due to the change on the feature distributions caused by migration, thus, new test data would adopt different distances from the pre-defined cluster centroids, making detection accuracy to vary.

Obviously, a different number of clusters may result in better clusters, e.g., if the migration already shows distinct periods of very low and very high traffic volume under normal conditions. However, the determination of an optimum number of clusters based on a cluster evaluation criterion is beyond the scope of this work. The results show that, under migration, the probability that a selected instance will be put in the correct cluster has decreased more. These results are similar to those obtained using PCA, indicating that VM migration makes the use of this form of AD technique unreliable for deployment of high assurance services. For a further examination of results, we refer the reader to [15] where we conducted a number of experiments on various state-of-the-art detection techniques by varying the type of and intensity of anomalies, and examining the live VM migration impact. As expected, we observed similar behaviour which indicates that VM live migration affects negatively the performance of several off-the-shelf detection techniques.

Further work will explore different options to devise solution around this problem. One approach might involve making migration aware anomaly detection technique whereby we could train anomaly detector for the amount of migration, and using this information to suppress alarms that relate to migration.

V. Conclusion

The new and intrinsic capabilities of cloud environments pose a number of security concerns that have not yet been fully assessed with respect to their implications on the performance of traditional off-the-shelf anomaly detection solutions. In particular, methods that empower the aspect of elasticity, resource management and service transparency such as live migration involve a range of system and network-wise activities. These are hard to be monitored by cloud providers and further are non-trivial to relate with any anomalous incidents that are likely to be initiated. In this work we have assessed the impact of intra-cloud live migration over the *de facto* anomaly detection techniques of PCA and K-means clustering within a controlled experimental cloud testbed. Through our results we demonstrate that live migration has a negative impact on both techniques since their performance degrade under different intensities of netscan and DoS scenarios.



(a) Scatter plot for netscan without migration

(b) Scatter plot for netscan with migration

Fig. 7: Effect on clustering under migration where blue and red clusters represent normal and anomalous class respectively.

Acknowledgements

This work is sponsored by EU FP7 Project SECCRIT (Secure Cloud Computing for Critical Infrastructure IT), grant agreement no. 312758 and UK-EPSRC funded EPSRC IU-ATC project, grant agreement no. EP/J016675/1

References

[1] M.A.C.Dekker, "Critical cloud computing: A CIIP perspective on cloud computing services," European Network and Information Security Agency, Tech. Rep., 2012.

[2] S. Berman, L. Kesterson-Townes, A. Marshall, and R. Srivathsa, "The power of cloud. driving business model innovation," IBM Institute for Business Value, Tech. Rep., 2012.

[3] VMware, "Migrating virtual machines in vsphere client," 2010. [Online]. Available: http://pubs.vmware.com/vsphere-51/topic/com.vmware.vsphere.vcenterhost.doc/GUID-3EE13ED8-172F-4560-B806-1E342AD7C486.html

[4] A. Bakshi and B. Yogesh, "Securing cloud from ddos attacks using intrusion detection system in virtual machine," in *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on*, 2010, pp. 260–264.

[5] F. Azmandian, M. Moffie, M. Alshawabkeh, J. Dy, J. Aslam, and D. Kaeli, "Virtual machine monitor-based lightweight intrusion detection," *SIGOPS Oper. Syst. Rev.*, vol. 45, no. 2, pp. 38–53, Jul. 2011.

[6] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '05. New York, NY, USA: ACM, 2005, pp. 217–228.

[7] C. Pascoal, M. Rosario de Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust pca for internet traffic anomaly detection," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 1755–1763.

[8] M. Lima, B. Zarpelao, L. Sampaio, J. Rodrigues, T. Abrao, and M. Proena, "Anomaly detection using baseline and k-means clustering," in *Software, Telecommunications and Computer Networks (SoftCOM), 2010 International Conference on*, 2010, pp. 305–309.

[9] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1. ACM, 2007, pp. 109–120.

[10] D. Brauckhoff, K. Salamatian, and M. May, "Applying pca for traffic anomaly detection: Problems and solutions," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2866–2870.

[11] N. Wu and J. Zhang, "Factor-analysis based anomaly detection and clustering." *Decision Support Systems*, vol. 42, no. 1, pp. 375–389, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/dss/dss42.html#WuZ06

[12] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[13] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[14] N. Wu and J. Zhang, "Factor analysis based anomaly detection," in *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, 2003, pp. 108–115.

[15] S. Steven, S. Noor-ul hassan, H. David, and B. Helge, "Anomaly detection techniques for cloud computing," Dec. 2013. [Online]. Available: https://www.seccrit.eu/upload/D4-1-Anomaly-Detection-Techniques-for-Cloud.pdf