Source-based Detection of Malicious Behavior in Cloud Environment

Rémi Cogranne

Source-based Detection of Malicious Behavior in Cloud Environment

Rémi COGRANNE

Université de Technologie de Troyes
LIST3N

Topic: Cybersecurity in Telecommunication and Networks ; Signal processing applied to telecommunications
Keywords: Statistical detection, signal processing, Security, Malicious behavior.

Abstract:
This short paper summarized our research and our contributions on the problem of source-based detection of BotCloud at large scale. We carefully took into account two major constraints of the real-life context: the tenant privacy and the cloud architecture in which tenant containers are spread out over different servers which requires a distributed detection solutions. In addition, one of the main strength of our works rests on the assessment over a real dataset collected on PlanetLab research infrastructure in which we created containers executing DDoS-type attacks with several software.

Introduction and Context
Cloud computing has gained an important role in providing high quality and cost-effective IT services by outsourcing part of their operations to dedicated cloud providers. If intrinsic security issues of this architecture have been extensively studied, it has recently been considered as a ready-to-use platform able to perform malicious activities, thus offering new targets for indirect threats. However, its large scale, the heterogeneous and dynamic nature of the activities it executes, as well as multi-tenancy and privacy-related issues, make the security operation complex. Consequently, cloud providers can hardly detect and mitigate malicious activities they unknowingly host.
The works presented in this paper form a contribution to the source-end detection of malicious activities a Cloud Service Providers can host unknowingly. We proposed a practical and efficient method for source-end detection of malicious activities in large scale virtualized environments while preserving tenants privacy.

Summary of the Proposed Method
The problem of source-based detection of malicious activities for Cloud Service Providers is threefold :

- First, the various type of malicious activities may exhibit very different pattern ; for instance, brute-force attacks for cracking passwords requires a lot of computing resources while DDoS attacks over-flood network interfaces. In addition, the legitimate activities of containers are also of very different activities.
- Second, cloud environment may be of very large scale hence the need of fully distributed and scalable solutions.
- Last and not least, the Cloud Service Provider must guarantee the tenants' privacy. It is indeed simple to look at the name of the running processes but that is not acceptable because it clearly violates the tenants privacy. However, for the sake of pricing of resource management, the Cloud Service Provider can access the use of each and every container at large (disk usage, network I/O, memory and CPU usage) other simple task such as load balancing and pricing would not even be possible.

To summarize, the proposed solution must be distributed and scalable, adaptative to allow the analysis of a very large range of traffic and can only use coarse grain metrics on the usage of the container.

The solution we have proposed is not a breakthrough in terms of statistical detection theory but it carefully takes into account all those constraints related with the operational context. We developed an original method based on legitimate activity self-adaptation which is used for background rejection and we use only the remaining data (that are not part of legitimate behavior) to detect matches with known malicious behaviors.

The proposed detection method if fully decentralized, hence scalable, and its principle is based on the three following steps :

1. The legitimate "workload" is estimated using the well-celebrated Principale Component Analysis (PCA) ; the PCA allows us to estimate the main "legitimate behavior patterns" adaptively using all containers metrics. Note that in practice PCA would require all metrics from all containers to be centralized at one central node. To allow a fully distributed implementation, we developed an interactive and decentralized method for the estimation of principal components (for benchmarking purpose it was implemented using gossip for epidemic information diffusion).
2. Using the PCA one can reject what is considered as the "leigitimate" workload by projecting (at containers level) the metrics onto the orthogonal complements spanned by the first principal components ; in other words, the metrics are projected on the linear subspace spanned by the principal components (which is considered as "legitimate" part) and the remaining part (lying into the orthogonal complement subspace) are the residues.
3. The detection is based on the matching of the residues with the traces left by the malware it is aimed at detecting which is measured simply by the projection onto "signatures" (metric values measured offline executing the BotCloud alone in separated containers).

The deployment of the proposed methodology in depicted in Figure 1 and leverages the distribution of tenants containers over different servers. First, at the server level, the metrics from all containers a server host are used to compute "local PCA"; then those estimations are shared by all containers from the same tenant from different servers. In the next iterations, the estimations of PCA are used and we used for benchmarking the method the Gossip epidemic diffusion process. We showed that this method allows an accurate estimation of the first principal components within only a few iterations.

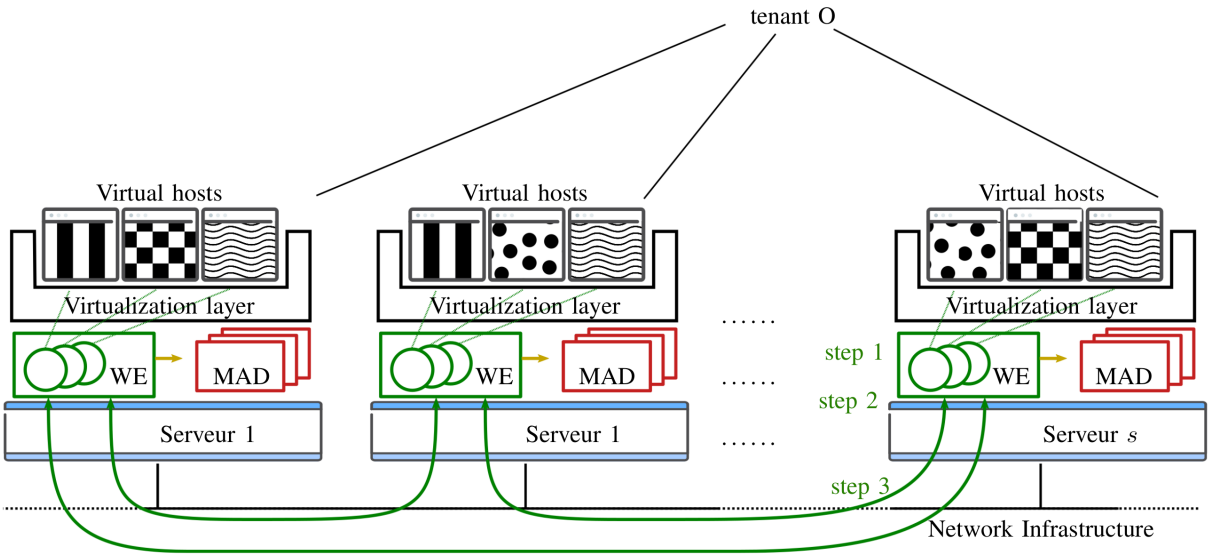
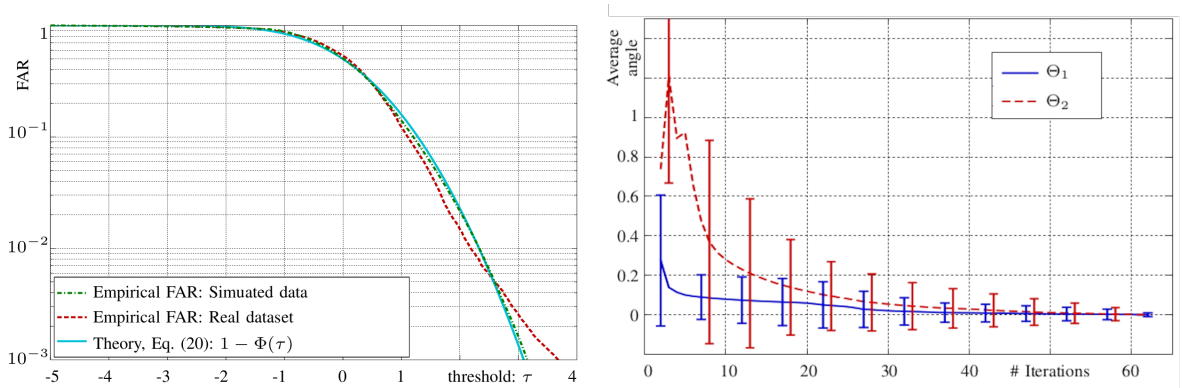


Figure 1: Representation of the distributed PCA estimation in two steps, gathering estimations from containers and then merging at tenants level.

Numerical Results

Due to the space limitations we will only present two main results. First Figure 2 shows the interest of the theoretical methodology. On the left, Figure 2 (a) compares the difference between the expected and theoretically established distribution of the decision statistics and the empirical one. Note that two types of data are used, some randomly generated and hence simulated (in dashed green) which fits very well with the established results. The dashed red line presents the results obtained from real data, using the monitoring of planetLab containers. On this real dataset it can be seen that the decision threshold fits well the empirical results for false-positive rate up to 2x 10e-3. Beyond that our auto-adaptative model based on distributed PCA is not accurate enough which create too much false-positive as compared to the desired rate.

On the right, Figure 2 (b) show, using a box plot emphasizing the average and 95% confidence interval, the evolution of the "angle" (in radians) between the first two principal components and the ground truth (obtained using a centralized computation). It can be clearly seen that the first principal components converge extremely fast. However, the distributed estimation of the second PCA requires more iterations. This can be explained in part by the fact that PCAs are computed with respect to each other: the second PCA can only be estimated when the first is already estimated accurately.



(a) Comparison between the theoretical and empirical false alarm rate (FAR) α_0.

(b) Averaged angle between estimated first two principal components vs. ground truth.

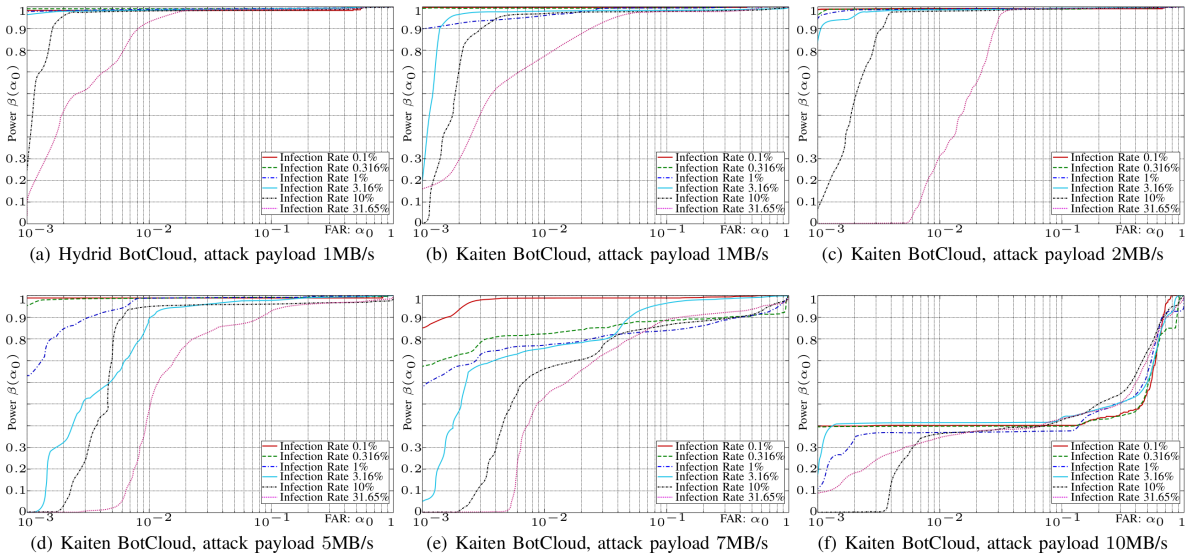
Figure 2: Results showing the efficiency of the theoretical method for computing the distribution of the decision statistics (left) and the distributed estimation of the PCA.

Last, figure 3 show the ROC (Receiver Operational Characteristics) curves for two different BotCloud implementation (Hybrid and Kaiten) and for various attack payload. In each and every figure, the ROC curve shows the true positive rate against the false-positive rate (in semi-logarithmic scale, to show the detection accuracy for low false-alarm rate) for various ratio of infected containers.

Contrast the results from Figure 3 (a) and 3 (b) which show the same results, with payload 1MB/s, but with the two different implementations. It can clearly be noted that the traces from Kaiten are much harder to detect. For these reasons the other results are presented using this implementation but the trends are similar in both cases.

Looking at Figure 3 (b)-(d) one can note that the detection accuracy generally increase which is what one can expect: the detection tends to become sharper and sharper when the payload increase. However, a careful look shows that for very high infection rate (32%) the trends go in the exact opposite direction. When looking at figure 3 (d)-(f) one can even note that the detection becomes quite poor when the attack payload becomes very large (as compared to the overall legitimate traffic); that is when either the payload and/or the infection rate are very large.

This apparently counterintuitive phenomenon can be explained by the estimation of legitimate "workload". Our method relies on the PCA on the collective metrics themselves to estimate the main behaviors. When the attack payload become much larger than it starts representing a majority of the traffic, it becomes naturally modeled within the principal components analysis and hence becomes parts of what our method consider as the legitimate background activity.



(a) Hydrid BotCloud, attack payload 1MB/s (b) Kaiten BotCloud, attack payload 1MB/s (c) Kaiten BotCloud, attack payload 2MB/s

(d) Kaiten BotCloud, attack payload 5MB/s (e) Kaiten BotCloud, attack payload 7MB/s (f) Kaiten BotCloud, attack payload 10MB/s

Figure 3: ROC Curves showing the efficiency of the proposed detection method for two BotCloud implementations, for various attack payload and various infection rates.

Follow-up Works and Conclusion

We have proposed a fully-decentralized method for estimation of containers workload and have exploited this method in a statistical well-founded detection method. While our method is not perfect, is original and has been shown rather efficient on a real and large dataset in terms of both computing complexity and detection accuracy. A follow-up works we are currently finishing is to implement this source-based detection method using in-network-caching system instead of the resources-consuming gossip epidemic diffusion and shown that a good trade-off between accuracy with large cache-size requirement and low complexity can be obtained.

References:

[1] R. Cogranne, G. Doyen, N. Ghadban and B. Hammi, "*Detecting BotClouds at Large Scale: A Decentralized and Robust Detection Method for Multi-Tenant Virtualized Environments*," in IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 68-82, March 2018, doi: 10.1109/TNSM.2017.2785628.

[2] N. Ghadban, R. Cogranne and G. Doyen, "*A decentralized approach for adaptive workload estimation in virtualized environments*," *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 1186-1194, doi: 10.23919/INM.2017.7987460.

Additional information

Contact details: Rémi COGRANNE : remi.cogranne@utt.fr
Researcher profile on the web: ORCID ; google Scholar ; UTT webpage
Member of labs / working groups / institutes: LIST3N lab ; UT Troyes
Topics of research: Detection methods for network security ; malicious behavior detection
Interest in the institute: Find collaboration within my topic of research (or perhaps explore novel research opportuinities)